



**PROPUESTA DE UN META-
MODELO DE CALIDAD PARA
EVALUAR HERRAMIENTAS DE
DESARROLLO DE SOFTWARE
CON SOPORTE A MDA**



ANA MERCEDES DÍAZ

UNIVERSIDAD CENTROCCIDENTAL "LISANDRO ALVARADO"
DECANATO DE CIENCIAS Y TECNOLOGÍA

Barquisimeto, Julio 2009

**PROPUESTA DE UN META-MODELO DE CALIDAD PARA
EVALUAR HERRAMIENTAS DE DESARROLLO DE
SOFTWARE CON SOPORTE A MDA**

Por

Ana Mercedes Díaz

Trabajo de Ascenso presentado para optar a la
categoría de Agregado en el escalafón del Personal
Docente y de Investigación

UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”

Decanato de Ciencias y Tecnología

Barquisimeto, Julio 2009

**PROPUESTA DE UN META-MODELO DE CALIDAD PARA
EVALUAR HERRAMIENTAS DE DESARROLLO DE
SOFTWARE CON SOPORTE A MDA**

Por

Ana Mercedes Díaz

Trabajo aprobado

Barquisimeto, de Julio del 2009

**PROPUESTA DE UN META-MODELO DE CALIDAD PARA
EVALUAR HERRAMIENTAS DE DESARROLLO DE
SOFTWARE CON SOPORTE A MDA**

ANA MERCEDES DÍAZ

**UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”
DECANATO DE CIENCIAS Y TECNOLOGÍA**

Barquisimeto, Julio 2009

DEDICATORIA

A mi hermosa hijita **Ana Paula**....DIOS te Bendiga mi amor. Te amo!!!

A mi madre **Justa Pastora (Tori)**, por todo el amor y el apoyo que me ha dado.

A la memoria de mi padre, **Luís Beltrán Linarez Cortéz**, por todas las enseñanzas que me dio durante toda su vida....Que DIOS lo tenga en la gloria.

AGRADECIMIENTOS.

A **María Angélica**, mi tutora de la Universidad Simón Bolívar, por apoyarme en todo lo referente a mi trabajo en Investigación.

A los profesores: **José Gregorio Sánchez, Wilfredo Angulo, José Antonio Gándara, Ninfa Barón, Maribel Mendonca, Freddy Torrealba, Maritza Bracho.**

A todos...mil Gracias...

PROPUESTA DE UN META-MODELO DE CALIDAD PARA EVALUAR HERRAMIENTAS DE DESARROLLO DE SOFTWARE CON SOPORTE A MDA

Prof. Ana Mercedes Díaz

RESUMEN

Uno de los objetivos principales de la Ingeniería del Software, es construir productos software de Calidad, guiados a través de un proceso de desarrollo. Estos procesos de desarrollo de software deben ser soportados por herramientas de desarrollo que apoyen si no todo el proceso de desarrollo, una etapa o conjunto de etapas de tal manera de ir construyendo la aplicación software de una manera documentada y consistente con respecto a todas las fases que incluya el proceso de desarrollo seleccionado.

Hoy en día existe la propuesta de un proceso de desarrollo dirigido por la transformación de modelos o Arquitectura Dirigida por Modelos (MDA), este proceso de desarrollo ha sido divulgado por la Object Management Group (OMG), y en virtud de esta nueva tendencia para desarrollar software, ya existen en el mercado Herramientas de Desarrollo de Software (HDS) que manifiestan ser herramientas MDA, es decir que soportan el enfoque MDA. Es por esta razón, que se origina este proyecto de investigación el cual tiene por objetivo **Proponer un Meta-Modelo de Calidad para evaluar herramientas de desarrollo de software con soporte a MDA.**

Durante el desarrollo de este Trabajo se estudiaron los aspectos más relevantes relacionados con Herramientas de desarrollo de software, el enfoque MDA, las herramientas de desarrollo MDA y el MC-CASE que es un modelo de calidad para evaluar herramientas CASE. Posteriormente se profundizó en la forma en que los investigadores en los últimos años han caracterizado a una herramienta MDA. Adicionalmente se hizo un Análisis Semántico del MC-CASE para determinar el nivel de cobertura con respecto a las caracterizaciones de MDA, para finalmente en base a todas estas observaciones, proponer un Meta-Modelo de Calidad, el cual incluye: un Modelo del Producto, el cual a su vez esta formado por el Modelo Teórico y el Modelo Matemático, es segundo lugar se tiene el Modelo del Proceso, donde se concentran todos los pasos que de manera sistemática se deben seguir para instanciar el modelo del producto y obtener resultados, y en tercer lugar se tiene el Modelo Humano, que establece los roles de los distintos integrantes que deben participar en la evaluación de una herramienta de desarrollo de software.

HERRAMIENTAS CASE (MC-CASE)		
5.1.	ANÁLISIS SEMÁNTICO DEL MC-CASE A NIVEL ARQUITECTURAL	101
5.2.	ANÁLISIS SEMÁNTICO DE LA APLICACIÓN DEL MODELO MC-CASE	126
5.3.	ANÁLISIS SEMÁNTICO COMPARATIVO DEL MC-CASE CON LAS	128
CARACTERIZACIONES DE MDA		
VI	PROPUESTA DE UN META-MODELO DE CALIDAD PARA EVALUAR	134
HERRAMIENTAS DE DESARROLLO DE SOFTWARE CON SOPORTE A		
MDA		
6.1.	MODELO DEL PRODUCTO	134
6.1.1.	Propuesta del Modelo Teórico MC-HDS-MDA	135
6.1.2.	Modelo Matemático del MC-HDS-MDA	195
6.2.	MODELO DEL PROCESO	201
6.3.	MODELO HUMANO	204
VI.	CONCLUSIONES Y RECOMENDACIONES	206
6.1.	META-MODELO DE CALIDAD PARA EVALUAR HERRAMIENTAS	206
DE DESARROLLO DE SOFTWARE CON SOPORTE A MDA		
6.2.	RECOMENDACIONES	208
REFERENCIAS BIBLIOGRÁFICAS		209

CAPÍTULO I

INTRODUCCIÓN

Las organizaciones que desarrollan Sistemas de Información (SI) deben tomar en cuenta la productividad y la calidad para decidir sobre cuál es el ambiente de desarrollo más apropiado. Una de las tecnologías que apoyan estos aspectos son las herramientas que soportan el desarrollo de los SI, o como se conoce en el área de sistemas, **Computer Aided Software Engineering (CASE)**, con las cuales ya se han tenido experiencias, tanto en el exterior como en Venezuela, y puede decirse que han incrementado la productividad de los analistas de los SI. [Mendoza, 1999].

Entre los beneficios de las herramientas CASE se encuentra, un aumento de la productividad de los analistas, diseñadores y programadores de SI, y un sistema de mayor calidad, lo cual origina una reducción de costos de mantenimiento y un mejor desempeño del sistema para los usuarios. [Sommerville, 1998] También fomenta un ambiente controlado en vez de promover un ambiente “ad hoc” de desarrollo de SI, lo que permite que los desarrolladores puedan crear y mantener a la vez SI grandes y pequeños. [Topper et. al., 1994].

Ahora bien, de acuerdo a los planteamientos anteriores, queda establecido que el uso de herramientas CASE es totalmente indispensable durante el desarrollo de un producto software, y esta importancia queda reforzada con la evolución que han sufrido las herramientas CASE luego se divulga a través de la OMG, una nueva forma de desarrollar software asistida por la transformación de los modelos.

La OMG en (OMG, 2003), establece que MDA proporciona una solución para los cambios de negocio y de tecnología, permitiendo construir aplicaciones independientes de la plataforma e implementarlas en plataformas como CORBA, J2EE o Servicios Web. Según la OMG, MDA se trata de un framework de desarrollo de software que define una nueva forma de construir software en la que se usan modelos del sistema a distintos niveles de abstracción para guiar todo el proceso de desarrollo, desde el análisis y diseño hasta el mantenimiento del sistema y su integración con futuros sistemas.

De acuerdo con (Anacleto, 2006), la ventaja principal de MDA radica en una clara y estricta **separación de responsabilidades**. Por un lado, se modelaran los MIP, que representan los modelos de nuestro negocio, y por el otro, los MEP con las preocupaciones tecnológicas. Esto permitirá que ambos modelos evolucionen por separado. De esta manera, si se quisiera, por ejemplo, modificar un aspecto técnico, bastaría con modificar el MEP sin que éstos tuvieran impacto en la lógica de negocios. Esta idea parte de un concepto que, en ingeniería de software, se llama **Guías de Diseño**. Particularmente, una de esas guías dice que el modelado de la solución debe ser guiado por el negocio. Esta guía se basa en la afirmación de que un cambio en el negocio seguramente producirá uno en el código, pero que los cambios en el código no deberían impactar en el negocio. MDA también permite lidiar con la complejidad del negocio, modelando a éste por separado, y permitiendo su análisis y mejora; **disminuir costos**, si se cuenta con una herramienta MDA adecuada a nuestras necesidades; y mejorar la **calidad** de nuestros modelos y procesos, mediante su análisis y la separación de responsabilidades.

La concepción de una nueva forma de construir software requiere de manera indispensable el uso de herramientas de transformación, que son las mismas herramientas CASE, pero soportando el enfoque MDA. Si se toma en cuenta que la Ingeniería de software busca la construcción de productos de calidad, se debe garantizar que todo el proceso de desarrollo exhiba características de calidad, por lo que las herramientas CASE que soporten MDA también deben ser productos de calidad. Entonces pareciera que las herramientas CASE - MDA como producto de software deben ser desarrolladas de una manera impecable, de tal forma que garanticen soportar el desarrollo de un SI, de todo esto surge una gran interrogante que es, ¿las herramientas CASE que soportan MDA son un producto de calidad?, ¿quién mide la calidad de una herramienta CASE - MDA?, ¿los compradores de herramientas CASE - MDA tienen al alcance algún mecanismo para comprobar la calidad de la herramienta CASE adquirida?

En función de poder adoptar productos de calidad Díaz A. en [Díaz, 2000] propone un Modelo de Calidad (MC-CASE) el cual tiene por objetivo evaluar y seleccionar Herramientas CASE. Este modelo se divide en tres grandes Categorías que son: Internos, Usuario y Externos; luego cada una de estas categorías esta conformada por un grupo de Características, las características se desglosan en Subcaracterísticas y éstas a su vez en Subcaracterísticas Atómica; finalmente cada subcaracterística atómica posee un grupo de métricas las cuales servirán para medir un aspecto específico de la Herramienta CASE.

Tomando como base que las herramientas CASE han evolucionado a herramientas de desarrollo de software donde algunas soportan el enfoque MDA, el modelo de calidad que permite medir su calidad también debe evolucionar. Esta es la razón por la cual surge este trabajo de investigación, con la intención de adaptar el MC-CASE a las herramientas de desarrollo de hoy en día. Claro está el objetivo de este trabajo es más ambicioso, en vista de que se quiere proponer un Meta-Modelo de Calidad que incluya el Modelo del Producto, el Modelo del Proceso y el Modelo Humano.

Sobre la base de estas inquietudes, el presente Trabajo de Investigación tiene los siguientes objetivos:

Objetivo General.

Proponer un Meta-Modelo de Calidad para evaluar Herramientas de Desarrollo de Software con soporte a MDA (MC-HDS-MDA).

Objetivos Específicos:

- Construir un marco teórico referencial que permita comprender y sustentar los tópicos relacionados con herramientas CASE o herramientas de desarrollo de software, Arquitectura Dirigida por Modelos (MDA) y Modelos de Calidad para evaluar herramientas de desarrollo de software.
- Caracterizar a las herramientas de desarrollo de software con soporte a MDA.
- Realizar un Análisis Semántico del Modelo de Calidad MC-CASE.
- Proponer el Meta-Modelo de Calidad, el cual incluye:
 - un Modelo del Producto que a su vez se compone de un modelo Teórico y un Modelo Matemático; el modelo teórico debe incluir la definición de la Arquitectura del Modelo de Calidad para evaluar herramientas de desarrollo de software con soporte a MDA y la propuesta de métricas que lo conforman.
 - un Modelo de Proceso y
 - un Modelo Humano.
- Estructurar las conclusiones finales del trabajo desarrollado y las recomendaciones para futuras investigaciones relacionadas con el tema.

Con la finalidad de alcanzar los objetivos ya mencionados, este Trabajo de Investigación ha sido estructurado de la siguiente manera:

En el Capítulo II - Marco Teórico, se esbozan todos los conceptos y teorías que sirven de base para desarrollar el Trabajo de Investigación, en primer lugar se estudian los aspectos relacionados con las herramientas CASE y su contribución en el desarrollo de SI, haciendo énfasis en el proceso de adopción y uso de las herramientas CASE por parte de las organizaciones desarrolladoras de SI, posteriormente se plantea la propuesta de la OMG, en cuanto al enfoque MDA, allí se estructuran todos los componentes que posee MDA y se establece la relación de un proceso de desarrollo tradicional y un proceso basado en MDA. Finalmente se comenta todo lo relativo al Modelo de Calidad para Seleccionar Herramientas CASE (MC-CASE), allí se describe con detalle su arquitectura y se hace énfasis en el conjunto de métricas que lo conforman.

En el Capítulo III - Marco Metodológico, se describen los pasos que se siguieron para el desarrollo de este Trabajo de Investigación.

En el Capítulo IV – Caracterización de MDA, se hace todo un recorrido por los diferentes autores que han publicado investigaciones referentes a los elementos que se deben considerar a la hora de caracterizar a una herramienta MDA.

En el Capítulo V - Análisis Semántico del MC-CASE, se realiza una revisión minuciosa del MC-CASE desde un punto de vista crítico, determinando fortalezas y debilidades, con respecto al modelo como tal, y con respecto al soporte brindado a MDA.

En el Capítulo VI – Propuesta del Meta - Modelo de Calidad para evaluar herramientas de desarrollo de software con soporte a MDA, se proponen los tres modelos que conforman al meta-modelo. Partiendo del Modelo del Producto, el cual internamente esta compuesto por el Modelo Teórico, donde se presenta toda la arquitectura del MC-HDS-MDA y la propuesta de métricas para todo el modelo, el otro componente del modelo de producto es el modelo matemático que le da robustez a los resultados obtenidos. Posteriormente se propone el Modelo del proceso, que describe los pasos que se deben seguir para aplicar el MC-HDS-MDA a una herramienta de desarrollo de software, y finalmente se plantea el Modelo Humano, el cual muestra los diferentes roles que deben ejercer las personas involucradas en la evaluación de una HDS.

En el Capítulo VII - Conclusiones y Recomendaciones, se estructuran las conclusiones que se derivaron del trabajo realizado, así como las sugerencias sobre la aplicación de los hallazgos obtenidos. Finalmente, se exponen las recomendaciones que sirvan de horizonte para la realización de futuras investigaciones en el área.

CAPÍTULO II

MARCO TEÓRICO

El objetivo de este capítulo es el de **establecer los fundamentos teóricos que permiten el desarrollo del presente Trabajo**. Para ello en la primera sección se tratan todos los aspectos relacionados con las **Herramientas CASE** partiendo de la conceptualización, caracterización, clasificación, ventajas y desventajas de las Herramientas CASE haciendo énfasis en su importancia para el desarrollo de los Sistemas de Información, hasta llegar al proceso de adquisición y los factores que determinan los problemas de su adopción y uso. Luego en la próxima sección se esboza lo concerniente a la **Arquitectura Dirigida por Modelos (MDA)**, allí se trata la definición de MDA, los modelos que lo conforman, los beneficios de MDA para el proceso de desarrollo de software y los fundamentos que lo soportan. Finalmente se desarrolla el tema del **Modelo de Calidad para seleccionar Herramientas CASE (MC-CASE)** propuesto por **Díaz, A.** donde se presenta su estructura arquitectónica y todas las métricas que lo componen.

2.1 HERRAMIENTAS CASE

Las organizaciones que desarrollan Sistemas de Información (SI) deben tomar en cuenta la productividad y la calidad para decidir sobre cuál es el ambiente de desarrollo más apropiado. Una de las tecnologías que apoyan estos aspectos son las herramientas que soportan el desarrollo de los SI, o como se conoce en el área de sistemas, **Computer Aided Software Engineering (CASE)**, con las cuales ya se han tenido experiencias, tanto en el exterior como en Venezuela, y puede decirse que han incrementado la productividad de los analistas de los SI. [Mendoza, 1999].

*"El objetivo de todas las herramientas CASE es **soportar el proceso de desarrollo de software**; en este sentido, recuérdese que $SI = software + hardware + gente + organización$ ".* [Mendoza, 1999]

Las herramientas CASE son un complemento de la "caja de herramientas" del desarrollador de SI; le proporcionan la posibilidad de automatizar actividades manuales y de mejorar su visión general del proceso de desarrollo de los SI. Al igual que las herramientas de la ingeniería y del diseño asistido por computadora que utilizan los ingenieros de otras disciplinas, las herramientas CASE ayudan a asegurar que la calidad sea algo diseñado antes de llegar a construir el producto. [Pressman, 1998]

Según Topper et. al. [Topper et. al., 1994], citado por [Mendoza, 1999], la relación entre las herramientas CASE y el proceso de desarrollo de SI es bastante fuerte, ya que si **las herramientas CASE son utilizadas inteligentemente, ellas pueden acelerar sustancialmente el proceso de desarrollo.**

Entre los beneficios de las herramientas CASE se encuentra, un aumento de la productividad de los analistas, diseñadores y programadores de SI, y un sistema de mayor calidad, lo cual origina una reducción de costos de mantenimiento y un mejor desempeño del sistema para los usuarios. [Sommerville, 1998] También fomenta un ambiente controlado en vez de promover un ambiente “ad hoc” de desarrollo de SI, lo que permite que los desarrolladores puedan crear y mantener a la vez SI grandes y pequeños. [Topper et. al., 1994].

Debido a que la tecnología CASE ofrece a los desarrolladores una amplia selección de herramientas innovadoras de software que pueden utilizarse como componentes automatizados de un proceso de desarrollo de SI, en la próxima sección se definirán las herramientas CASE como producto de software.

2.1.1 HERRAMIENTAS CASE COMO PRODUCTO DE SOFTWARE

Para Norman y Forte [Norman y Forte, 1992], las herramientas CASE son un software que soporta un enfoque de ingeniería para desarrollo de software en todas las etapas del proceso. Por su parte McClure en [McClure, 1989], señala que CASE es automatización del software, la idea básica es proveer un conjunto de herramientas bien integradas que automaticen el proceso de desarrollo del software y su mantenimiento.

Según Freedman [Freedman, 1993], herramientas *CASE* *"es el software que se utiliza en cualquiera o en todas las fases de desarrollo de un Sistema de Información, incluyendo el análisis, diseño y programación. Las herramientas CASE proporcionan métodos automáticos para diseñar y documentar las técnicas tradicionales de programación estructurada. La meta última de este enfoque CASE es proveer un lenguaje para describir el sistema completo, que sea suficiente para generar todos los programas necesarios"*.

Por su parte, Brown et. al. [Brown et. al., 1994] citado por [Mendoza, 1999], establecen que *CASE es el uso del soporte computarizado en el proceso de desarrollo de software. Una herramienta*

CASE es un producto de computadoras que ayuda en el soporte de una o más actividades de la ingeniería de software dentro de un proceso de desarrollo de software.

Paralelamente, Topper et. al. [Topper et. al., 1994] citado por [Mendoza, 1999], definen CASE como *"la automatización de las técnicas, notaciones y métodos para el desarrollo de software. CASE incluye herramientas de diagramación para los modelos y las especificaciones, verificación y chequeo, un repositorio común y capacidades de generación para la documentación, código y casos de evaluación"*.

Además, Sommerville [Sommerville, 1998] cita que *"en pocos años, un amplio rango de herramientas para soportar el desarrollo de software han sido desarrolladas y el término computer-aided software engineering está siendo usado como un término genérico para el soporte automatizado de la ingeniería de software"*.

Finalmente, Whitten y Bentley [Whitten y Bentley, 1998], exponen que CASE es la aplicación de la tecnología de la información a las actividades, las técnicas y las metodologías de desarrollo de sistemas. Las herramientas CASE son programas (software) que automatizan o soportan una o más fases del ciclo de vida de desarrollo de sistemas. La tecnología tiene el propósito de acelerar el proceso de desarrollo de los sistemas y mejorar su calidad.

Para efectos de esta investigación se definirá herramientas CASE como un **producto de software que soporta en forma automatizada todo o parte de el proceso de desarrollo de Sistemas de Información incorporando las actividades, técnicas y metodologías tanto estructuradas como orientadas a objetos** con el propósito de **generar software de calidad** o en otras palabras software que satisfaga plenamente las necesidades del cliente, y que evolucione con el cliente.

Debido a la creciente innovación tecnológica que ofrece a los desarrolladores herramientas que soportan el desarrollo de los SI, con el principal objetivo de garantizar la calidad del producto de software, es importante discutir las principales características que presentan las herramientas CASE.

2.1.2 CARACTERÍSTICAS DE LAS HERRAMIENTAS CASE

En la investigación realizada por De Freitas en Venezuela, [De Freitas, 1997], se indica que ninguna de las herramientas CASE que ella ha examinado hasta el momento es perfecta, además menciona

la opinión de diferentes expertos acerca de las características que debe poseer una buena herramienta CASE.

Yourdon citado por [De Freitas, 1997] considera que la mejor herramienta CASE es la que permite analizar el modelo de los requerimientos de los usuarios. Señala que los elementos esenciales de una buena CASE son: a) flexibilidad de diagramación, de esta manera, el usuario de la CASE puede ver todo lo que el quiera, en todas las formas que desee; b) el buen chequeo de errores hacia los métodos de análisis orientado a objetos y el diseño orientado a objetos y c) el bajo precio con la finalidad de experimentar con la herramienta sin un compromiso.

Según Constantine citado por [De Freitas, 1997], una excelente herramienta CASE es más que una herramienta de diagramación. Actualmente debe soportar el trabajo de los desarrolladores reales trabajando con metodologías particulares. También añade que una excelente herramienta CASE, es la que permite ir hacia delante y hacia atrás en el análisis y desarrollo de los diferentes puntos de vista de algunos modelos fundamentales. Considera además, que la mejor herramienta CASE es la que soporta la interactividad, el pensamiento no lineal y la exploración de los procesos en el desarrollo y evolución de los sistemas reales.

En tal sentido, Coad citado por [De Freitas, 1997], plantea que una excelente herramienta CASE debe tener el modelo objeto en una ventana y C++ en la otra. Pudiendo editar en una u otra ventana, y en la segunda ventana obtener continuidad entre ellas.

Por otro lado, Booch citado por [De Freitas, 1997] señala, la herramienta CASE ideal es la que permite ayudar en la creación y visualización de los modelos. Sin embargo, para Rumbaugh la transparencia es el elemento clave de una excelente herramienta CASE. El mecanismo para usar la herramienta debe aparecer en el fondo, haciendo la herramienta muy amigable.

Por su parte, Mellor y Montrose citado por [De Freitas, 1997], señalan que una excelente herramienta CASE debe incluir los siguientes elementos: a) grupo de soporte, quienes analizan las actividades del grupo; b) control de configuración, en un menor nivel de diagrama; c) aplicación de un método en particular, esto en contraste a los modelos de las CASE populares que sólo soportan una notación; d) flexibilidad de los reportes e impresiones; e) soporte de grandes modelos abstractos; f) habilidad para analizar múltiples modelos de un mismo proyecto y de otros proyectos

presentes en una ventana y g) integridad de las bases de datos, que incluye autochequeo, autoreparación y librería de administración y diferentes niveles de seguridad.

Además, Shlaer y Mellor citado por [De Freitas, 1997] establecen tres características principales: a) particionar el problema dentro de un dominio; b) rigurosidad de análisis de cada dominio y ejecución de cada dominio del modelo abstracto; y c) trasladar el análisis de cada dominio a una implementación. En una herramienta CASE, esto se traduce en: dominio de navegación, modelo de subsistemas, descripción de los enlaces y soporte para el análisis de cada diagrama, incluyendo el modelo de información de objeto, modelo de transición de estado, modelo de procesos, descripciones de los objetos, y sus atributos, modelo de comunicación de los objetos, modelo de acceso de objetos y el control de navegación. Las herramientas CASE deberían hacer un chequeo donde detectaran errores en los modelos de análisis además de soportar la simulación y ejecutar cada dominio del modelo estudiado.

Recientemente Losavio junto a otros investigadores [Losavio et al., 1999], proponen un modelo conceptual dividido en tres niveles; procesos, servicios y mecanismos donde a cada uno de estos le asignan cierta funcionalidad, que se traduce en características para las herramientas CASE orientadas a objetos, la cual es detallada a continuación.

Procesos:

- El ambiente CASE debe apoyar el paradigma orientado a objeto, es decir, debe permitir que los procesos de desarrollo de software hagan uso de las ventajas de la orientación a objetos.
- Debe contemplar el uso de los artefactos (componentes de software) así como el reuso de los componentes de diseño por medio de las estructuras o los patrones de diseño.
- Debe permitir la combinación de métodos de diferentes fases del desarrollo de software.
- Debe adaptarse a los procesos organizacionales de la compañía.

Servicios, son clasificados en base a las categorías dadas por Sharon y Anderson; desarrollo, gerencia, soporte y generales.

Los servicios de desarrollo deben:

- Incluir las herramientas que soportan el análisis, diseño e implementación del software.
- Poseer la capacidad para expresar con notaciones los diferentes modelos de los diferentes métodos orientados a objetos.
- Proveer el servicio de control de versión para el desarrollador.
- Proveer generación de código.

- Proveer la trazabilidad de objetos, en el sentido de que las actividades de análisis, diseño y programación se enlacen estrechamente y sean interactivas.

Los servicios de gerencia, incluyen herramientas para la planificación, el control y monitoreo de proyectos; y deben:

- Soportar dos tipos de usuarios: gerentes y desarrolladores.
- Proveer un "backup" de la definición y la gerencia de las estructuras de los proyectos.

Los servicios de soporte incluyen herramientas de ayuda y procedimientos para el mantenimiento y mejoramiento de los sistemas existentes. Ellos soportan:

- Groupware: Los desarrolladores deben trabajar concurrentemente como un equipo y en diferentes localidades.
- Visualización de la información.
- La definición, acceso y control de uso sobre los artefactos usados en el desarrollo del sistema.
- La planificación y el control de los proyectos.

Los servicios generales incluyen las facilidades que son comunes a todos los servicios anteriores:

- Browsing
- Integración de interfaces.

Mecanismos, para un OOCE (Ambiente CASE orientado a objeto):

- Arquitecturas Cliente/Servidor con mecanismos para la comunicación entre herramientas, usando varias aproximaciones como (CORBA) Common Object Request Broker Architecture, (OLE) Object Linking and Embedding, etc.
- El estándar PCTE (Portable Common Tool Environment).
- Bases de datos orientadas a objetos.

Para un OOCE Federativo:

- Arquitecturas Cliente/Servidor con mecanismos para comunicaciones entre herramientas usando varios acercamientos como CORBA, OLE, etc.
- Mecanismos estándares de importación/exportación.
- Bases de datos independientes.
- Arquitecturas que soporten la integración por interfaces entre varias herramientas participantes.

Paralelamente con Losavio: Rojas y Pérez [Rojas y Pérez, 1999], realizan una investigación donde evalúan una serie de herramientas CASE de acuerdo con las siguientes características: fases del ciclo de vida que soportaba la herramienta, metodologías, número de usuarios concurrentes, componentes de análisis, repositorio común, facilidad de documentación, control de proyectos,

plataformas, manejadores de bases de datos, generación de código, nivel de generación de prototipo, la ingeniería de reverso. Otras características consideradas fueron la curva de aprendizaje, flexibilidad, tipo de integración, soporte técnico, y el material de apoyo; certificaciones obtenidas por la empresa con una determinada herramienta y al grado de mejoramiento de la misma.

Note que las características de las herramientas CASE propuestas por los diferentes autores convergen en los siguientes aspectos: las CASE además de permitir la diagramación de forma sencilla, deben soportar las metodologías ya establecidas, deben cuidar el balanceo o el chequeo entre los modelos de una fase y otra del ciclo de desarrollo, debe permitir ir hacia delante y hacia atrás en el diseño y desarrollo de los principales modelos, debe ser amigable permitiendo la visualización de los modelos fácilmente, deben permitir la integración de las bases de datos y además establecer niveles de seguridad que garanticen la confiabilidad de todos los elementos involucrados entre otros.

En virtud de que existe un amplio espectro de herramientas CASE innovadoras de software que pueden utilizarse como componentes automatizados de un proceso de desarrollo de SI, es necesario hacer una clasificación de las mismas.

2.1.3 CLASIFICACIÓN DE LAS HERRAMIENTAS CASE

Siempre que se intenta clasificar las herramientas CASE, se corren riesgos, ya que en, en primer lugar, se puede dar lugar a confusión al situar una herramienta determinada dentro de una categoría cuando otros podrían pensar que pertenece a otra categoría, y, en segundo lugar, se puede pensar que se ha omitido una categoría; sin embargo, se aclara que es necesario crear una taxonomía para comprender mejor el alcance de la CASE y para apreciar mejor dónde se pueden aplicar estas herramientas en el proceso de desarrollo de SI. [Pressman, 1998].

Para [Mendoza, 1999], el desarrollo de un proyecto de software incluye todos los aspectos de dirección, administración y tecnologías utilizadas. En consecuencia, una herramienta CASE es, en general, *un producto computarizado que soporta una o más actividades de la Ingeniería de Software, en el contexto de un proceso de desarrollo de software*. [Brown et. al., 1994] [Losavio, 1997].

Pressman [Pressman, 1998] propone la siguiente clasificación de herramientas CASE de acuerdo con la función que apoyan (ver Tabla N° 1), y unas que él denomina Herramientas de Estructura,

que permiten la gestión de bases de datos, gestión de configuraciones y la integración con otras herramientas CASE.

Tipos de Herramientas CASE de acuerdo a la función (Pressman)
Planificación de Sistemas de Gestión
Gestión de Proyectos
Soporte de Desarrollo
Análisis y Diseño
Creación de Prototipos
Programación
Integración y Prueba
Mantenimiento

Tabla N° 1. Clasificación de las Herramientas CASE.

Fuente: [Pressman, 1998]

Por otra parte, se puede explotar cada uno de los tipos que propone Pressman, para seguir clasificándolas más agudamente a partir de los modelos y las metodologías que se utilicen para realizar las tareas correspondientes.

Por ejemplo, Martin [Martin, 1994] propone una clasificación a partir de su trabajo que se concentra básicamente en la orientación a objetos. Éste explica que una clasificación amplia para este tipo de herramientas se dividen en: Aquellas que atienden el mundo de la orientación a objetos y aquellas que apuntan en otra dirección.

Por su parte, Sommerville [Sommerville, 1996] sugiere que la funcionalidad es una posible dimensión de clasificación. Otras posibles dimensiones son aquellas que prestan :

1. **Soporte al proceso:** cuáles fases del proceso son soportadas por la herramienta, trayendo entonces la clasificación de herramientas de diseño, de programación, de mantenimiento, etc.. Esta es coincidente con la dada anteriormente por Pressman.
2. **Amplitud del soporte:** es decir cuán extensamente es soportada la actividad del proceso. Un soporte especializado implica que soporta unas actividades muy específicas del proceso, tal como un compilador. Un soporte generalizado implica soportar toda una fase del proceso como pudiera ser por ejemplo la de diseño.

Fuggetta [Fuggetta, 1993] en [Sommerville, 1996] también propone una clasificación de tres categorías:

1. **Herramientas:** soportan tareas individuales del proceso, por ejemplo chequeo de consistencia en un diseño.
2. **Workbenches** (mesas de trabajo): soportan fases del proceso. Normalmente consisten en un conjunto de herramientas con un mayor grado de integración.

3. **Ambientes:** soportan todo o al menos una parte muy significativa del proceso. Normalmente incluyen varios workbenches

De acuerdo con Gibson en [Gibson, 1989], las herramientas CASE se pueden clasificar de acuerdo con los componentes que las conformen, como puede verse a continuación:

1. **Upper CASE:** llamado también planificación asistida por el computador, se refiere a componentes asistidos por el computador que soportan planificación corporativa.
2. **Middle CASE:** un componente que soporta análisis y diseño de sistemas.
3. **Lower CASE:** se refiere a un componente que soporta el desarrollo de sistemas.

Para Smith en [Smith, 1989], las herramientas CASE se pueden categorizar de acuerdo con las etapas del ciclo de vida de sistemas, entre las cuales se pueden identificar las siguientes:

1. **Herramientas de Planeación:** ayudan en el desarrollo de los modelos de la empresa y el plan de los datos estratégicos. Estas herramientas proporcionan una matriz de análisis que permite la revisión de las relaciones existentes entre la organización y los procesos, los procesos y la data; y las aplicaciones y los procesos.
2. **Herramientas de Análisis y Diseño:** soportan las fases de análisis y diseño del ciclo de vida de sistemas. Estas herramientas están divididas en dos subcategorías que son herramientas de modelado de procesos y herramientas de modelado de datos.
 - 2.1 **Herramientas de modelado de procesos:** analizan las funciones, que un sistema realiza, las entradas y las salidas y las relaciones entre las funciones.
 - 2.2 **Herramientas de modelado de datos:** representan la estructura de los datos, junto con las relaciones entre los objetos de datos.
3. **Herramientas de nivel de código:** generan código estructurado o analizan el código existente. Dos subcategorías son identificadas:
 - 3.1 **Herramientas de generación de código:** generan código para nuevas aplicaciones desde las especificaciones.
 - 3.2 **Herramientas de ingeniería de reverso:** generan las especificaciones de diseño o nuevo código estructurado a partir del código existente.

Finalmente, Brown et. al. [Brown et. al., 1994] también hace un resumen de criterios de clasificación estos son:

1. **Interacción.** aquellas que son interactivas con respecto a aquellas que no lo son.
2. **Situación en el ciclo de vida (Horizontal).** aquellas que asisten en procesos iniciales del ciclo de vida del software, denominadas generalmente como Herramientas Front-End, con respecto a aquellas que atienden hacia al final del ciclo de vida, denominadas generalmente como Back-End.

3. **Situación en el ciclo de vida (Vertical).** aquellas que se especializan en un nodo particular del ciclo de vida. Esta es la dirección en la que apunta Pressman en su clasificación.

Brown [Brown et. al., 1994] señala que, desafortunadamente, todos los criterios (que él identifica) son problemáticos. Por ejemplo para clasificar una herramienta de interactiva o no, habría que especificar formalmente lo que significa la interacción, situación que genera un conflicto conceptual en el espacio del software.

Por otra parte, para poder clasificar a una herramienta de “back-end” o “front-end” habría que partir de la base de que las herramientas se inscriben en un contexto donde se asume el ciclo de vida clásico para el software. En este sentido, aquellas herramientas que se definan a partir de modelos que proponen otros ciclos, no podrían ser herramientas sujetas a este tipo de clasificación. Otro problema con esta clasificación tiene que ver con la delimitación de lo que es el “front” con respecto a lo que incluye el “back”. ¿Cuál es la frontera que separa a un espacio del otro?

Sin embargo, para efectos de este estudio y considerando que la mayoría de las ofertas de herramientas CASE en el mercado están orientadas hacia Workbenches o Ambientes, se adoptará la clasificación según lo propuesto por Fuggetta. Puesto que tienen como característica principal la posibilidad de integrarse con otras herramientas CASE.

Desafortunadamente dentro de la gama tan amplia de herramientas CASE que existen actualmente en el mercado, hay algunos productos con determinadas limitaciones por lo cual en la próxima sección se tratará este tópico.

2.1.4 VENTAJAS Y DESVENTAJAS DE LAS HERRAMIENTAS CASE

Las herramientas CASE incorporan muchos beneficios, pero como lo señala Smith en [Smith, 1989], también tienen sus limitaciones. Es por ello que es necesario mencionar las ventajas más importantes de las herramientas CASE, de acuerdo con los puntos de coincidencia de diferentes autores.

- Permiten a los programadores tomar ventajas de las especificaciones de los subsistemas o del código fuente generado directamente desde las especificaciones de los subsistemas generadores de código. Por lo tanto la tecnología CASE constituye una herramienta útil para los distintos niveles de la organización en los esfuerzos futuros de desarrollo de sistemas [Kwol y Arnett, 1993].

- Facilitan la estandarización de los procedimientos de trabajo y crean una disciplina de diseño [Orlikowski, 1993] y así mejorar la calidad de sistemas [Burch y Grudnitski, 1992], [McClure, 1989], [Premkumar y Potter, 1995], [Pressman, 1998], [Viloria, 1995].
- Son utilizadas por los analistas para incrementar la productividad y mejorar la calidad de los sistemas [Argilia, 1993], [Chen y Norman, 1992], [Davis, 1992], [Kendall y Kendall, 1997], [McClure, 1993], [Pressman, 1993], [Senn, 1992], [Smith, 1989], citados por [De Freitas, 1997].
- Las herramientas CASE son: *“la solución que puede proveer el Gerente de Sistemas de Información (SI) la capacidad de incrementar la productividad del desarrollo de sistemas, mejorar la calidad de los sistemas, suministrar la mejor documentación y así reducir el tiempo de respuesta de los requerimientos de sistemas que redundan en una mejora de servicio a los usuarios”*. [Premkumar y Potter, 1995].
- La ventaja principal de las herramientas CASE es incorporar disciplina [Lazar, 1995]. Paralelamente, Viloria [Viloria, 1995] dice que, para los analistas venezolanos, las herramientas CASE aumentan la productividad.
- El poder contar con un intercambio de información y datos entre las herramientas U-CASE y L-CASE, permite al analista hacer cambios en la fase de programación, haciendo pocas correcciones en las fases de análisis y diseño [Goddard, 1994]. Las herramientas CASE, mediante la comprobación de la validez y la consistencia del modelo, aumentan la confiabilidad en la representación del análisis, pues ayuda a eliminar errores antes de que se propaguen al diseño o, lo que es peor, al código mismo. [Pressman, 1993].
- Las herramientas CASE, no solo traducen la descripción de un sistema a un programa operativo, sino que también ayudan a verificar la corrección de las especificaciones del sistema, de forma tal, que la salida resultante satisfaga los requisitos del usuario. [Pressman, 1993]. Las herramientas CASE permiten la integración de los proyectos en cualquiera de las fases o a través de las diferentes fases del ciclo de vida. [Smith, 1993].
- Según Smith [Smith, 1989], las ventajas que presentan las herramientas CASE, se pueden resumir teniendo en cuenta los beneficios que le aportan a la organización, entre los cuales se citan: 1) Integración del proyecto, o dentro de una fase específica o entre diferentes fases del ciclo de vida. 2) Implementación de: la planificación estratégica de la data, modelado de la data, y metodologías estructuradas. 3) Una mayor probabilidad de que los sistemas entregados se ajusten a los requerimientos de los usuarios. 4) Sistemas que se adaptan mejor a los cambios del negocio. 5) Mejor mantenimiento de las especificaciones y del código. 6) Mejoramiento de la comunicación con los usuarios.

- Para Yourdon [Yourdon, 1993], las herramientas CASE dan la posibilidad de verificar la consistencia de diversos tipos de modelos de un sistema. Los beneficios de las CASE integradas (I-CASE) incluyen: 1) la transferencia fluida de información (modelos, programas, documentos, datos) entre herramientas y etapas de la ingeniería del software; 2) la reducción del esfuerzo requerido para realizar actividades de soporte como la gestión de la configuración del software, el control de calidad y la generación de documentos; 3) un aumento en el control de los proyectos que se consigue mediante una mejor planificación, el monitoreo y la comunicación y 4) mejora de la coordinación entre los miembros del equipo que trabajan en un proyecto grande de software.
- Gibson [Gibson, 1989] señala que las herramientas CASE tienen la habilidad de: a) impulsar la productividad y prevenir los defectos; b) resaltar los esfuerzos del equipo a través de la tecnología de coordinación, el chequeo de la consistencia de todo el proyecto y los datos de un diseño compartido; c) ayudar a hacer el desarrollo de software más interactivo y consistente con la manera en que la gente realmente piensa y trabaja; d) un entendimiento más claro de la dirección de la compañía y cómo contribuir a su éxito. Estos modelos de planeación dejan evaluar el impacto de cambiar valores para ciertas especificaciones de planeación; e) los modelos también proveen una base para especificaciones del proyecto; f) proveen métodos más fáciles para cambiar el diseño de sistemas; g) facilita que los analistas entiendan el problema y como resolverlo a través de un proceso iterativo con los usuarios; h) facilita las sesiones JAD; i) facilitan la creación de prototipos; j) generación de código de programa y k) facilita las actividades de mantenimiento.
- Para Kendall y Kendall [Kendall y Kendall, 1999], existen cuatro ventajas que brindan las herramientas CASE al ser incorporadas en la organización: 1) alcanzar un nivel competitivo sobre otras empresas que están en la misma misión, b) mejorar la uniformidad o estandarización en el desarrollo de sistemas, c) la emergencia de afrontar nuevos requerimientos al ver problemas antiguos de diferentes maneras y d) la adopción de la automatización del proceso de desarrollo como parte del trabajo de rutina de los analistas.

Resumiendo lo anterior, queda claro que las herramientas CASE permiten alcanzar un nivel competitivo a las organizaciones, mejorar la estandarización de los procesos de desarrollo, mejorar la calidad de los Sistemas de Información, incrementar la productividad, incorporar disciplina en la organización, y verificar la consistencia y validez de los modelos que componen el proceso de desarrollo.

A pesar de que existen muchas ventajas los expertos consideran que las herramientas CASE poseen las siguientes desventajas:

- Para Kendall y Kendall [Kendall y Kendall, 1999] existen cuatro grandes desventajas: 1) el problema de retraso del trabajo del analista al querer adoptar la herramienta CASE; 2) la necesidad de superar dudas y la posible falta de experiencia del responsable del desarrollo; 3) el cambio de los métodos de trabajo y 4) la inversión en un producto costoso.
- El alto costo que tienen las herramientas CASE como factor inhibitor para su adquisición. [Huff, 1992]; [Kemerer, 1992]; [Premkumar y Porter, 1995]; [Smith, 1989]; [Yourdon, 1993].
- Las herramientas CASE en determinadas ocasiones se quedan cortas con respecto a las expectativas creadas. Esto se manifiesta en la incapacidad general para auxiliar a los analistas en todo el ciclo de vida de un sistema, sobre todo en las inadecuadas capacidades de generación de código de algunas de ellas [Kendall y Kendall, 1999].
- Por otro lado para Tate [Tate et. al., 1992], las razones por las cuales las herramientas CASE no cumplen con los objetivos esperados son las siguientes: a) dificultades para adaptarse al cambio, pues es un cambio cultural, no sólo técnico, el que se debe asimilar; b) muchas de las herramientas necesarias para automatizar el proceso son inmaduras o no están disponibles. La inmensa mayoría presenta insuficiencias a la hora de integrar las actividades de un proyecto; c) Es difícil pasar de un análisis casi solitario, a la realización del análisis en colaboración con los usuarios; d) muchas empresas no tienen o no cumplen con las prácticas de gestión de software apropiadas, sin las cuales la automatización del proceso suele resultar ineficaz o imposible; e) a veces se tiene la creencia de que las nuevas herramientas son soluciones mágicas, cuando en realidad el resultado se ve a mediano y largo plazo; f) muchas empresas de software están en situación de "Backlog" y están tan ocupadas cumpliendo los plazos, que carecen de tiempo para pensar en nuevas políticas de desarrollo; g) desengaño debido a la ficción de los anunciantes.
- Las limitaciones que presentan las herramientas CASE expuestas por Smith [Smith, 1989] son: a) Aplicación desigual de las fases del ciclo de desarrollo de sistemas. b) Interfaces débiles entre las herramientas; c) Capacidades limitadas para la conexión en red; d) Integración débil con los diccionarios de los datos de la corporación; e) Poco flexibles; f) Inadecuadas capacidades para la generación de código.

De lo anterior se puede concluir que si bien las herramientas CASE poseen poderosos beneficios para el desarrollo de Sistemas de Información, también poseen algunas limitaciones que pueden resultar en razones para que las organizaciones tomen en consideración su incorporación,

resumiéndolas se tiene, cambios en los procesos de desarrollo de la organización, elevados costos y expectativas no satisfechas.

Ahora bien, si se sabe que la utilización de herramientas CASE ayuda en todas las tareas y en la mayoría de las actividades del proceso de desarrollo de SI, resulta interesante conocer ¿cómo ha sido el proceso de adopción y uso de las herramientas CASE por parte de las organizaciones encargadas del desarrollo de Sistemas de Información?.

2.1.5 ADQUISICIÓN Y USO DE LAS HERRAMIENTAS CASE EN VENEZUELA

Las organizaciones deben buscar las herramientas que mejor se adapten a sus requerimientos, en función de la eficiencia del proceso de desarrollo de SI y la efectividad esperada de sus resultados [Rojas y Pérez, 1995]. Ahora bien, el proceso de selección de una herramienta CASE no es fácil, dado lo amplio del mercado y lo poco objetivos que son los vendedores de estas herramientas al dar sus características. [Topper et. al., 1994]

Según [Mendoza, 1999], el uso de herramientas CASE es visto, por una parte, como una solución para los mayores problemas relacionados con las formas tradicionales de desarrollo de SI: demoras, errores, inconsistencias, dificultad para las pruebas e inadecuada documentación [Nelson y Rottman, 1996], y por otra parte, como un soporte para el desarrollo de sistemas de calidad a través de la automatización del ciclo de vida de desarrollo. [Chau, 1996] Al adoptar una herramienta CASE, las organizaciones tienen la gran expectativa de que ello implica grandes mejoras en la calidad y velocidad en el proceso de desarrollo de los SI [Iivari, 1996]. Un estudio de mercado sobre herramientas CASE realizado a profesionales de las TI, en 1996, arrojó que el 70% de los encuestados mostró un alto interés en las herramientas CASE y que, en promedio, planean emplear el 24% de su presupuesto en la adquisición de herramientas CASE. [Chau, 1996].

A pesar de que hay muchas referencias sobre el incremento de la productividad del software con el uso de herramientas CASE, se han dado casos en los cuales los resultados no han sido tan satisfactorios como se esperaban. De acuerdo con McClure [McClure, 1993], las causas que han motivado resultados no positivos en la adopción de herramientas CASE, se corresponden con algunos de los siguientes aspectos: confusión sobre lo que realmente hace cada producto CASE individualmente, el uso de herramientas CASE en problemas para las que no están diseñadas, poner demasiado empeño en las herramientas CASE como solución completa, ignorar la importancia de una buena gestión, no tener desarrollados estándares o metodologías en la organización,

herramientas CASE pobremente integradas, herramientas pobres en documentación y formación, sin funcionalidad suficiente en las herramientas CASE, falta de claridad sobre que problema del software necesita solución, falta de métodos para medir el impacto de la herramienta CASE en el desarrollo y mantenimiento del software, falta de formación en la metodología del desarrollo del software, indecisión y/o falta de predisposición para tomar una decisión sobre cómo utilizar la tecnología CASE, falta de predisposición para cambiar de modo de trabajo en el desarrollo y mantenimiento del software, ver la herramienta CASE como una tecnología de alto riesgo, inexistencia de una planificación para implantar la herramienta CASE. McClure concluye diciendo que las herramientas CASE son una parte importante de la solución de la crisis del software pero, no debe considerarse como la solución total. El simple hecho de comprar unas herramientas CASE no es condición suficiente para garantizar un efecto importante en la mejora global de la productividad y la calidad del software. Sin embargo, una solución total a la crisis del software debe verse a través de múltiples facetas: gestión, personal, herramientas y metodologías.

Más tarde, Iivari [Iivari, 1996] realiza un estudio donde indica que después del primer año de haberse incorporado una herramienta CASE, el 70% de las herramientas no fueron usadas, el 25% sólo fue usada por un grupo y el 5% restante no aprovechó el total de su capacidad. Además, afirma que las investigaciones realizadas indican que el uso de las herramientas CASE varía considerablemente de organización en organización, sin ofrecer una orientación teórica o una explicación tentativa de los factores que afectan la adopción.

A continuación se presentan las conclusiones de dos de las investigaciones académicas más recientes y publicadas, relacionadas con estudios realizados sobre los factores que afectan la adopción de herramientas CASE en las organizaciones que desarrollan SI en Venezuela.

La primera investigación, realizada por Vilorio y Rojas [Vilorio y Rojas, 1996], presenta un trabajo motivado por “los problemas generados alrededor del proceso de asimilación de las herramientas CASE en las organizaciones venezolanas y el deseo de otras organizaciones de adoptarlas”. Esta investigación muestra los resultados obtenidos al aplicar una encuesta a personal especializado en herramientas CASE y que labora en organizaciones venezolanas ubicadas en la región capital.

Los aspectos evaluados se relacionan con la productividad de los analistas y la calidad de los SI desarrollados con estas herramientas, la actitud hacia el uso de ellas, los cambios ocurridos en la

estructura organizacional después del proceso de implementación de esta tecnología y el nivel de madurez de las organizaciones desarrolladoras de SI.

Tomando como base el modelo de Meyer y Goes, [Meyer y Goes, 1988] citado en [Viloria y Rojas, 1996], Viloria y Rojas consideran que 6 variables son críticas para el éxito de la asimilación de las herramientas CASE (ver Tabla N° 2):

Variable	Definición
Personal de SI	La orientación que tienen los desarrolladores de SI a los resultados y a la búsqueda funcional de los sistemas; en caso de estar orientados solamente al procesamiento de datos y a la búsqueda de la eficiencia del sistema, se limitará el éxito de las herramientas CASE.
Cultura de SI	Grado de territorialismo existente dentro de la organización que desarrolla SI; en ésta, no debe existir territorialismo; la responsabilidad debe ser compartida y no mantenida por los analistas, y se le deben delegar controles a los usuarios.
Participación y apoyo de la gerencia senior	Defensa que los gerentes senior hacen a las herramientas CASE, gastando esfuerzos personales en la influencia y adquisición de recursos, como a la promoción que los gerentes senior le dan a la innovación.
Metodología de desarrollo	Grado de compatibilidad que existe entre la metodología de desarrollo de SI que se tiene en la organización que desarrolla SI, y la herramienta CASE
Participación del usuario	Rol que desempeñan los usuarios con respecto al uso de las herramientas CASE
Implementación estratégica	Proceso seguido por las organizaciones que desarrollan SI para facilitar la asimilación de las herramientas CASE

Tabla N° 2. Definición de las variables propuestas por Viloria y Rojas.

Fuente: [Viloria y Rojas, 1996]

Una de las conclusiones de esta investigación, establece, que en Venezuela existe una actitud positiva hacia el uso de las herramientas CASE; según la opinión de los analistas, las herramientas CASE aumentan la productividad y la calidad de los sistemas desarrollados con su uso; en la industria petrolera venezolana se encuentran los analistas con mayor disposición síquica hacia el uso de las herramientas CASE; las herramientas son más usadas para el desarrollo de SI que para el mantenimiento de SI; y se requieren de cambios en la estructura organizacional de las empresas para incorporar las herramientas CASE, sobre todo en aquellas que han alcanzado un nivel de madurez óptimo.

La segunda investigación que se hace referencia, es la realizada por De Freitas [De Freitas, 1997], recoge la opinión que tienen los tomadores de decisiones de los niveles gerenciales alto y medio de las organizaciones que han pasado por el proceso de evaluar las herramientas CASE y algunas características organizacionales de los departamentos de desarrollo de SI que las utilizan.

De Freitas [De Freitas, 1997] expone que debido a que no existe un modelo o hipótesis única de la asimilación y uso de las herramientas CASE, en el estudio exploratorio se recogieron algunas variables utilizadas por diferentes investigadores, las cuales pueden ser clasificadas en tres factores llamados atributos: a) Atributos de innovación, b) Atributos del contexto organizacional, y c) Atributos que surgen de la interacción entre la innovación y el contexto organizacional (ver Tabla N° 3)

Factores	Variables
Atributos de Innovación	Riesgo de usar o incorporar la tecnología (Miedo). Destreza / entrenamiento (Aptitud). Observación / Calidad. Costo. Soporte por parte del vendedor.
Atributos del Contexto Organizacional	Tamaño del Departamento de SI. Tipo de organización. Tipo de Datos que procesan. Complejidad del Producto Características de los empleados. Orientación del personal de SI Territorialismo del Departamento de SI (Cultura) Backlog de Aplicaciones. Outsourcing / insourcing. Tecnología que soporte la innovación.
Atributos que surgen de la Interacción de innovaciones y el Contexto organizacional	Compatibilidad de la metodología con las herramientas CASE. Apoyo por parte de la gerencia senior. Apoyo por parte de la alta gerencia. Involucrar al usuario. Estrategia de implementación. Reajuste organizacional.

Tabla N° 3. Factores asociados a la implementación de las herramientas CASE en las organizaciones venezolanas
Fuente: [De Freitas, 1997]

A continuación se da una breve descripción de las 21 variables recopiladas por De Freitas (ver Tabla N° 4).

Variable	Definición
Riesgo de usar o incorporar la tecnología (Miedo).	Se refiere al grado de resistencia al cambio.
Destreza / entrenamiento (Aptitud).	Se refiere a la “condición o serie de características consideradas como síntomas de la capacidad de un individuo para adquirir, con un entrenamiento adecuado, habilidad o serie de reacciones”, (Diccionario de Psicología, 1970) citado en [De Freitas, 1997].
Observación / Calidad	Se refiere a la medición del impacto de las herramientas CASE en el desempeño y calidad de los sistemas desarrollados.
Costo	Se refiere a que “El costo de la innovación incluye el costo inicial de la inversión, el costo del entrenamiento y el costo de operación de la herramienta CASE, facilitando el uso efectivo de la innovación”, [Premkumar y Potter, 1995] citado en [De Freitas, 1997].
Soporte por parte del vendedor	Se refiere al grado de soporte que brinda el vendedor de la herramienta CASE después de su adquisición
Tamaño del Departamento de SI	Se refiere a la cantidad de personas que trabajan en la organización que

	desarrolla SI.
Tipo de organización	Se refiere al tipo de organización que caracteriza a la organización que desarrolla SI: centralizada, descentralizada o mixta.
Tipo de Datos que procesan	Se refiere a la relación con el tipo de datos se procesan en la organización de SI.
Complejidad del Producto	Se refiere a “el grado por el cual la innovación es percibida como una dificultad para ser entendida y usada”, [Premkumar y Potter, 1995] citado en [De Freitas, 1997].
Características de los empleados	Se refiere al grado de experiencia de los analistas de SI.
Orientación del personal de SI	Se refiere a la orientación que tienen los desarrolladores de SI a los resultados y a la búsqueda funcional de los sistemas.
Territorialismo del Departamento de SI (Cultura).	Se refiere al grado de resistencia al cambio que presenta la organización que desarrolla SI en cuanto a la innovación de las herramientas CASE.
Backlog de Aplicaciones	Se refiere a la cantidad de SI por desarrollar acumulado; “los artículos publicados involucran discusión acerca del potencial de las CASE, para reducir y eliminar backlogs de aplicaciones, mejorar la calidad de los sistemas y aumentar la productividad de los analistas”, [Friesen y Orlikowski, 1989] citado en [De Freitas, 1997].
Outsourcing / insourcing	Se refiere a un mayor uso de software desarrollado externamente y menos desarrollado internamente.
Tecnología que soporte la innovación	Se refiere a los recursos adicionales de hardware y software que deben adquirirse para soportar a las herramientas CASE.
Compatibilidad de la metodología con las herramientas CASE	Se refiere al grado de compatibilidad que existe entre la herramienta CASE y la metodología de desarrollo utilizada por la organización que desarrolla SI.
Apoyo por parte de la gerencia senior	Se refiere al respaldo que dan los gerentes senior a la incorporación de las herramientas CASE en el proceso de desarrollo de SI.
Apoyo por parte de la alta gerencia	Se refiere al respaldo que da la alta gerencia a la incorporación de las herramientas CASE en el proceso de desarrollo de SI.
Involucrar al usuario	Se refiere al rol que desempeñan los usuarios de las herramientas CASE en la adopción de éstas.
Estrategia de implementación	Se refiere a las estrategias que establecen las organizaciones que desarrollan SI para asimilar las herramientas CASE.
Reajuste organizacional	Se refiere a la capacidad de adaptación que tiene la organización que desarrolla SI para hacer los reajustes organizacionales necesarios por efecto de la adopción de las herramientas CASE.

Tabla N° 4. Definición de las Variables propuestas por De Freitas.

Fuente: [De Freitas, 1997]

Los hallazgos de esta investigación determinan que, la incorporación de las herramientas CASE en las organizaciones venezolanas es exitosa y medianamente exitosa; existe un gran uso de las herramientas CASE; las empresas que utilizan las herramientas son relativamente jóvenes en su uso; y los analistas que laboran en el departamento de SI valoran intrínsecamente su trabajo, la misión de la organización, y además, participan y se identifican con la organización.

Al igual que Viloría y Rojas [Viloría y Rojas, 1996], De Freitas [De Freitas, 1997] presenta un análisis de los factores que afectan la adopción de las herramientas CASE en Venezuela muy parecido al anterior; sin embargo, puede observarse que además de aumentar significativamente las variables del estudio, la mayoría de éstas, más del 76%, están referidas al contexto organizacional y su relación con las innovaciones tecnológicas. Por otro lado, también resalta que las conclusiones del estudio están más referidas a condiciones de agrado y satisfacción de los analistas de SI frente a

la organización que adopta las herramientas CASE. En este sentido, lo anterior permite afirmar que cada día está tomando mayor importancia la relación que existe entre las herramientas CASE y el entorno organizacional que la adoptará y la usará.

Más tarde Jarzabek y Huang [Jarzabek y Huang, 1998], realizan un estudio donde han analizado los problemas que impiden la adopción de las herramientas CASE y han sugerido propuestas para aliviar algunos de los problemas. El principal aspecto radica en que las herramientas CASE actuales están centradas en los métodos y estos no son atractivos a los usuarios, lo mejor para los procesos de desarrollo, sería que las herramientas CASE estén orientadas al usuario y soportar los aspectos creativos para la solución de problemas del desarrollo de software así como también el modelamiento riguroso. Estas herramientas deberían también proporcionar una estructura de desarrollo orientada al proceso natural más que orientadas a métodos y jugar un rol más activo en el proceso de software que las actuales herramientas CASE. Estos autores concluyen que el futuro de las herramientas CASE debería estar basado sobre modelos sólidos de un proceso de software y los comportamientos de usuario.

Como puede verse, el uso de las herramientas CASE en las organizaciones tiene su justificación: **éstas tienen como objetivo el mejoramiento del proceso de desarrollo de los SI.** Ahora bien, ¿por qué las empresas que desarrollan SI no las han adoptado en la medida que se esperaba?, o las que las han adquirido, ¿por qué no han logrado mejorar el proceso de desarrollo de los SI?.

2.1.6 FACTORES QUE AFECTAN LA ADOPCIÓN Y USO DE HERRAMIENTAS CASE.

Hoy en día, una inmensa cantidad de proyectos de desarrollo de SI utilizan herramientas CASE, las cuales se usan en conjunto por distintos miembros del equipo de desarrollo. Sin embargo, la decisión de adoptar una nueva o más avanzada herramienta para un proyecto dado está basada en el conocimiento intuitivo de los beneficios esperados. [Premkumar y Potter, 1995]. En este sentido, Kavi y Nahouraii [Kavi y Nahouraii, 1996] indican que *"la calidad del software puede verse mejorada por una correcta selección de la herramienta que apoye cada fase del proceso de desarrollo, desde el análisis de requerimientos hasta las pruebas finales y la integración. La selección de una herramienta inapropiada, por el contrario, puede impedir el buen desarrollo del software"*.

La inserción de una nueva herramienta usualmente tiene altos costos; éstos incluyen: el costo de adquisición, el de entrenamiento de los miembros del proyecto para usar la herramienta, el de

conversión de los diseños y datos existentes de un formato a otro, el de volver a dar entrada a los datos de los diseños existentes a los nuevos diseños y, el de escribir apropiados programas y filtros para las conversiones entre diferentes herramientas y formatos [Kavi y Nahouraii, 1996]. De allí, que los gerentes de proyectos de desarrollo de los SI se tornen renuentes a considerar una(s) nueva(s) herramienta(s), a menos que los beneficios de su adopción puedan ser evaluados y presentados de una manera convincente.

Todas las herramientas, viejas o nuevas, pueden dar soporte al desarrollo de los SI. Sin embargo, **los beneficios que una nueva herramienta puede tener sobre la productividad o calidad del software resultante es difícilmente medible**. La dificultad de evaluar estos aspectos se debe a la gran variedad de sistemas que pueden ser desarrollados usando dichas herramientas. *“No es fácil diseñar un conjunto simple de criterios que puedan ser usados para evaluar todas las herramientas y ambientes de desarrollo”*. [Kavi y Nahouraii, 1996]

A continuación se analizarán las investigaciones más actualizadas en torno a los factores que afectan drásticamente la adquisición, adopción y uso de las herramientas CASE por parte de las organizaciones desarrolladoras de Sistemas de Información.

De acuerdo con McClure [McClure, 1993], las organizaciones que necesitan resolver su crisis de software inmediatamente; necesitan desarrollar sistemas de mejor calidad en mucho menos tiempo y con un costo mucho menor, además, necesitan reducir la carga de mantenimiento de los sistemas, que es una de las principales causas de los retrasos en el proceso de desarrollo. Por otro lado, cada organización sufre su propia crisis de software, lo que conduce a que no exista una solución universal. Para este autor, la solución práctica debe hacerse a la medida de cada organización, lo que significa que cada una debe empezar por especificar y asignar prioridades detalladas y ajustadas a sus necesidades. Simplemente decir, que se necesita aumentar la calidad y la productividad del software no es una especificación adecuada para determinar una solución práctica para adquirir una herramienta CASE. McClure propone una serie de aspectos técnicos y de gestión, ya que deben comprenderse ambos aspectos para especificar totalmente las necesidades. Entre los aspectos de gestión que propone McClure están los siguientes: aceptación de la adquisición de la herramienta CASE, de la formación y de un apoyo continuado, establecimiento de objetivos y presupuestos, aceptación de la tecnología CASE por la dirección y por los profesionales del desarrollo y del mantenimiento, creación de un grupo de implantación y de soporte continuado de la herramienta CASE, asignación de responsabilidades para la investigación, implantación y evaluación de la

herramienta CASE, definición de los controles de gestión, obligatoriedad de los estándares, creación de los programas de formación sobre la herramienta CASE y sobre las metodologías, y la creación de una planificación de tiempos para los pasos necesarios del plan de implantación de la herramienta CASE. Dentro de los aspectos técnicos a considerar están los siguientes: selección de la metodología de desarrollo, definición de los estándares técnicos para interfaces con la herramienta, sistemas gestores de bases de datos, lenguajes de cuarta generación, diccionarios y los controles de calidad del software, definición de las capacidades de la herramienta CASE, definición de las necesidades del hardware y de las redes de área local, selección de las herramientas CASE que satisfagan las metodologías, estándares y procedimientos de la organización, definición de la formación técnica necesaria, definición de las necesidades de soporte técnico para las herramientas y metodologías CASE, definición de los tipos de sistemas a desarrollar y mantener con las herramientas CASE, realización del proyecto piloto de la CASE, y la gestión de la configuración de la herramienta CASE. Nótese que de la propuesta de McClure se desprenden dos fuertes componentes que intervienen en la adopción de una herramienta CASE, como lo son aspectos de gestión, propios de la organización que desea adquirir la CASE y por otro lado un grupo de aspectos técnicos que dependen de la herramienta y de las aplicaciones que se deseen desarrollar con dicha CASE.

Por otro lado, Orlikowski [Orlikowski, 1993], realiza una investigación en la cual se presentan las experiencias que dos organizaciones tuvieron en la adopción y el uso de herramientas CASE a lo largo del tiempo, usando fundamentalmente un enfoque teórico de investigación. El estudio caracteriza la experiencia de las organizaciones en términos de procesos de cambio organizacional, tanto incremental como radical. Los logros alcanzados fueron usados para desarrollar un esquema teórico que **conceptualiza los aspectos organizacionales que giran en torno a la adopción y el uso de las herramientas CASE.**

Específicamente, el esquema teórico y los hallazgos sugieren que los investigadores deben considerar el contexto social del desarrollo de sistemas, las intenciones y las acciones de los individuos claves (desarrolladores, analistas, jefes de proyectos, gerentes, etc.) y el proceso de la implementación de la herramienta CASE seguido por la organización. De manera similar, el estudio sugiere que *“los profesionales del desarrollo de Sistemas de Información pueden gerenciar mejor las experiencias de las organizaciones con las herramientas CASE, si ellos entienden que la implementación de las herramientas CASE conlleva a un proceso de cambio organizacional a lo*

largo del tiempo y no solamente la simple instalación de una nueva tecnología”. [Orlikowski, 1993].

En primer lugar, Orlikowski plantea el proceso de cambio organizacional entorno a las herramientas CASE a través de la Figura N° 1. Este proceso fue propuesto como una formulación inicial de las categorías, los conceptos claves que emergieron del análisis de los datos y, su interacción como consecuencia del proceso de adopción y uso de herramientas CASE visto como un proceso de cambio organizacional.

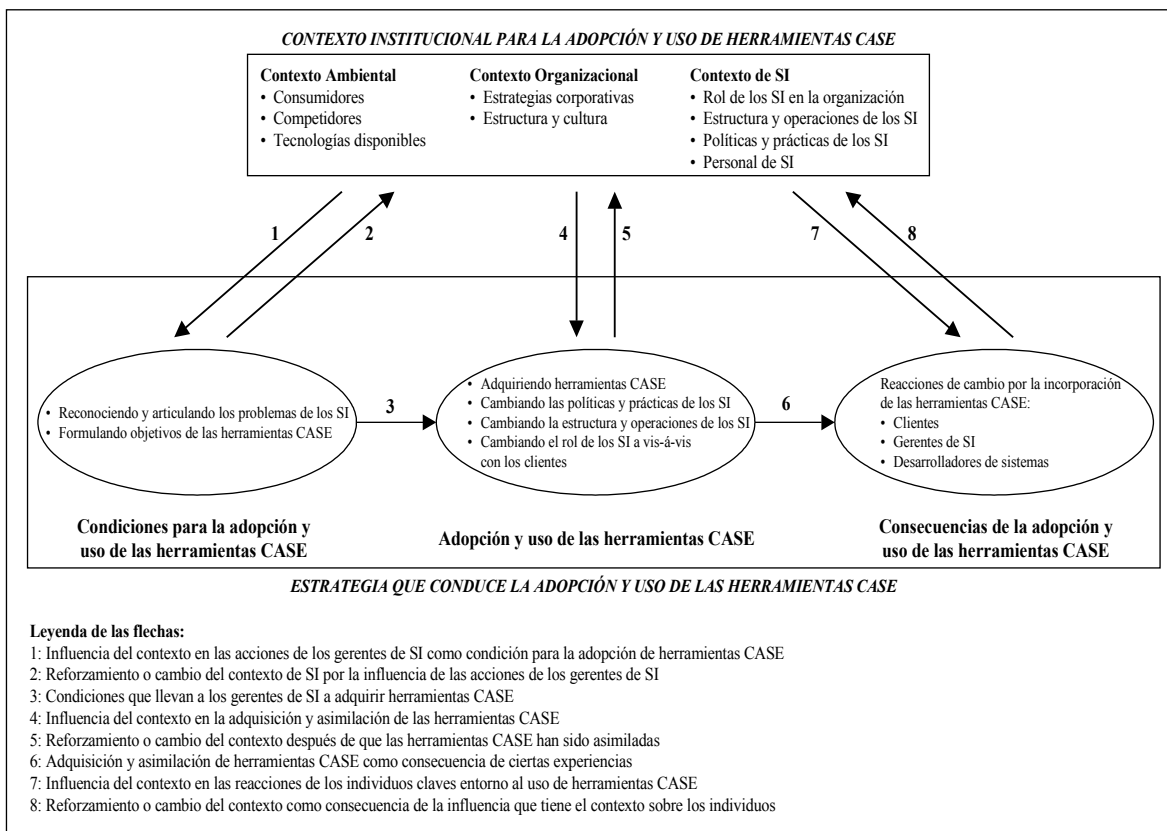


Figura N° 1. Proceso de cambio organizacional en torno a las herramientas CASE.

Fuente: [Orlikowski, 1993]

En segundo lugar el autor utiliza los conceptos de cambio incremental –representa una extensión del status quo, es decir, ajustes o refinamientos de los productos, las prácticas, las relaciones, las habilidades y las normas- [Orlikowski, 1993], y cambio radical –cuestionamiento del status quo, requiriendo un cambio fundamental de los productos, las prácticas, las relaciones, las habilidades y las normas- [Orlikowski, 1993], para caracterizar las diferentes intenciones y acciones que se pueden tomar entorno a las herramientas CASE (ver Figura N° 2).

		CENTRO DEL CAMBIO	
		Proceso del desarrollo de sistemas	Producto del desarrollo de sistemas
NATURALEZA DEL CAMBIO	Cambio incremental	Variación del proceso	Variación del producto
	Cambio radical	Reorientación del proceso	Reorientación del producto

Figura N° 2. Naturaleza y centro del cambio asociado con las herramientas CASE.
Fuente: Adaptado de [Orlikowski, 1993]

Como resultado del estudio, Orlikowski, [Orlikowski, 1993] expresa que los procesos representados en la Figura N° 1 y la naturaleza y centro del cambio asociado a las herramientas CASE dada en la Figura N° 2, proveen una manera de examinar los cambios organizacionales que están en torno a las herramientas CASE. Tomando en cuenta el contexto organizacional existente, las intenciones de los gerentes de SI y las acciones a ejecutar en torno a las herramientas CASE, la naturaleza y el centro del cambio asociado con las herramientas CASE pueden ser anticipados, explicados y evaluados, así como las consecuencias de esos cambios. El esquema desarrollado no implica que las herramientas CASE por si solas, causen variaciones del proceso y del producto, o la reorientación del proceso y del producto; “los cambios organizacionales emergen de la particular interacción del contexto institucional, las intenciones y acciones de los individuos claves y las herramientas CASE” [Orlikowski, 1993].

Por otra parte, al analizar un trabajo realizado por Premkumar y Potter [Premkumar y Potter, 1995], se observa **el impacto que tienen varias características organizacionales y tecnológicas en la adopción de herramientas CASE**. En este sentido, se desarrolló un modelo de investigación (ver Figura N° 3) que identifica los aspectos más importantes para la adopción de herramientas CASE.

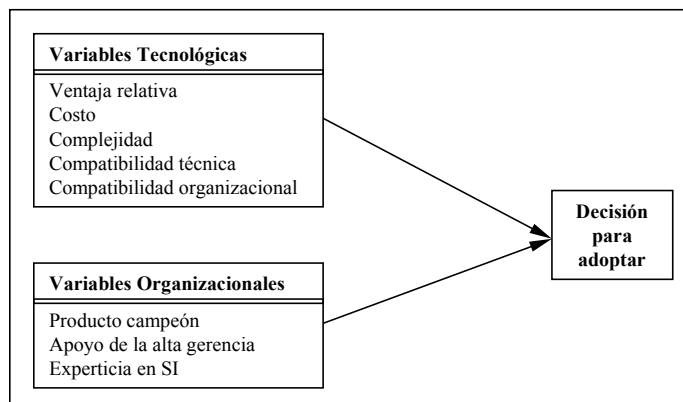


Figura N° 3. Modelo de investigación y Adopción de Herramientas CASE
Fuente: [Premkumar y Potter, 1995]

A continuación se da una breve descripción de las ocho variables que conforman el modelo (ver Tabla N° 5):

Variable	Definición
Ventaja relativa.	Se refiere al grado con el cual la innovación es percibida como la mejor.
Costo.	Se refiere a lo poco costoso de la innovación. Incluye tanto el costo inicial de la inversión como los costos asociados a la operación y el entrenamiento que facilitan el uso efectivo de la innovación.
Complejidad.	Basado en la definición dada por Rogers, [Rogers, 1983] citado en [Premkumar, 1995], se refiere a “el grado con el cual una innovación es percibida como relativamente difícil para entender y usar”.
Compatibilidad técnica.	Se refiere a la compatibilidad con las diferentes plataformas de software y hardware, y otras herramientas CASE, existentes en la organización.
Compatibilidad organizacional.	Basado en la definición dada por Rogers, [Rogers, 1983] citado en [Premkumar, 1995], se refiere al “grado con el cual una innovación es percibida como bien consistente con los valores existentes, las experiencias pasadas y las necesidades de los adoptadores potenciales”.
Producto líder.	Se refiere al producto que educa a los gestores de proyectos y a los usuarios sobre los beneficios potenciales de la tecnología y crea conciencia del papel de la tecnología y lo necesario que es para la organización; además, coordina varias actividades relacionadas con la adquisición de tecnología y asegura la dotación de recursos adecuados para la adopción e implementaciones futuras.
Apoyo de la alta gerencia.	Se refiere a la certeza de que los recursos adecuados serán dotados si la innovación es adoptada; es la apropiada visión estratégica que se le da a la importancia de la innovación.
Experticia en SI.	Se refiere a la existencia de experiencia en el desarrollo de SI dentro de la organización, en términos de la familiaridad con metodologías estructuradas, técnicas de modelaje de procesos y de bases de datos, que crean un clima favorable para la fácil introducción de las herramientas CASE.

Tabla N° 5. Definición de las variables.

Fuente: [Premkumar y Potter, 1995]

Los resultados del análisis de los datos revelaron que, de las ocho variables, la existencia de un producto líder, un fuerte apoyo de la alta gerencia, una baja experticia en SI, la percepción de que la herramienta CASE tiene mayor ventaja relativa por encima de otras alternativas, y la convicción de la efectividad del costo de la tecnología, fueron encontrados como las variables más importantes para diferenciar las organizaciones que adoptan las herramientas CASE de las que no las adoptan. [Premkumar y Potter, 1995]

Lo más resaltante que presentan Premkumar y Potter, es que su modelo de investigación distingue los tipos de variables que tienen impacto en la adopción de herramientas CASE: las variables tecnológicas y las variables organizacionales. Esto permite concluir que las características propias de la herramienta CASE son tan importantes como las características que tiene la organización para adoptar la herramienta CASE, es decir, que la adopción de una herramienta CASE depende directamente de las condiciones que presenta el ambiente organizacional donde ésta se va a utilizar y de las características tecnológicas propias de la herramienta.

Posteriormente, Iivari [Iivari, 1996] presenta un modelo conceptual (ver Figura N° 4) donde **identifica ocho factores que potencialmente afectan el uso de las herramientas CASE en las organizaciones** (ver Tabla N° 6):

Variable	Definición
Participación.	Se refiere a la participación de los profesionales de SI y de software en la selección de herramientas CASE y en la planificación de la implementación organizacional (adopción).
Apoyo gerencial.	Se refiere a las facilidades para la implantación de SI. En este sentido, implantación de SI tiene dos significados, primero, el control gerencial sobre los recursos necesarios para el desarrollo de sistemas y, segundo, el comportamiento gerencial que procuran la participación.
Entrenamiento.	Se refiere a la calidad y cantidad de entrenamiento para el uso de las herramientas CASE.
Expectativa realista.	Se refiere a la diferencia entre el impacto percibido por el uso de la herramienta CASE y las expectativas dadas al impacto esperado de la herramienta CASE.
Complejidad.	Se refiere al grado de percepción de lo difícil que es entender y usar una innovación.
Compatibilidad.	Se refiere al grado de percepción que se tiene de la consistencia entre los valores actuales, las experiencias pasadas y las necesidades de los potenciales usuarios de la herramienta CASE.
Ventaja relativa.	Se refiere al grado de percepción que se tiene de los beneficios de usar la herramienta.
Voluntariedad de uso.	Se refiere al grado con el cual una innovación es usada espontáneamente y no de manera impositiva. En este sentido, el uso de la herramienta CASE depende de si los usuarios quieren utilizarla o no, más que por una política expresa de la organización que desarrolla SI.

Tabla N° 6. Definición de las variables.

Fuente: [Iivari, 1996]

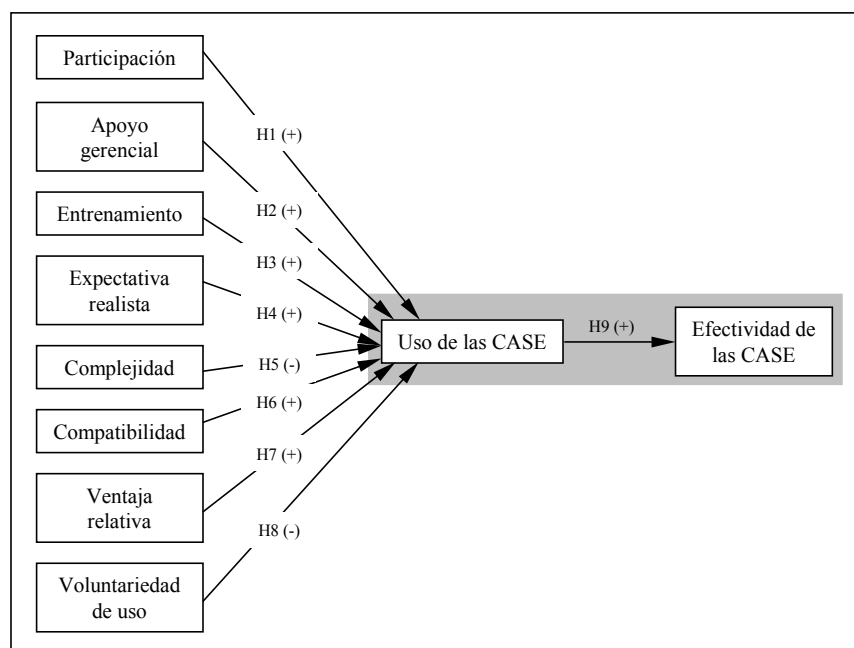


Figura N° 4. El modelo conceptual

Fuente: [Iivari, 1996]

Además de los factores descritos anteriormente, Iivari [Iivari, 1996] afirma que un área importante abandonada en la investigación sobre SI y herramientas CASE, es la relacionada con el uso de las herramientas CASE como un factor que afecta directamente la efectividad de las herramientas CASE. Este autor indica que existe una relación entre el uso de los SI y la efectividad de los SI, cuando la efectividad de los SI es medida como el impacto que tienen los SI sobre la efectividad de la organización, es decir, sobre la productividad y la calidad de los SI.

El modelo propuesto por Iivari [Iivari, 1996] en la Figura N° 4 también sugiere las hipótesis sobre los factores que afectan el uso y efectividad de las herramientas CASE en las organizaciones (el signo + indica una asociación positiva y el signo – indica una asociación negativa). El análisis del modelo se basó en regresiones múltiples en dos fases: en la primera fase, el uso de las herramientas CASE representa la variable dependiente y los 8 factores citados anteriormente representan las variables independientes; en la segunda fase, los dos factores que indican la efectividad de la herramienta CASE (productividad y calidad) representan las variables dependientes y el uso de la herramienta CASE representa la variable independiente.

Entre las conclusiones del estudio, está que los factores tales como: soporte gerencial, ventaja relativa y voluntariedad de uso, fueron encontrados como los más importantes para predecir el uso de las herramientas CASE; además, concluye que el uso de las herramientas CASE está altamente relacionado con la percepción que se tiene de la efectividad de las herramientas CASE. Por otro lado, *“una notable interpretación de los hallazgos del estudio es que el elevado uso de las herramientas CASE tiende a incrementar la productividad y calidad de los SI y del desarrollo de Software y, que la elevada libertad de uso de herramientas CASE tiende a mermar el uso de las herramientas CASE”*. [Iivari, 1996] Finalmente, Iivari indica que los resultados de la investigación están basados en usuarios de las herramientas CASE de un solo país, Finlandia, y que queda abierta la interrogante: ¿éstos resultados pueden ser generalizados a otros países?.

A diferencia de Premkumar y Potter [Premkumar y Potter, 1995], Iivari [Iivari, 1996] denomina ‘factores’ a lo que los primeros llamaron ‘variables’; sin embargo, son manejados de igual manera: **indican los aspectos que afectan el uso y adopción de las herramientas CASE**. Por otro lado, puede verse que hay ‘variables’ o ‘factores’ coincidentes: ventaja relativa, complejidad, compatibilidad y apoyo gerencial, independientemente de que Iivari no hace una distinción entre aspectos de tipo organizacional o aspectos de tipo tecnológico. Sin embargo, es posible señalar que de los factores coincidentes, ventaja relativa, complejidad y compatibilidad corresponden a aspectos de tipo tecnológico y sólo el apoyo gerencial corresponde a aspectos de tipo organizacional.

Paralelamente con Iivari, Chau [Chau, 1996] realiza una investigación que se enfoca más en la **aceptación de las herramientas CASE por parte de los desarrolladores de sistemas, ya que éstos son generalmente los usuarios de las herramientas CASE**. Propone un modelo de investigación (ver Figura N° 5) que es el resultado de la integración de otros dos modelos: el *modelo de aceptación tecnológica*, [David, 1989] citado en [Chau, 1996], y el *modelo de utilización de computación personal*, [Thompson et. al., 1991] citado en [Chau, 1996].

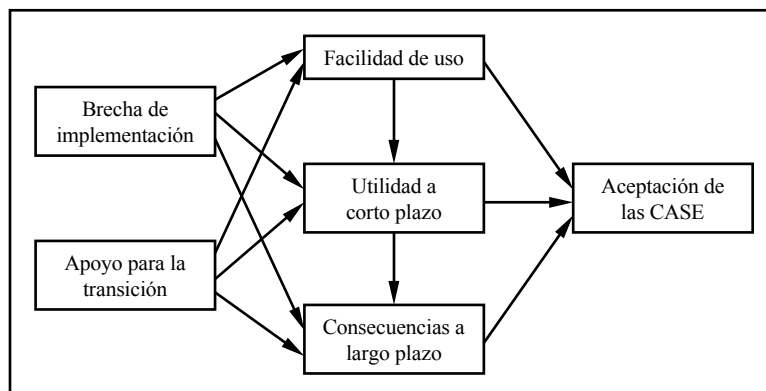


Figura N° 5. El modelo de investigación

Fuente: [Chau, 1996]

El modelo consiste en 5 factores que son planteados hipotéticamente según sus efectos directos o indirectos en la aceptación de las herramientas CASE; estos factores son (ver Tabla N° 7):

Variable	Definición
Brecha de implementación.	Se refiere a la brecha entre lo viejo y lo nuevo; a lo largo que puede ser el tiempo para que los individuos aprendan las habilidades y los conocimientos necesarios que les permitan adaptarse a los nuevos procedimientos de trabajo.
Apoyo para la transición.	Se refiere al apoyo suministrado por la gerencia durante el período de transición.
Facilidad de uso.	Se refiere a lo fácil que es usar la herramienta CASE.
Utilidad a corto plazo.	Se refiere al grado de percepción acerca de que la herramienta CASE pueda incrementar su desempeño en el trabajo.
Consecuencias a largo plazo.	Se refiere al grado de percepción de que la herramienta CASE incrementa la habilidad para cambiar de trabajo, o incrementa las oportunidades para realizar un trabajo con mayor relevancia para el desarrollador; este factor también incluye aspectos intangibles como el incremento del prestigio del desarrollador frente a sus colegas.

Tabla N° 7. Definición de las variables.

Fuente: [Chau, 1996]

Chau [Chau, 1996] expone que los resultados indican que la brecha de implementación tiene un impacto negativo tanto en la percepción de la facilidad de uso como la utilidad a corto plazo de las herramientas CASE. El adoptar una metodología estructurada junto con un soporte técnico apropiado, durante el período de transición, puede ayudar a reducir el impacto negativo de la brecha

de implementación. **Lo más certero es proveer entrenamiento formal en la metodología en que se basa la herramienta CASE y en el manejo de la herramienta CASE.** El estudio indica que la percepción de la utilidad a corto plazo de las herramientas CASE es un factor determinante en la percepción de las consecuencias a largo plazo del uso de las CASE, teniendo además un efecto significativo en la aceptación de las herramientas CASE. Esto sugiere que, si es posible, la gerencia debe asegurar que sus desarrolladores de sistemas tengan una firme creencia en la utilidad de las herramientas CASE en su trabajo diario. *“Desde el punto de vista del usuario, utilizar una nueva tecnología puede afectar no sólo la productividad diaria sino también las perspectivas profesionales”*. [Chau, 1996]

Aunque Chau [Chau, 1996] enfoca su investigación hacia la aceptación de las herramientas CASE por parte de los desarrolladores de SI (usuarios primarios de las herramientas CASE), más que hacia un enfoque que englobe a la organización y al desarrollador en conjunto -a diferencia de los autores anteriores, Premkumar y Potter [Premkumar y Potter, 1995] e Iivari [Iivari, 1996]-, no puede deslindar al desarrollador de SI de su entorno organizacional. Esto es tan cierto, que entre los indicadores (llamados por él, ‘factores’) pautados en su modelo, se incluye el *“Apoyo para la transición”*, que se refiere al *“Apoyo gerencial”* descrito anteriormente también por los investigadores Premkumar y Potter e Iivari, que a su vez, Premkumar y Potter [Premkumar y Potter, 1995] lo clasifican como un aspecto de tipo organizacional. Además, ligado a este ‘factor’, está el que denomina ‘Brecha de implementación’, el cual establece una gran carga de responsabilidad sobre la organización al momento de adoptar una herramienta CASE.

Por otro lado, Nelson y Rottman [Nelson y Rottman, 1996] analizaron 253 encuestas para esclarecer las siguientes interrogantes: **¿en qué se diferencian los usuarios de herramientas CASE de los que no las usan?** y **¿cuáles son las diferencias que presentan los métodos de trabajo de los desarrolladores de SI antes y después de la adopción de herramientas CASE?**. Para dar respuesta a estas interrogantes, el estudio se enfocó en cuatro áreas principales: las motivaciones y obstáculos para la adopción de herramientas CASE, una comparación entre usuarios y no usuarios de herramientas CASE, los métodos de trabajo de los desarrolladores de SI antes y después de la adopción de herramientas CASE y, el grado con el cual las herramientas CASE son usadas y apreciadas.

La más importante conclusión que arrojó este estudio es que **los principales obstáculos para la adopción de herramientas CASE, son los cambios de enfoque de desarrollo de SI asociados a**

su utilización y los altos costos de adquisición e implementación. Normalmente, después de la adopción de las herramientas CASE, los desarrolladores de SI experimentan menos problemas en análisis, mantenimiento, documentación y otras áreas del desarrollo de sistemas. Los usuarios de herramientas CASE gastan más tiempo durante las primeras fases del ciclo de vida (análisis y diseño) y menos durante las últimas fases del ciclo de desarrollo; esto probablemente mejora sus oportunidades de desarrollar SI exitosos. [Nelson y Rottman, 1996]

Independientemente de que Nelson y Rottman [Nelson y Rottman, 1996] no establecen una serie de ‘variables’ o ‘factores’ como las investigaciones analizadas previamente, sino áreas que engloban varios aspectos, los resultados indican que la utilización de herramientas CASE impactan aspectos organizacionales como los referidos a los cambios del enfoque de desarrollo de SI asociados al uso de la herramienta CASE y la implementación. Aunque ésta investigación difiere en metodología con respecto a las investigaciones anteriores, refleja que el uso de la herramientas CASE está influenciado notablemente por la organización que la adopta.

Paralelamente, Bruckhaus y sus colegas [Bruckhaus et. al., 1996], realizan un estudio con el objeto de **evaluar la hipótesis de que el tamaño de un proyecto y el modelo del proceso de desarrollo, pueden influenciar el efecto que tiene la inserción de una herramienta CASE para el desarrollo de un SI.** Ellos se interesaron en hallar si es posible adoptar un proceso más riguroso de desarrollo (el cual requiere normalmente más recursos) y al mismo tiempo incrementar la productividad, soportando el proceso con una herramienta CASE apropiada.

La unidad de análisis utilizada fue la inserción de una herramienta para el manejo de requerimientos dentro del proceso de planificación de requerimientos. Las sub-unidades de análisis fueron distintos procesos de desarrollo, tamaños de proyectos y tecnologías, y sus impactos en la productividad de SI. Estas últimas constituyeron las tres variables independientes y la productividad de SI fue la variable dependiente. En este sentido, las variables independientes quedaron definidas de la siguiente forma: [Bruckhaus et. al., 1996]. (ver Tabla N° 8):

Variable	Definición
Procesos.	La expectativa de que la inserción de la herramienta CASE para el manejo de requerimientos pueda cambiar el proceso en tres áreas principales: documentación de los requerimientos de información, acceso a los requerimientos de información y compartir la información de los requerimientos y del aseguramiento de la calidad.
Tamaño de los proyectos.	Los proyectos fueron medidos según la cantidad de nuevas funcionalidades esperadas por los planificadores para cada versión del producto.
Herramientas y tecnologías.	Para soportar los procesos de planificación de requerimientos.

Tabla N° 8. Definición de las variables.

Fuente: [Bruckhaus et. al., 1996]

Bruckhaus et. al. [Bruckhaus et. al., 1996] concluyeron que **la inserción de una herramienta CASE está justificada cuando se tiene previsto adoptar un proceso de desarrollo bien sofisticado y además, cuando se está preparado para realizar el esfuerzo que esto requiere.** Por otro lado, en base a los resultados del estudio, se formulan cuatro reglas tentativas: a) Lo poco complejo del proceso y lo largo del proyecto hacen altamente probable que la inserción de una herramienta CASE restrinja la productividad; b) Cuando se adopta un proceso más riguroso, se debe estar preparado para aumentar los recursos necesarios, aun cuando se esté mejorando con el apoyo de la herramienta CASE al mismo tiempo; c) El mejor desempeño de una herramienta CASE se alcanza cuando es usado en un proyecto de un tamaño apropiado, mientras que es menos productiva cuando es usada en proyectos más grandes o más pequeños; y d) En términos de esfuerzo, adoptar un proceso riguroso puede ser substancialmente menos costoso si también se incorporan las herramientas CASE apropiadas.

A diferencia de las investigaciones antes analizadas, Bruckhaus et. al. [Bruckhaus et. al., 1996] enfocan su investigación hacia el objeto central que tiene la influencia de una herramienta CASE dentro de una organización: **el proceso de desarrollo y la productividad.** Las ‘variables’ manejadas en ésta investigación permiten visualizar con mayor claridad el verdadero impacto que tiene una herramienta CASE cuando es adoptada por una organización; permite reconocer los aspectos de tipo organizacional que son realmente importantes a tomar en cuenta cuando se quiere incorporar una herramienta CASE como parte del trabajo diario del desarrollo de SI.

Por su parte, Valaer y Babb [Valaer y Babb, 1997] **proveen una lista de chequeo para seleccionar herramientas para el desarrollo de interfaces gráficas (Graphical User Interface -GUI-).** Los criterios propuestos para ayudar a la selección, son los siguientes: (ver Tabla N° 9):

Variable	Definición
Usabilidad.	Se refiere a si la herramienta es fácil de instalar, aprender y usar.
Funcionabilidad.	Se refiere a la satisfacción de las necesidades requeridas para la interfaz.
Flexibilidad.	Se refiere al soporte de diferentes metodologías de diseño, lenguajes de programación, plataformas de hardware y apoyo para múltiples idiomas.
Portabilidad.	Se refiere a las distintas plataformas que son soportadas por la herramienta y que tan fácil es migrar la interfaz entre distintas plataformas.
Soporte.	Se refiere a la documentación, entrenamiento y otros materiales provistos por el fabricante, así como también, la ayuda técnica.
Costo.	Se refiere a todos los costos relacionados a la adquisición y uso de la herramienta.

Tabla N° 9. Definición de las variables.

Fuente: [Valaer y Babb, 1996]

Estos autores indican que el trabajo presentado contempla solamente un instrumento para que los responsables de adquirir herramientas GUI tomen mejores decisiones. Sin embargo, el escoger la herramienta más apropiada para el desarrollo de SI no garantiza la efectividad del producto final; el usar la herramienta correcta refuerza el trabajo pero no reemplaza al buen diseñador ni a las técnicas de desarrollo. [Valaer y Babb, 1997] Por otro lado, los factores presentados, a excepción del costo, pueden ser clasificados como de tipo tecnológico.

Según Kavi y Nahouraii [Kavi y Nahouraii, 1996], los criterios usados por Valaer y Babb [Valaer y Babb, 1997], son muy parecidos a las siguientes características que contempla el estándar ISO 9126 para evaluar la calidad de productos de software: (ver Tabla N° 10):

Variable	Definición
Funcionabilidad.	Se refiere a si el producto exhibe las funciones requeridas para satisfacer ciertas necesidades en particular.
Usabilidad.	Se refiere al esfuerzo necesario para su uso.
Fiabilidad.	Se refiere a la capacidad del producto de mantener su nivel de desempeño.
Eficiencia.	Se refiere a los recursos necesarios para mantener el nivel requerido de desempeño.
Mantenimiento.	Se refiere al esfuerzo necesario para hacer cambios específicos al producto.
Portabilidad.	Se refiere al esfuerzo requerido para transportar el producto de un ambiente operativo a otro.

Tabla N° 10. Definición de las variables.

Fuente: [Kavi y Nahouraii, 1996]

El trabajo presentado por Valaer y Babb [Valaer y Babb, 1997], a diferencia de todas las investigaciones presentadas previamente, expone una lista de ‘criterios’ (que pueden ser entendidos como las denominaciones ‘variables’ o ‘factores’ utilizadas por los autores anteriores) los cuales son referidos más a la herramienta CASE en sí misma que a su incorporación dentro de la organización; además, el listado propuesto se refiere concretamente a herramientas para el desarrollo de interfaces gráficas (Graphical User Interface -GUI-). Sin embargo, dado el hecho de que el listado presentado tienen un parecido con las características del estándar ISO 9126 [Kavi y Nahouraii, 1996], es posible discriminar de éste, que la característica **eficiencia** es la que involucra directamente a la organización y el resto están referidos directamente a las características tecnológicas, de esto se puede concluir que para Valaer y Babb existe un fuerte ingrediente tecnológico con respecto a su influencia en la adopción de herramientas CASE.

Recientemente, Yang [Yang, 1999] presentó una investigación que tuvo por objetivo entender el uso de las herramientas CASE en Taiwan, basado en trabajos previos realizados en torno a la adopción e implementación de herramientas CASE. En este sentido, como resultado de la literatura

revisada (entre otras investigaciones, Yang consultó los estudios publicados por Orlikowski y, Premkumar y Potter) se elaboraron dos cuestionarios -uno para usuarios de herramientas CASE y el otro para instructores y vendedores de herramientas CASE-. Los modelos conceptuales de investigación son ilustrados en la Figura N° 6 y en la Figura N° 7.

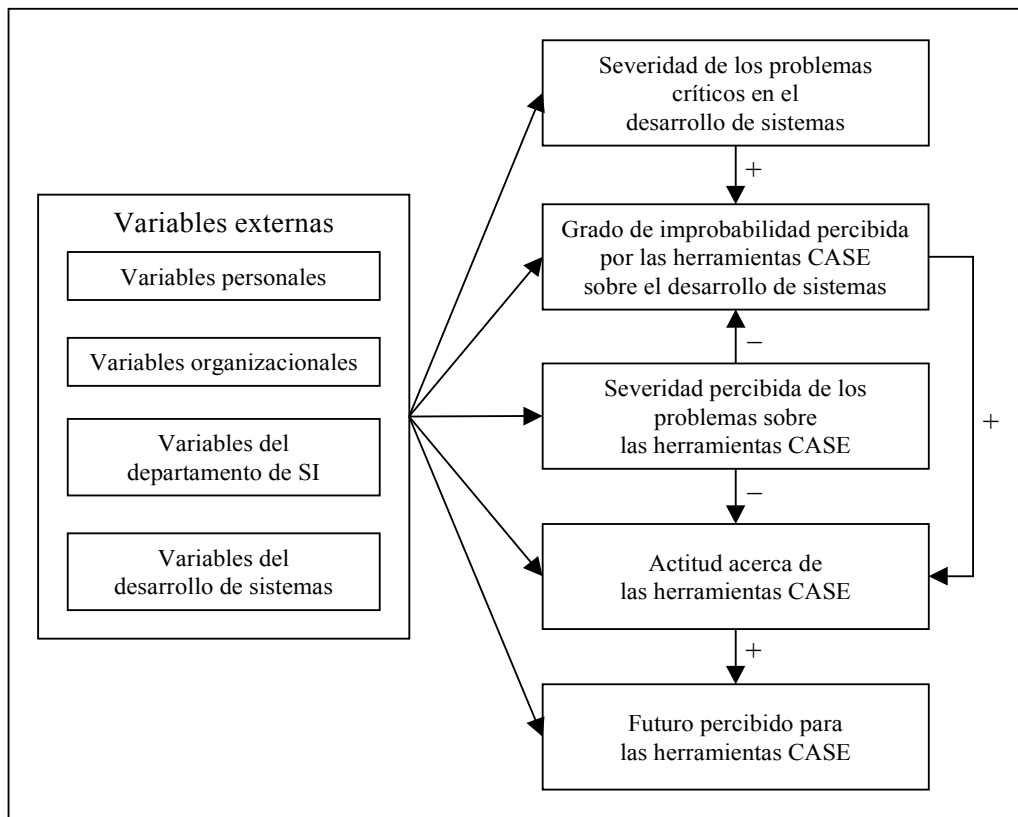


Figura N° 6. Modelo de investigación para la actitud hacia las herramientas CASE.

Fuente: [Yang, 1999]

Los signos + y - muestran la asociación hipotética, positiva o negativa, entre las variables internas. Con respecto a las variables externas, estas fueron agrupadas según se muestra en la Tabla N° 11.

Variable	Definición
Personales	Incluyen: posición, tiempo necesario para la realización de los trabajos de SI, nivel de especialidad para el análisis y diseño tradicional de sistemas sin la aplicación de herramientas CASE, nivel de especialidad en herramientas CASE y, preferencia hacia la planificación.
Organizacionales	Incluyen: tipo de organización, tipo de industria, tamaño de la organización y estado de desarrollo de los SI
Departamento de SI	Incluyen: tiempo de funcionamiento, cantidad de empleados de SI, cantidad de empleados de SI responsables del mantenimiento de sistemas, cantidad de empleados de SI con habilidades en herramientas CASE, uso del hardware de computación, porcentaje del presupuesto de SI gastado en ventas y porcentaje del gasto de desarrollo de software consumido del presupuesto de SI.
Desarrollo de Sistemas	Incluyen: outsourcing o no de SI, tamaño del proyecto de SI, tipo de metodología usada, uso o no de herramientas CASE, cantidad de herramientas CASE usadas, tiempo de uso de herramientas CASE y beneficios de las herramientas CASE usadas.

Tabla N° 11. Definición de las variables

Fuente: [Yang, 1999]

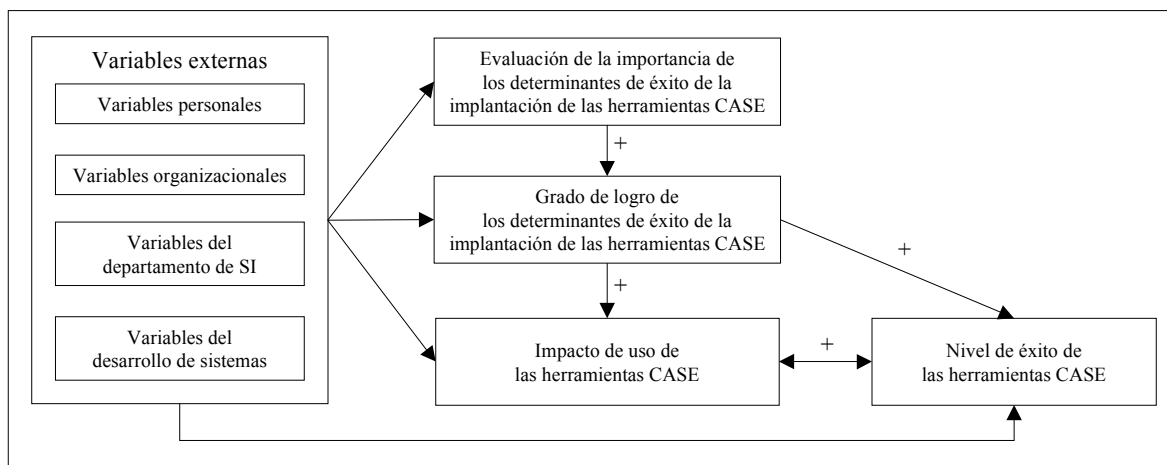


Figura N° 7. Modelo de investigación para el éxito de las herramientas CASE.

Fuente: Adaptación de [Yang, 1999]

Yang [Yang, 1999] afirma que mientras unas investigaciones concluyen que las herramientas CASE tienen beneficios en el mejoramiento de la productividad del desarrollo de SI, la calidad, el reuso del software, la documentación, el mantenimiento, etc., otras investigaciones advierten sobre el impacto negativo del uso de las herramientas CASE, la disminución de la habilidad y creatividad de los desarrolladores de SI, que amenazan la estabilidad en el trabajo induciendo así la resistencia. Por otro lado, también afirma que varias investigaciones sugieren que los factores claves para la adopción exitosa de las herramientas CASE incluyen: uso de una metodología, uso de un proyecto piloto, entrenamiento, guías de consulta y ayudas, demostraciones, soporte de la alta gerencia, uso de métricas de software, etc. [Yang, 1999]

Para finalizar, Yang [Yang, 1999] expresa que el resultado del análisis de los datos revela, de manera general, que todos los encuestados perciben problemas críticos en el desarrollo de SI y que tienen una actitud favorable acerca del impacto y el futuro de las herramientas CASE, aunque también perciben algunos problemas con su uso. Por otro lado, indica que la implementación exitosa de estas herramientas sólo depende significativamente de la ‘metodología aceptada’. Además, usar solamente la variable ‘metodología usada’ puede predecir el éxito relativo de los adaptadores de los no adaptadores de las herramientas CASE en un 75% de las veces. Sin embargo, ‘metodología usada’ no fue el factor más importante evaluado por los encuestados. [Yang, 1999].

Un último estudio a referenciar es el de Mendoza [Mendoza, 1999] quien realizó una investigación con el objetivo de proponer una lista de indicadores organizacionales que permitieran evaluar herramientas CASE en organizaciones Venezolanas, para el cual requirió hacer una revisión bibliográfica de los trabajos realizados por: Orlikowski [Orlikowski, 1993], Premkumar y Potter

[Premkumar y Potter, 1995], Iivari [Iivari, 1996], Chau [Chau, 1997], Bruckhaus et. al. [Bruckhaus et. al., 1996], Valaer y Babb [Valaer y Babb, 1996], Nelson y Rottman [Nelson y Rottman, 1996], Yang [Yang, 1999]. El modelo conceptual de investigación utilizado por Mendoza es el que se muestra en la Figura N° 8.

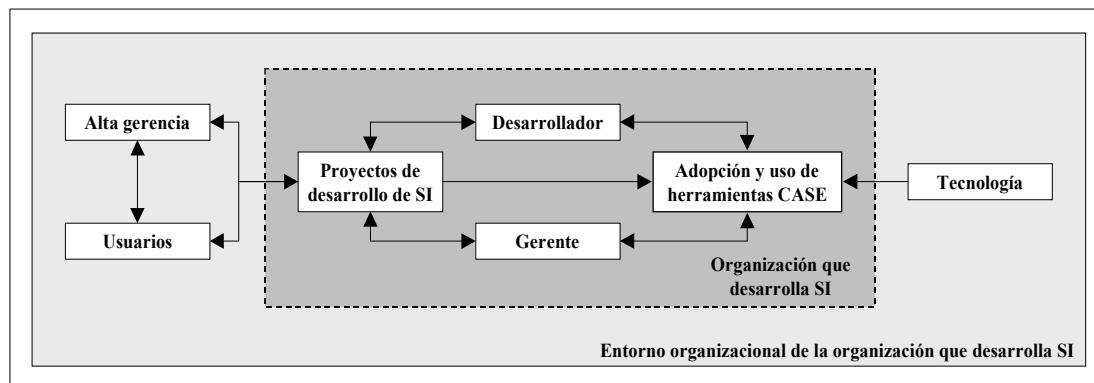


Figura N° 8. Proceso interactivo de adopción y uso de una herramienta CASE.

Fuente: [Mendoza, 1999]

Mendoza [Mendoza, 1999] resalta que la interacción que se da entre los usuarios, la alta gerencia y la unidad desarrolladora de SI, da origen a otros dos factores organizacionales fundamentales para la adopción y uso de herramientas CASE, pero que son externos porque su origen y su control no es exclusivo de la unidad desarrolladora de SI.

En la tabla N° 12, se presentan los factores organizacionales que permiten evaluar una herramienta CASE propuestos por Mendoza, [Mendoza, 1999].

Factores Externos	
<p>Imagen. Imagen que la unidad desarrolladora de SI refleja en su entorno; es decir, su importancia según el resto de la organización. Indicadores:</p>	<p>Impacto de los SI en la organización. Grado de impacto que tienen los SI que desarrolla la unidad desarrolladora de SI sobre la misión, objetivos y funcionamiento de la organización. Este indicador será medido a través de cinco (5) variables.</p> <p>Posición de la organización desarrolladora de SI en la estructura organizacional. Posición jerárquica de la unidad desarrolladora de SI dentro de la organización, en cuanto a si es considerada una unidad de servicio o una unidad estratégica para la organización. Este indicador será medido a través de tres (3) variables.</p> <p>Dependencia que la organización tiene de los SI para su productividad. Grado de dependencia que tiene la organización de los SI desarrollados por la unidad desarrolladora de SI para lograr sus metas y objetivos, y/o sobrevivir frente a sus competidores. Este indicador será medido a través de cinco (5) variables.</p>
<p>Corporación. Visión y postura que tiene la organización a la cual pertenece la unidad desarrolladora de SI frente al desarrollo de SI y a la adquisición y gerencia de innovaciones tecnológicas. Indicadores:</p>	<p>Apoyo de la alta gerencia. Apoyo que da la alta gerencia de la organización a los planes de mejoramiento tecnológico que lleva a cabo la unidad desarrolladora de SI, tanto a nivel interno como a nivel de toda la organización. Este indicador será medido a través de tres (3) variables.</p> <p>Resistencia a la innovación tecnológica. Grado de resistencia que tiene la organización para incorporar los adelantos e innovaciones tecnológicas, es decir, si es facilitadora o no de los procesos de adquisición de capacidades tecnológicas. Este indicador será medido a través de seis (6) variables.</p>

	<p>Backlogs de aplicaciones. Cantidad de aplicaciones o SI que la unidad desarrolladora de SI tiene acumulados. Este indicador será medido a través de seis (6) variables.</p> <p>Visión estratégica de los SI. Grado de relevancia que la organización le otorga a los SI para sobrevivir o para desarrollar ventajas competitivas. Este indicador será medido a través de tres (3) variables.</p>
Factores Internos	
<p>Gerencia. Desarrollo de los procesos gerenciales dentro de la unidad desarrolladora de SI: desde el punto de vista de la conducción de los proyectos de desarrollo de SI con el uso de herramientas CASE y desde el punto de vista de la dirección de la unidad para el logro de sus objetivos. Indicadores:</p>	<p>Apoyo gerencial. Grado de apoyo que los gerentes de los proyectos de desarrollo de SI de la unidad desarrolladora de SI le brindan a los procesos de adquisición de las herramientas CASE. Este indicador será medido a través de tres (3) variables.</p> <p>Proceso de implementación de innovaciones tecnológicas. Procesos que la unidad desarrolladora de SI tiene estipulados para acometer la adquisición y uso de las herramientas CASE dentro de sus proyectos de desarrollo. Este indicador será medido a través de cinco (5) variables.</p> <p>Proceso de actualización de hardware y software. Procesos que la unidad desarrolladora de SI tiene estipulados para acometer la actualización de HW y SW cuando se decide adquirir y usar una herramienta CASE dentro de los proyectos de desarrollo de la unidad. Este indicador será medido a través de cuatro (4) variables.</p> <p>Plan de adiestramiento. Procesos que la unidad desarrolladora de SI tiene estipulados para acometer la actualización de sus analistas y desarrolladores de SI, al momento de adoptar una herramienta CASE. Este indicador será medido a través de cuatro (4) variables.</p> <p>Estructura organizacional. Tipo de estructura organizacional que caracteriza a la unidad desarrolladora de SI, así como al grado de flexibilidad que ésta tiene para cambiar y/o adaptarse por efecto de la adopción de las herramientas CASE. Este indicador será medido a través de tres (3) variables.</p> <p>Gerencia de proyectos. Procesos que la unidad desarrolladora de SI tiene estipulados para acometer la gerencia de los proyectos de desarrollo de SI, previendo la adopción de una herramienta CASE. Este indicador será medido a través de cuatro (4) variables.</p>
<p>Operación. Se refiere al desarrollo de los procesos operacionales dentro de la organización desarrolladora de SI para la conducción de los proyectos de desarrollo de SI aprovechando en su totalidad las potencialidades de las herramientas CASE y para asegurar el éxito de la adquisición y gerencia de las innovaciones tecnológicas. Indicadores:</p>	<p>Participación de los desarrolladores de SI en la toma de decisiones dentro de la organización desarrolladora de SI. Grado de participación que tienen los analistas o desarrolladores de SI en la selección de herramientas CASE y en la planificación de los procesos de implementación de las herramientas dentro de la unidad desarrolladora de SI (adopción). Este indicador será medido a través de tres (3) variables.</p> <p>Compatibilidad con la metodología de desarrollo. Grado de consistencia que se espera que tenga la herramienta CASE con respecto a los procesos establecidos y con respecto a las necesidades que la unidad desarrolladora quiere satisfacer con la adopción de la herramienta CASE. Este indicador será medido a través de cuatro (4) variables.</p> <p>Capacidades y habilidades de los analistas. Grado de experiencia que tienen los analistas o desarrolladores de SI pertenecientes a la unidad desarrolladora de SI y, a las capacidades de éstos para adquirir, a través del respectivo entrenamiento, las capacidades y habilidades para el uso de las herramientas CASE. Este indicador será medido a través de tres (3) variables.</p>

Tabla N° 12. Factores Organizacionales.
Fuente: [Mendoza, 1999]

Mendoza [Mendoza, 1999] indica, que las organizaciones que utilizan herramientas CASE para el desarrollo de SI exhiben unas condiciones que facilitan la adopción y uso de éstas; las cuales han sido el resultado de los cambios que ellas han tenido que asumir y realizar por efectos de la

incorporación de los SI y las Tecnologías de Información a su ambiente organizacional y, en particular, por la experiencia que las organizaciones han tenido con las herramientas CASE. En este sentido sobre la base de la propuesta de los indicadores y de la realización del estudio de campo, Mendoza presenta un resumen de los aspectos más importantes que caracterizan a las organizaciones venezolanas que han tenido éxito en adoptar y usar herramientas CASE, de manera general se tiene que, los SI desarrollados y las Tecnologías de Información adoptadas por las organizaciones han tenido un buen impacto en éstas, lo que ha generado una actitud favorable hacia las innovaciones tecnológicas; además, las organizaciones tienen una clara conciencia acerca de la necesidad de contar con SI de calidad para mejorar la productividad y soportar la toma de decisiones; por otro lado, la alta gerencia de las organizaciones y la gerencia de las unidades desarrolladoras de SI, apoyan todos los procesos relacionados con las innovaciones tecnológicas, en especial, los relacionados con la actualización de hardware y software y, los referidos a la implementación de innovaciones tecnológicas; de igual manera, las organizaciones están convencidas de la importancia de los SI y las TI para desarrollar ventajas competitivas; las unidades desarrolladoras de SI cuentan con una sólida gerencia de proyectos, donde la participación activa de los desarrolladores de SI en la toma de decisiones es un estándar y, los planes de adiestramiento soportan el desarrollo de las capacidades y las habilidades de los analistas; además, cuando las organizaciones deciden adoptar una innovación tecnológica y en especial, una herramienta CASE, toman en cuenta la compatibilidad que ésta tiene con la infraestructura tecnológica instalada y con los procesos de trabajo (métodos y/o metodologías) existentes para el logro de los objetivos organizacionales.

Después de analizar 9 investigaciones concernientes a la incorporación de herramientas CASE al proceso de desarrollo de SI dentro de una organización, a nivel internacional y nacional, es posible afirmar entre otras cosas, que **el proceso de selección, adquisición e implementación de una herramienta CASE en una organización no es fácil**; son muchos los factores que pueden afectar la adopción de éstas, siendo los más importantes, el punto de vista del desarrollador de SI, en primer lugar, y el punto de vista de los altos gerentes de las organizaciones que las usan.

Independientemente del enfoque y las características propias de cada investigación, **ninguna deja a un lado la influencia que tiene las características tecnológicas de la herramienta CASE y el entorno organizacional frente a la adquisición y/o adopción de las herramientas CASE para soportar el proceso de desarrollo de SI.**

Reforzando lo anterior, Hoffer et. al. [Hoffer et. al., 1996], plantea que las organizaciones usan herramientas CASE para *“mejorar la calidad de los sistemas desarrollados, incrementar la velocidad con la cual los sistemas son diseñados y desarrollados, facilitar y mejorar el proceso de prueba a través del uso de chequeos automatizados, mejorar la integración de actividades de desarrollo mediante metodologías comunes, mejorar la calidad y la completitud de la documentación, ayudar a estandarizar el proceso de desarrollo, mejorar la gerencia del proyecto, simplificar los programas de mantenimiento, promover la reusabilidad de los módulos y de la documentación y mejorar la portabilidad del software entre distintos ambientes”*.

Ahora bien, de acuerdo al planteamiento de Hoffer, y a las conclusiones obtenidas de los factores que afectan la adopción de herramientas CASE por parte de las organizaciones desarrolladoras de SI a nivel mundial, queda establecido que el uso de herramientas CASE es totalmente indispensable durante el desarrollo de un producto software, y esta importancia queda reforzada con la evolución que han sufrido las herramientas CASE luego se divulgarse una nueva forma de desarrollar software asistida por la transformación de los modelos, es así como a continuación se presenta todo lo relacionado con **MDA** o **Arquitectura Dirigida por Modelos**, y poder entender hacia a donde apuntan las herramientas CASE hoy en día.

2.2 MDA – ARQUITECTURA DIRIGIDA POR MODELOS.

La OMG en (OMG, 2003), MDA proporciona una solución para los cambios de negocio y de tecnología, permitiendo construir aplicaciones independientes de la plataforma e implementarlas en plataformas como CORBA, J2EE o Servicios Web. Según la OMG, MDA se trata de un framework de desarrollo de software que define una nueva forma de construir software en la que se usan modelos del sistema a distintos niveles de abstracción para guiar todo el proceso de desarrollo, desde el análisis y diseño hasta el mantenimiento del sistema y su integración con futuros sistemas.

Según (OMG, 2003), MDA pretende separar, por un lado la especificación de las operaciones y datos de un sistema, y por el otro, los detalles de la plataforma en la que el sistema será construido. Para ello MDA proporciona las bases para:

- Definir un sistema independiente de la plataforma sobre la que se construye.
- Definir plataformas sobre las que construir los sistemas.
- Elegir una plataforma particular para el sistema y,
- Transformar la especificación inicial del sistema a la plataforma elegida.

Los principales objetivos de MDA son mejorar la productividad, la portabilidad, la interoperabilidad y la reutilización de los sistemas.

A grandes rasgos, el proceso de desarrollo de software con MDA se puede dividir en tres fases:

- **Construcción de un Modelo Independiente de la Computación (MIC):** es un modelo que muestra el entorno en el cual operará el próximo sistema software a desarrollar. Puede mostrar diferentes aspectos del negocio donde funcionará la nueva aplicación.
- **Construcción de un Modelo Independiente de la Plataforma (MIP):** es un modelo de alto nivel del sistema independiente de cualquier tecnología o plataforma de software.
- **Transformación del modelo anterior a uno o varios Modelos Específicos de la Plataforma (MEP):** Es un modelo de más bajo nivel de abstracción que el MIP que describe el sistema de acuerdo con una tecnología de implementación determinada.
- **Generación de código a partir de cada MEP:** Debido a que cada MEP esta muy ligado a una tecnología concreta, la transformación de cada MEP a código puede automatizarse.

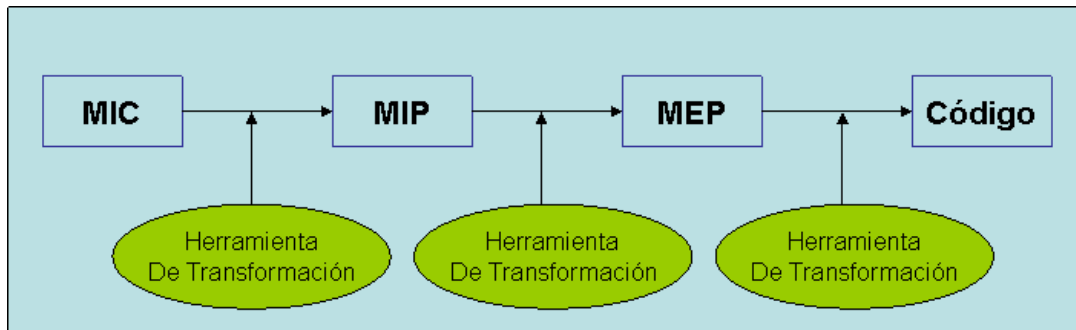


Figura N° 9. Pasos en el Desarrollo con MDA
Fuente: Elaboración Propia.

El paso de MIP a MEP y de MEP a código no se realiza a mano, sino que se usan herramientas de transformación, es decir herramientas CASE, para automatizar estas tareas.

Una vez definido el marco de trabajo de MDA se procede a conceptualizarla de acuerdo en lo expuesto por (OMG, 2003).

2.2.1 CONCEPTOS BÁSICOS DE MDA

Algunos de los conceptos más importantes que forman parte de la especificación MDA son según la OMG (OMG, 2003):

Modelo. Es una descripción o especificación mediante un lenguaje visual de un sistema.

Metamodelo. Es la descripción y especificación de los elementos y reglas que se utilizan para crear modelos semánticamente correctos para un dominio en particular. También puede definirse como el modelo de un lenguaje de modelado.

Dirigido por modelos (*Model Driven*). Se dice que es dirigido (o guiado) por modelos porque provee mecanismos que usan modelos para dirigir el curso del diseño, la construcción, la implementación, la operación, el mantenimiento y la modificación de una aplicación. Es decir, el proceso depende de los modelos.

Arquitectura. La arquitectura de un sistema es la especificación de las partes y conectores del sistema, así como las reglas de interacción entre éstas.

Vista. Es una representación del sistema desde la perspectiva de un punto de vista determinado.

Plataforma. Es un conjunto de subsistemas y tecnologías que proveen un conjunto de funcionalidades que cualquier aplicación soportada por la plataforma puede utilizar sin preocupación sobre los detalles de implementación de dicha plataforma.

Punto de vista. Un punto de vista en un sistema es una técnica de abstracción que utiliza un conjunto selecto de conceptos arquitecturales y reglas de estructuración, de manera que se enfoque la atención sólo en un problema particular del sistema. MDA especifica tres puntos de vista sobre

un sistema: el punto de vista independiente de la computación, el punto de vista independiente de la plataforma y el punto de vista específico de la plataforma.

- Punto de vista independiente de la computación. Este se enfoca en el ambiente del sistema y los requerimientos del mismo, es decir, la lógica del negocio. Los detalles de la estructura y el procesamiento del sistema están escondidos o no han sido determinados.
- Punto de vista independiente de la plataforma. Se enfoca en la operación del sistema mientras oculta los detalles específicos para cierta plataforma, es decir, muestra la parte de la implementación que es idéntica de una plataforma a otra.
- Punto de vista específico de una plataforma. Combina el punto de vista independiente de la plataforma con el detalle del uso de una plataforma específica.

Como puede verse MDA plantea una serie de enfoques con características muy particulares a través de los cuales se va construyendo una aplicación, a continuación se detallarán los modelos que conforman la arquitectura de MDA.

2.2.2 MODELOS EN MDA

Un Modelo es una descripción de todo o parte de un sistema escrito en un lenguaje bien definido. El hecho de que un modelo este escrito en un lenguaje bien definido tiene una gran importancia para MDA, ya que supone que el modelo tiene asociadas una sintaxis y una semántica bien definidas. Esto permite la interpretación automática por parte de transformadores o compiladores de modelos, fundamentales en MDA.

UML es un lenguaje de modelado bien definido que se ha adoptado como el principal lenguaje de modelado en MDA. Se dice el principal lenguaje de modelado porque MDA no está restringido a UML, sino que se puede usar cualquier lenguaje bien definido. No obstante, la realidad muestra que UML se ha convertido en el lenguaje de modelado oficial de MDA, así que de aquí en adelante se asumirá que todos los modelos se construyen usando este lenguaje.

De acuerdo con (García y Rodríguez, 2004), de los distintos tipos de modelos definidos en UML, los Diagramas de Clases que muestran la vista estática de un sistema, son los más importantes dentro de MDA, ya que el MIP y la mayoría de los MEP son modelos de este tipo. La figura N° 10 muestra un ejemplo de un Diagrama de Clases.

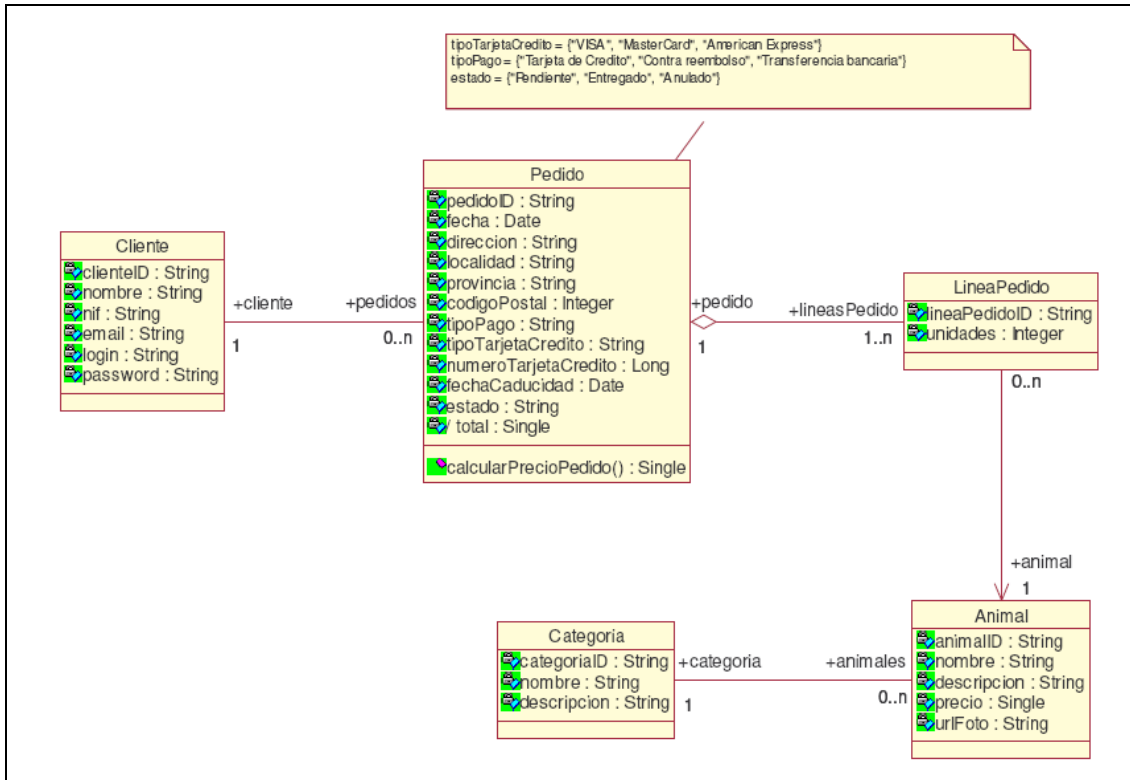


Figura N° 10. Diagrama de Clases en UML

Fuente: (García y Rodríguez, 2004)

2.2.2.1 MIC - MODELO INDEPENDIENTE DE LA COMPUTACIÓN

Es un modelo del sistema desde el punto de vista independiente de la computación. Un MIC no muestra detalles de la estructura del sistema y a veces es llamado modelo del dominio o modelo del negocio. El MIC forma parte importante en la construcción de puentes entre aquellos expertos en el dominio y sus requisitos, por un lado, y aquellos expertos en el diseño y construcción de artefactos para satisfacer estos requisitos por el otro.

2.2.2.2 MIP (MODELO INDEPENDIENTE DE LA PLATAFORMA)

Un Modelo Independiente de la Plataforma o (MIP) es un modelo del sistema de alto nivel que representa la estructura, funcionalidad y restricciones del sistema sin relacionarse con una plataforma determinada. Este modelo servirá de base para todo el proceso de desarrollo, y es el único que debe ser creado íntegramente por el desarrollador.

Al no incluir detalles específicos de una plataforma o tecnología determinada, este modelo es útil en dos aspectos:

- Es fácilmente comprensible por los usuarios del sistema, y por lo tanto, les resultará más sencillo validar la corrección del sistema.
- Facilita la creación de diferentes implementaciones del sistema en diferentes plataformas, dejando intacta su estructura y su funcionalidad básica

A continuación se muestra la figura N° 11, donde se presenta un MIP sencillo.

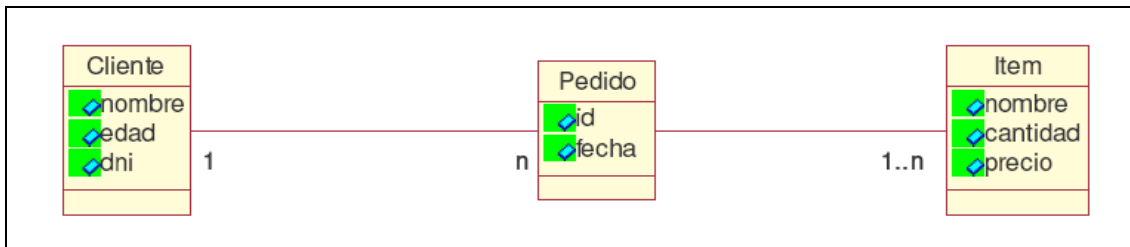


Figura N° 11. Ejemplo de MIP
Fuente: (García y Rodríguez, 2004)

2.2.2.3 MODELO ESPECÍFICO DE LA PLATAFORMA (MEP)

Un Modelo Específico de la Plataforma (MEP) es un modelo del sistema con detalles específicos de la plataforma en la que será implementado. Se genera a partir de un MIP, así que representa el mismo sistema pero a un nivel distinto de abstracción. Se puede decir que un MEP es un MIP al que se le añaden detalles específicos para ser implementado en una plataforma determinada.

Un MEP es un modelo del sistema de más bajo nivel, mucho más cercano a la vista de código que el MIP. Puede incluir más o menos detalles dependiendo de su propósito.

La figura N° 12, muestra un MEP a partir de un MIP mostrado en la figura N° 11, representado a través de un diagrama de clases de UML. Este sencillo ejemplo muestra un posible MEP para la plataforma EJB. Como se puede observar en el paso del MIP al MEP se han producido varias transformaciones:

- Se ha añadido el estereotipo (EJBEntity) a cada clase, para indicar que cada clase representa un EJB de tipo Entity.
- Se ha modificado la visibilidad de los atributos del MIP de público a privado.

- Se han generado métodos públicos de lectura y modificación (get y set) para cada atributo.

Para la construcción de MEP se usan los **Perfiles UML** (UML Profiles), que son extensiones de UML que permiten añadir información semántica a los modelos para expresar detalles específicos de la plataforma. Por ejemplo, el estereotipo EJBEntity de las clases de la figura N° 12, parte del Perfil UML para la plataforma EJB. Los Perfiles UML se detallarán más adelante en la sección N° 2.2.8.2.

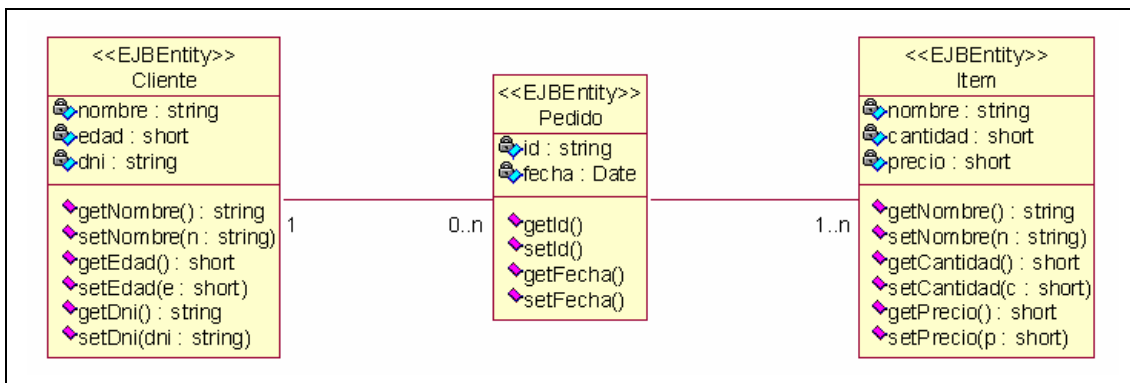


Figura N° 12. Ejemplo de MEP
Fuente: (García y Rodríguez, 2004)

Hay que destacar que a partir de un mismo MIP se pueden generar varios MEP, cada uno describiendo el sistema desde una perspectiva diferente.

La transformación de MIP a MEP se ejecutarse de varias formas:

- Construyendo manualmente el MEP a partir del MIP.
- De forma semiautomática, generando un MEP completo a partir del MIP.

Para las transformaciones automáticas se usan herramientas especializadas. Estas herramientas tienen implementados distintos algoritmos de transformación para pasar de un tipo de modelo a otro, y suponen unos de los pilares de MDA. Las herramientas de transformación de modelos se verán en detalle más adelante.

Un MEP también puede refinarse, transformándose sucesivamente en MEP de más bajo nivel, hasta llegar al punto en que pueda ser transformado a código de manera directa.

A partir de un MEP y mediante una herramienta de transformación, se obtiene gran parte del código que implementa el sistema para la plataforma elegida. El desarrollador tan solo tendrá que añadir aquella funcionalidad que no puede representarse mediante el MIP o el MEP.

2.2.3 GENERACIÓN DE PUENTES DE COMUNICACIÓN. EJEMPLO DE DESARROLLO CON VARIOS MEP.

En la figura N° 9, se mostró un esquema sencillo de las fases de desarrollo propuestas por MDA. Un esquema más general sería aquel en el que se tienen varios MEP derivados del mismo MIP, representando cada MEP una parte del sistema. En ese caso, además de generar los MEP a partir del MIP y el código a partir de cada MEP, también deberían generarse los **puentes de comunicación** entre las distintas partes, tanto a nivel de MEP como a nivel de código. El siguiente ejemplo sirve para aclarar esta idea.

Un esquema típico de desarrollo con MDA usando varios MEP es el que aparece en la figura N° 13. A partir del MIP se generan tres MEP:

- Un modelo relacional del sistema, que describe el esquema de base de datos del sistema a través de un Diagrama Entidad-Relación.
- Un modelo EJB, mostrando los aspectos relativos a la plataforma EJB.
- Un modelo Web para describir las interfaces Web del sistema.

Los puentes de comunicación entre los MEP y el código permitirían a la capa Web comunicarse con los componentes EJB, y a estos comunicarse con la base de datos del sistema.

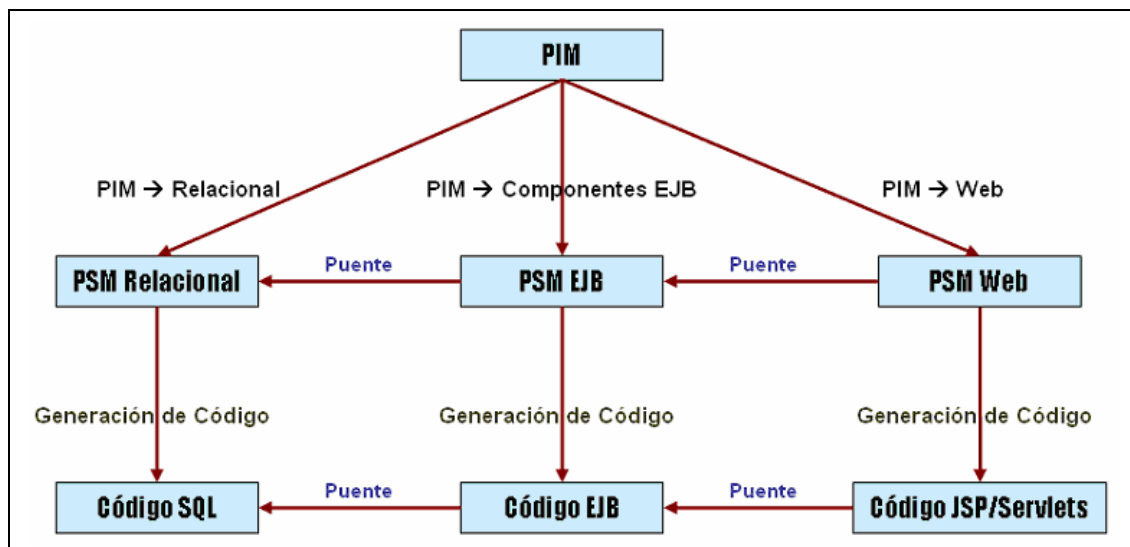


Figura N° 13. Ejemplo de Desarrollo en MDA con varios MEP
Fuente: (García y Rodríguez, 2004)

Ahora bien, resulta interesante saber como se pasa del proceso de desarrollo de software tradicional al proceso de desarrollo dirigido por MDA, a continuación se hace una revisión comparativa de ambos procesos.

2.2.4 DESARROLLO TRADICIONAL VS DESARROLLO CON MDA.

En el (Kleppe, et al, 2003) se realiza una interesante comparación entre el desarrollo tradicional y el desarrollo con MDA, ilustrando los beneficios de este nuevo proceso de desarrollo. Un proceso típico de desarrollo incluye las siguientes fases: Requisitos, análisis, diseño, implementación, pruebas y despliegue. La figura N° 14, muestra el desarrollo de software tradicional.

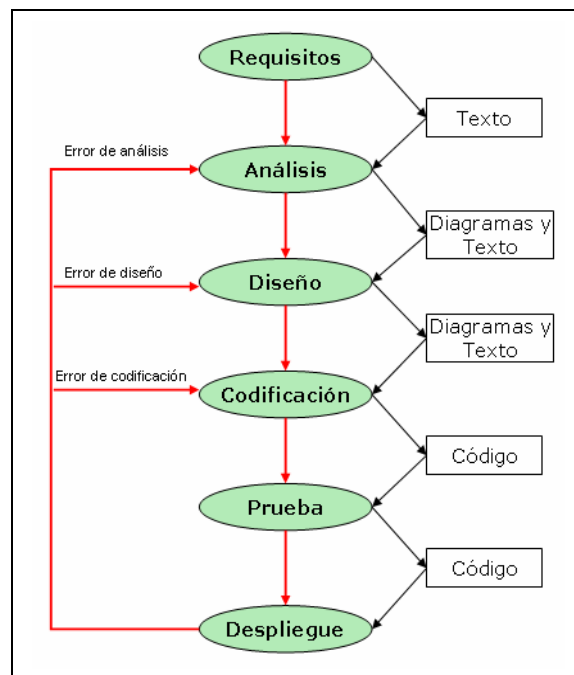


Figura N° 14. Proceso de Desarrollo de Software Tradicional
Fuente: (García y Rodríguez, 2004)

Durante los últimos años se han hecho muchos progresos en el desarrollo del software, que han permitido construir sistemas más grandes y complejos. Aun así la construcción de software de manera tradicional sigue teniendo ciertos problemas:

1. Productividad: El proceso tradicional produce una cantidad de documentos y diagramas para especificar requisitos, clases, interacciones, etc. La mayoría de este material pierde su valor en cuanto comienza la fase de codificación y gradualmente se va perdiendo la relación entre los diagramas. Y más aun cuando el sistema cambia a lo largo del tiempo: realizar los cambios en todas las fases (requisitos, análisis, diseño, etc.) se hace inmanejable, así que generalmente se realizan las modificaciones sólo en el código. Entonces, ¿para qué perder el tiempo en construir los diagramas y la documentación de alto nivel?. Lo cierto es que para sistemas complejos sigue siendo necesario. Lo que se necesita entonces, es un soporte para que un cambio en cualquiera de las fases se traslade fácilmente al resto.
2. Portabilidad: En la industria del software, cada año aparecen nuevas tecnologías y las empresas necesitan adaptarse a ellas, bien porque la demanda de esa tecnología es alta (es “lo que se lleva”), bien porque realmente resuelva problemas importantes. Como consecuencia el software existente debe adaptarse o migrar a la nueva tecnología. Esta migración no es ni mucho menos trivial, y obliga a las empresas a realizar un importante desembolso.
3. Interoperabilidad: La mayoría de sistemas necesitan comunicarse con otros, probablemente ya construidos. Incluso si los sistemas que van a interoperar se construyen desde cero, frecuentemente usan tecnologías diferentes. Por ejemplo, un sistema que use Enterprise Javabeans necesita también bases de datos relacionales como mecanismo de almacenamiento de datos. Se necesita que la interoperabilidad entre sistemas, nuevos o ya existentes, se consiga de manera sencilla y uniforme.
4. Mantenimiento y Documentación: Documentar un proyecto software es una tarea lenta que consume mucho tiempo, y que en realidad no interesa tanto a los que desarrollan el software, sino a aquellos que lo modifican o lo usarán más adelante. Esto hace que se ponga poco empeño en la documentación y que generalmente no tenga una buena calidad. La solución de este problema a nivel de código es que la documentación se genere directamente del código fuente, asegurándose que esté siempre actualizada. No obstante la documentación de alto nivel (diagramas y texto) todavía debe ser mantenida a mano.

Estos problemas se solucionan en MDA de la siguiente manera, según (Kleppe, et al, 2003):

2.2.5 BENEFICIOS DEL MDA

1. Productividad: En MDA el foco de desarrollo recae sobre el MIP. Los MEP se generan automáticamente (al menos en gran parte) a partir del MIP. Por supuesto, alguien tiene que definir las transformaciones exactas, lo cual es una tarea especializada y difícil. Pero una vez implementada la transformación, puede usarse en muchos desarrollos. Y lo mismo ocurre con la generación de código a partir de los MEP. Este enfoque centrado en el MIP aísla los problemas específicos de cada plataforma y encaja mucho mejor con las necesidades de los usuarios finales, puesto que se puede añadir funcionalidad con menos esfuerzo. El trabajo “sucio” recae sobre las herramientas de transformación, no sobre el desarrollador.
2. Portabilidad: En MDA la portabilidad se logra también enfocando el desarrollo sobre el MIP. Al ser un Modelo independiente de cualquier tecnología, todo lo definido en él es totalmente portable. Otra vez el peso recae sobre las herramientas de transformación, que realizarán automáticamente el paso del MIP al MEP de la plataforma deseada.
3. Interoperabilidad: Los MEP generados a partir de un mismo MIP normalmente tendrán relaciones, que es lo que en MDA se llaman puentes. Generalmente los distintos MEP no podrán comunicarse entre ellos directamente, ya que pueden pertenecer a distintas tecnologías. Este problema lo soluciona MDA generando no solo los MEP, sino también los puentes entre ellos. Como es lógico estos puentes serán construidos por las herramientas de transformación, que como se puede observar es uno de los pilares fundamentales de MDA.
4. Mantenimiento y Documentación: Como ya se ha dicho a partir de un MIP se generan los MEP, y a partir de los MEP se genera el código. Básicamente, el MIP desempeña el papel de la documentación de alto nivel que se necesita para cualquier sistema software. Pero la gran diferencia es que el MIP no se abandona luego de la codificación. Los cambios realizados en el sistema se reflejarán en todos los niveles, mediante la regeneración de los MEP y del código. Aun así se sigue necesitando documentación adicional que no puede expresarse en el MIP, por ejemplo para justificar las elecciones hechas para construir el MIP.

De acuerdo con (Anacleto, 2006), la ventaja principal de MDA radica en una clara y estricta **separación de responsabilidades**. Por un lado, modelaremos los PIMs, que representan los modelos de nuestro negocio, y por el otro, los PSMs con las preocupaciones tecnológicas. Esto

permitirá que ambos modelos evolucionen por separado. De esta manera, si quisiéramos, por ejemplo, modificar un aspecto técnico, bastaría con modificar el PSM sin que éstos tuvieran impacto en la lógica de negocios. Esta idea parte de un concepto que, en ingeniería de software, se llama **Guías de Diseño**. Particularmente, una de esas guías dice que el modelado de la solución debe ser guiado por el negocio. Esta guía se basa en la afirmación de que un cambio en el negocio seguramente producirá uno en el código, pero que los cambios en el código no deberían impactar en el negocio. MDA también permite lidiar con la complejidad del negocio, modelando a éste por separado, y permitiendo su análisis y mejora; **disminuir costos**, si se cuenta con una herramienta MDA adecuada a nuestras necesidades; y mejorar la **calidad** de nuestros modelos y procesos, mediante su análisis y la separación de responsabilidades.

2.2.6 EL NUEVO PROCESO DE DESARROLLO.

De acuerdo con (Kleppe, et al, 2003), si se compara el proceso MDA con el proceso de desarrollo tradicional se observa que las fases de requisitos, pruebas y despliegue seguirán igual. Lo que cambia son las fases de análisis, diseño e implementación, de la siguiente manera:

- **Análisis:** un grupo especial de personas desarrollaran el MIP, guiados por las necesidades del negocio y la funcionalidad que debe incorporar el sistema.
- **Diseño:** Un grupo diferente de personas se encargarán de la transformación del MIP a uno o más MEP. Estas personas tendrán conocimientos sobre distintas plataformas y arquitecturas, y conocerán también las transformaciones disponibles en las herramientas que usan. De este modo podrán elegir la plataforma o arquitectura que mejor se adapte a los requisitos del sistema y establecer los parámetros de las distintas transformaciones. Los creadores de los MEP tendrán comunicación constante con los diseñadores del MIP para tener más información sobre el sistema (por ejemplo, conocer los requisitos no funcionales).
- **Implementación:** Esta fase se reduce a generar el código del sistema mediante herramienta especializadas. Los programadores únicamente tendrán que añadir la funcionalidad que no puede reflejarse en los modelos y, si es necesario, retocar el código generado. Según los expertos, en un futura desaparecerá esta fase de codificación dentro de MDA, y se pasará directamente del MEP a las pruebas.

Pero en este nuevo proceso de desarrollo aun falta un tercer grupo de personas, aquellos que escriben definiciones de transformaciones para empresas desarrolladoras de software o

principalmente para los vendedores de herramientas MDA, pues permiten a las herramientas pasar de MIP a MEP, y de MEP a código.

La figura N° 15 y 16, muestran un esquema sencillo que ilustra este nuevo proceso de desarrollo con MDA. Se ve como los cambios se centran en el MIP, base de todo el desarrollo, y gracias a las herramientas de transformación estos cambios se trasladan rápidamente al resto de las fases.

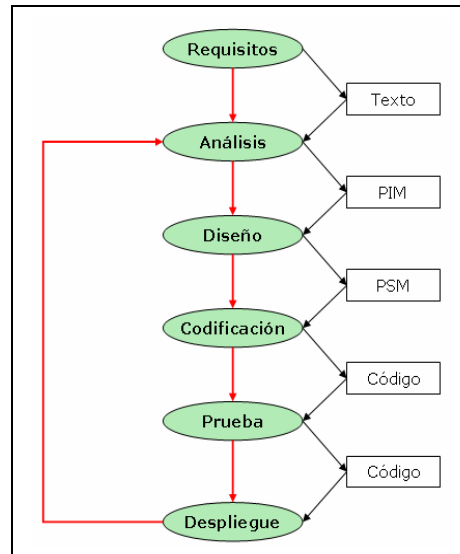


Figura N° 15. Proceso de Desarrollo con MDA
Fuente: (García y Rodríguez, 2004)

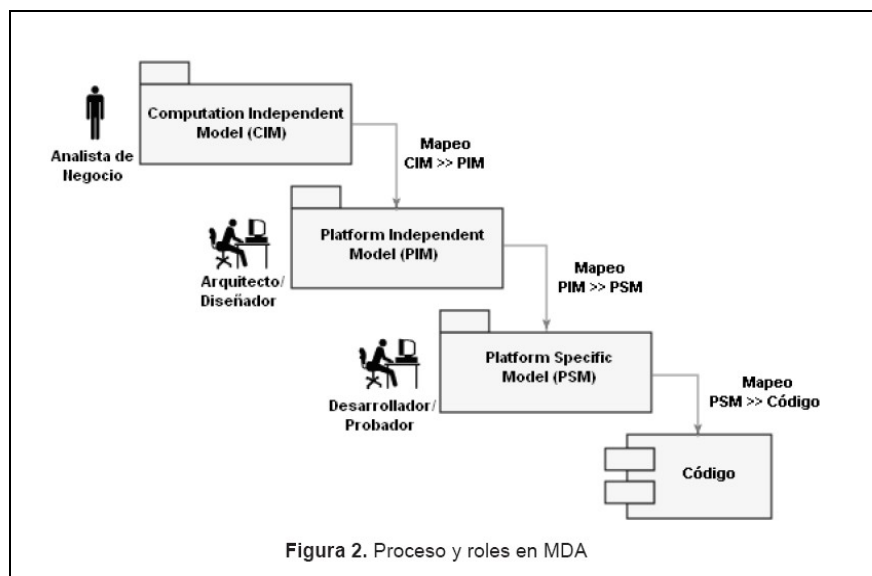


Figura N° 16. MDA y el papel de los modelos en el proceso de desarrollo de software.

Fuente: Juan Bernardo Quintero y Raquel Anaya

2.2.7 VISIÓN ALTERNATIVA DE MDA.

Una visión alternativa y práctica de MDA se encuentra en el manifiesto (2) elaborado por los principales investigadores de IBM trabajando en MDA, en donde se resume MDA como la combinación de tres ideas complementarias. La figura N° 17, muestra estas tres ideas que serán descritas a continuación.

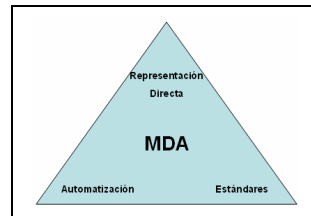


Figura N° 17. Visión Alternativa de MDA

Fuente: (García y Rodríguez, 2004)

- **Representación Directa:** Se desplaza el foco del desarrollo de software del dominio de la tecnología hacia las ideas y conceptos del dominio del problema. Reduciendo la distancia semántica entre el dominio del problema y su representación se permite un mayor acoplamiento de las soluciones a los problemas, logrando diseños más acertados e incrementando la productividad.
- **Automatización:** Usar herramientas para mecanizar aquellas facetas del desarrollo del software que no dependen del ingenio humano. MDA incrementa la velocidad de desarrollo y reduce errores usando herramientas automatizadas para transformar modelos específicos de la plataforma en código de implementación. Es lo mismo que hacen los compiladores para los lenguajes de programación tradicionales.
- **Estándares abiertos:** Los estándares son uno de los mejores impulsores del progreso en la historia de la tecnología. Los estándares de la industria no solo ayudan a eliminar la diversidad gratuita, sino que también alentan a los vendedores a producir herramientas tanto para propósitos generales como para propósitos especializados, incrementando así las posibilidades del usuario. El desarrollo de código abierto asegura que los estándares se implementan consistentemente y anima la adopción de estándares por parte de los vendedores.

En la siguiente sección se esbozan los fundamentos teóricos sobre los cuales se apoya MDA.

2.2.8 FUNDAMENTOS DE MDA.

Dentro de las bases fundamentales en las cuales se apoya MDA están el Metamodelado y el MOF, los Perfiles UML y el OCL. A continuación se irán detallando cada uno respectivamente.

2.2.8.1 METAMODELADO Y MOF

El **Metamodelado** es un mecanismo que permite definir formalmente lenguajes de modelado. Así, un metamodelo de un lenguaje es una definición precisa de sus elementos mediante conceptos y reglas de cierto metalenguaje, necesaria para crear modelos en ese lenguaje. Básicamente se trata de usar modelos para describir otros modelos. Por ejemplo, el metamodelo de UML define los conceptos y reglas que se necesitan para crear modelos UML.

Este concepto se entenderá mejor cuando se analice a continuación la arquitectura de cuatro capas de modelado que define la OMG.

Las Cuatro Capas de Modelado de la OMG.

La OMG usa una arquitectura de cuatro niveles o capas de modelado para sus estándares. En la terminología de la OMG estas capas se llaman M0, M1, M2 y M3, y se explicarán a continuación:

Capa M0. Instancias: En el nivel M0 están todas las instancias reales del sistema, es decir, los objetos de la aplicación. Aquí no se habla de clases y de atributos, sino de entidades físicas que existen en el sistema. El siguiente ejemplo, puede ayudar a entender este concepto.

Supongamos un sistema que debe manejar clientes de un videoclub. En la figura N° 18, se pueden observar las entidades que almacenan los datos (nombre, dirección y teléfono) de los clientes Frank Mendoza y Luis Linarez. Estas entidades son instancias pertenecientes a la capa M0.

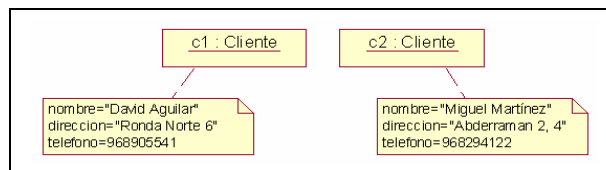


Figura N° 18. Instancias del Sistema (Capa M0)
Fuente: (García y Rodríguez, 2004)

Capa M1. Modelo del Sistema: Por encima de la capa M0 se sitúa la capa M1, que represente el modelo de un sistema software. Los conceptos del nivel M1 representan categorías de las instancias de M0. Es decir, cada elemento de M0 es una instancia de un elemento de M1.

Como se puede observar en la figura N° 19, en el nivel M1 aparecería la entidad Cliente con los atributos: nombre, dirección y teléfono, que representa a todos los clientes de la aplicación. El cliente de nombre Luis Linarez, es una instancia de esa entidad. La entidad Artículo, que representa a todos los artículos que pueden alquilarse en el videoclub, también pertenece al nivel M1.

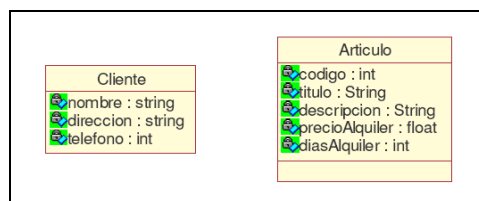


Figura N° 19. Instancias del Modelo del Sistema (Capa M1)
Fuente: (García y Rodríguez, 2004)

Los modelos usando UML, pertenecen a este nivel M1.

Capa M2. Metamodelo: Análogamente como ocurriría con M0 y M1, los elementos del nivel M1 son a su vez instancias del nivel M2. Esta capa recibe el nombre de capa de metamodelo. En este nivel aparecerán conceptos como Clase, Atributo o Relación.

La entidad Cliente mostrada en la figura anterior sería una instancia de la metaclase UML Class del metamodelo de UML que aparece en la figura N° 20, junto a la metaclase UML Atributo.

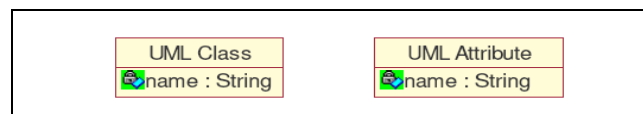


Figura N° 20. Entidades del Metamodelo de UML (Capa M2)
Fuente: (García y Rodríguez, 2004)

El metamodelo de UML estaría situado en este nivel de M2 y gracias a él se pueden construir modelos de nivel M1 en UML.

Capa M3. Meta-metamodelo: En la misma línea, se pueden ver los elementos de M2 como instancias de otra capa superior M3, la capa de meta-metamodelo.

Dentro de la OMG, MOF es un lenguaje estándar de la capa M3. Esto supone que todos los meta-metamodelos de la capa M2, por ejemplo, el metamodelo de UML, son instancias de MOF, o lo que es lo mismo, UML se define usando MOF. La figura N° 21, muestra un ejemplo de entidades de la capa M3. La clase de M2 UML Class es una instancia de la clase de M3 MOF Class.

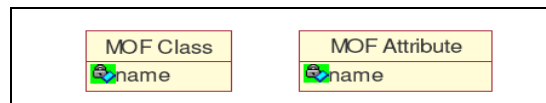


Figura N° 21. Entidades de MOF (Capa M3)
Fuente: (García y Rodríguez, 2004)

Una Visión Completa de las Capas.

Se podrían añadir más niveles a los ya descritos, pero la realidad es que no sería muy útil. En lugar de definir una capa M4, la OMG estableció que todos los elementos de M3, se pueden definir con instancias de conceptos de M3, lo que significa que MOF se define a si mismo.

La Tabla N° 13, muestra un ejemplo completo donde se aprecia la relación existente entre las capas de modelado definidas por la OMG.

Capa	Contenido	Ejemplo
M0 – Instancias	Instancias reales del sistema	Cliente con nombre="David Aguilar" y DNI=48491673
M1 – Modelo	Entidades del modelo del sistema	Clase "Cliente" con atributos "nombre" y "DNI"
M2 – Metamodelo	Entidades de un lenguaje de modelado	Entidad "UML Class" del metamodelo de UML
M3 – Meta-metamodelo	Entidades para definir lenguajes de modelado	Entidad "MOF Class" de MOF

Tabla N° 13. Relación entre las Capas del Modelado definidas por la OMG
Fuente: (García y Rodríguez, 2004)

Importancia del Metamodelado en MDA.

En primer lugar, el metamodelado es importante en MDA porque actúa como mecanismo para definir lenguajes de modelado, de forma que su definición no sea ambigua. Esta no ambigüedad permite que una herramienta pueda leer, escribir y entender modelos como los construidos con UML.

Por otro lado, las Reglas de Transformación, usadas para transformar un modelo en lenguaje A en otro modelo en lenguaje B, usan los metamodelos de los lenguajes A y B para definir las transformaciones.

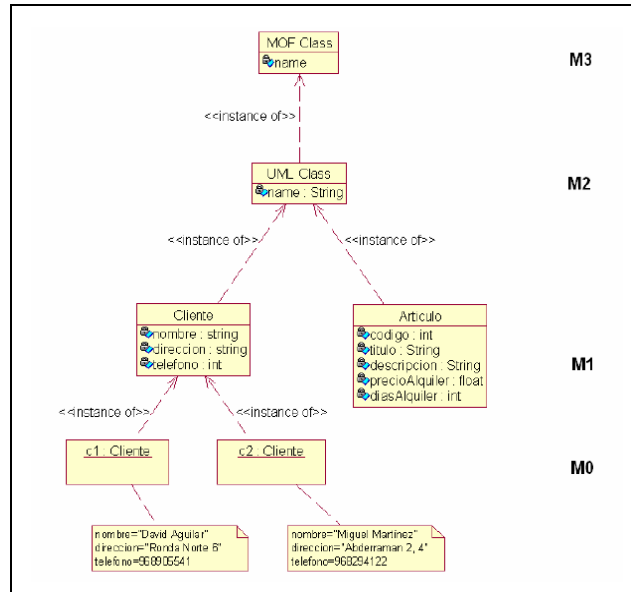


Figura N° 22. Relación entre las Distintas Capas de Modelado

Fuente: ()

MOF.

Uno de los estándares más importantes usados en MDA es Meta Object Facility o MOF: Se trata de un estándar de la OMG que define un lenguaje común y abstracto para definir lenguajes de modelado, y como acceder e intercambiar modelos expresados en dichos lenguajes. MOF usa cinco construcciones básicas para definir un lenguaje de modelado:

- Clases: usadas para definir tipos de elementos en un lenguaje de modelado. Por ejemplo, la relación de dependencia de UML, se llamará (UML::Dependencia) es una clase definida en MOF. Esta clase representa el tipo de todas las dependencias que pueden crearse en un modelo UML.
- Generalización: define herencia entre clases. Por ejemplo, UML::Clasificador es una generalización de UML::Clase. La subclase hereda todas las características de la clase padre.
- Atributos: usados para definir propiedades de los elementos del modelo. Los atributos tienen un tipo y una multiplicidad. Por ejemplo, UML::Elemento::nombre es un atributo de tipo string que define que cada elemento de un modelo UML tiene un nombre.

- Asociaciones: definen relaciones entre clases. Una asociación tiene dos extremos, cada uno de los cuales puede tener definido un nombre de rol, navegabilidad y multiplicidad. Por ejemplo, UML::Atributo tiene una asociación con UML::Clasificador con nombre tipo. Esta asociación se usa para definir el tipo de un atributo definido dentro de una clase UML.
- Operaciones: definen operaciones dentro del ámbito de una clase, junto con una lista de parámetros.

Atendiendo a las cuatro capas de modelado de la OMG, MOF estaría situado en el nivel M3 (nivel de meta-metamodelado). A través de MOF puede definirse cualquier lenguaje de modelado. En la figura N° 23, puede verse una versión simplificada del metamodelo de MOF.

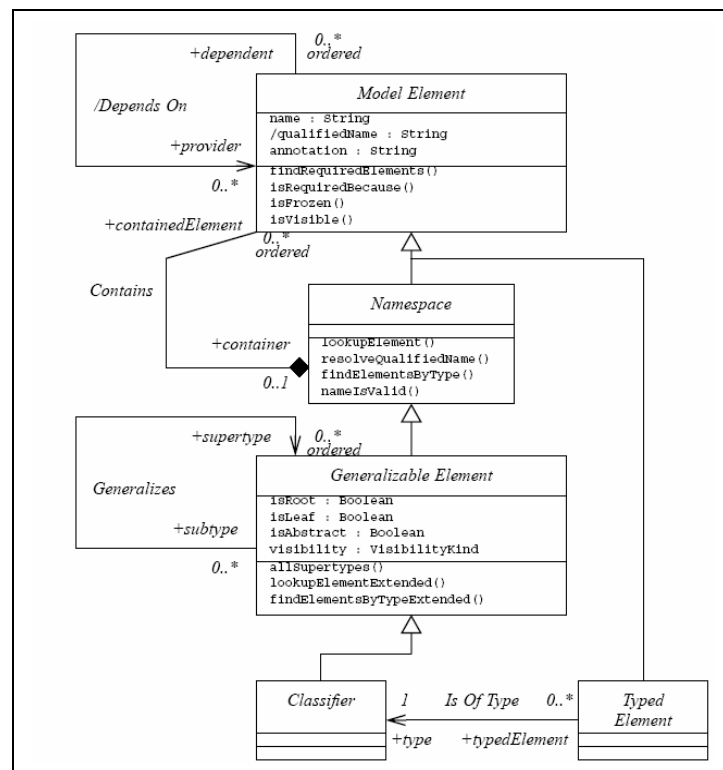


Figura N° 23. Abstracciones Principales de MOF.
Fuente: (OMG, MOF, 2002)

La definición de MOF también permite la construcción de herramientas para definir lenguajes de modelado, además de diversa funcionalidad adicional:

- MOF Repository Interface: Se trata de una interface que permite obtener información sobre modelos de nivel M1 de un repositorio basado en MOF.

- Intercambio de Modelos: MOF también define un formato de intercambio basado en XML para modelos M1, llamado XMI (XML Metadata Interchange). Ya que MOF se define usándose a si mismo, puede usarse también XMI para generar formatos de intercambio de metamodelos de una manera estándar.

2.2.8.2 PERFILES UML

Un **Perfil UML** se define como un conjunto de estereotipos, restricciones y valores etiquetados. Estos conceptos forman parte del mecanismo de extensión de UML. A continuación se explican cada uno de ellos.

- Mediante **estereotipos** se pueden crear nuevos tipos de elementos de modelado basados en elementos ya existentes en el metamodelo de UML. Por lo tanto un estereotipo será un nuevo tipo de elemento de modelado que extiende la semántica del metamodelo. Los estereotipos están definidos por un nombre y por una serie de elementos del metamodelo sobre los que puede asociarse. Gráficamente un estereotipo se representa con su nombre entre comillas “nombre-estereotipo”. Opcionalmente el elemento estereotipado se puede dibujar con un nuevo icono asociado al estereotipo.
- Las **Restricciones** imponen condiciones que deben cumplir determinados elementos del modelo para que este esté bien formado, según un dominio de aplicación específico. Gráficamente, una restricción se representa como una cadena de caracteres entre llaves colocada junto al elemento al que esta asociada o conectada a él por una relación de dependencia. Una restricción generalmente se define mediante una expresión en OCL.
- Un **Valor Etiquetado** es una extensión de las propiedades de un elemento de UML, permitiendo añadir nueva información en la especificación del elemento. Gráficamente un valor etiquetado se representa como una cadena de caracteres entre llaves asociada al nombre del elemento. La cadena incluye un nombre (etiqueta), un separador (=), y un valor (el de la etiqueta).

En la siguiente figura N° 24, se puede ver un ejemplo sencillo de la representación gráfica de estos tres elementos en UML.

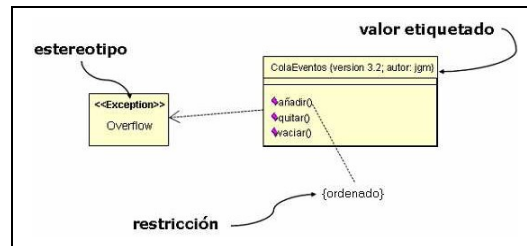


Figura N° 24. Ejemplo de Estereotipo, Restricción y Valor Etiquetado de UML
Fuente: (García y Rodríguez, 2004)

Para obtener un perfil se tendrá entonces que especializar un subconjunto de UML a través de estereotipos, restricciones y valores etiquetados. Como resultado de crear un perfil UML se obtiene una variante del UML para un propósito específico, que se puede usar para completar un modelo con detalles específicos de una tecnología o plataforma determinada, o lo que es lo mismo, se puede usar como lenguaje para definir MEP.

Los perfiles UML permiten:

- Disponer de una terminología y vocabulario propio de un dominio de aplicación o de una plataforma de implementación concreta.
- Definir una nueva notación para símbolos ya existentes, más acorde con el dominio de la aplicación final.
- Añadir cierta semántica que no existe o no aparece determinada de forma precisa en el metamodelo.
- Añadir restricciones a las existentes en el metamodelo, restringiendo su forma de utilización.
- Añadir información que pueda ser útil a la hora de transformar el modelo a otros modelos o a código.

Actualmente existen perfiles para CORBA, Java, EJB o C++, lo que permitirá construir MEP específicos para estas tecnologías.

2.2.8.3 OCL

Object Constraint Language (OCL) es un lenguaje de especificación con el que se pueden escribir expresiones sobre modelos, por ejemplo el cuerpo de una operación de consulta, invariantes y pre y

post condiciones. De esta manera se puede definir modelos más precisos y completos. Algunos de los usos de OCL son:

- Especificar valores iniciales de atributos.
- Definir el cuerpo de operaciones de consulta
- Establecer condiciones de guardia en Diagramas de Estado.
- Especificar las reglas de derivación para atributos o asociaciones.
- Expresar restricciones sobre clases o atributos.

A continuación se muestra algunos ejemplos de expresiones en OCL extraídos de (25)

La primera expresión define una invariante para la clase Compañía, estableciendo que el número de empleados debe ser mayor de 50.

```
context Compañia inv:
self.numeroDeEmpleados > 50
```

El siguiente ejemplo muestra la postcondición de una operación que establece las ganancias de una persona en una fecha determinada a 5000. La palabra reservada **result** representa el resultado de la operación.

```
context Persona::ganancias(d : Date) : Integer
post: result = 5000
```

El último ejemplo muestra el cuerpo (body) de una operación de consulta que recupera el cónyuge actual de una persona. La operación establece como precondition (pre) que la persona este casada.

```
context Persona::getConyugeActual() : Persona
pre: self.estaCasado = true
body: self.matrimonios->select( m | m.finalizado = false ).conyuje
```

Además de dar más precisión a los modelos; OCL puede usarse de manera muy efectiva en la definición de transformaciones, una primera expresión en OCL especifica los elementos del modelo origen, y una segunda expresión describe los elementos en el modelo destino de la transformación. Las condiciones necesarias para aplicar una transformación también pueden expresarse usando

OCL. No obstante, el lenguaje QVT, que será explicado más adelante, es el lenguaje estándar propuesto por la OMG para definir transformaciones.

2.2.9 TRANSFORMACIONES DE MODELOS.

DEFINICIONES DE TRANSFORMACIONES.

Una **Definición de Transformación** o **Mapping MDA** proporciona la especificación de la transformación de un MIP a un MEP para una plataforma determinada. Según (OMG, 2003), se pueden distinguir dos tipos de definiciones de transformaciones.

- **Transformaciones de Tipos, (Model Type Mapping).** Según la especificación “un mapping de tipos especifica un mapping para transformar cualquier modelo construido con tipos del MIP a otro modelo expresado con tipos del MEP”. Es decir, a cada tipo de elemento del MIP, se le aplica una regla determinada para transformarlo en uno o varios elementos del MEP. En el caso de UML, estas reglas pueden estar asociados a tipos del metamodelo (clase, atributo, relación, etc.) o a nuevos tipos definidos mediante estereotipos, como puede verse en el primer ejemplo. También pueden definirse reglas en función de valores de instancias en el MIP, como se ilustra en el segundo ejemplo.

Ejemplos:

1. Para cada instancia de clase con el estereotipo “ComponentSegment” en el MIP, transformarlas a un EJB Entity en el MEP para la tecnología EJB.
2. Para cada atributo del MIP con valor “public” en la visibilidad, transformarlo en el MEP a un atributo con visibilidad “private” y añadir dos métodos de consulta y actualización (get y set) a la clase a la que pertenezca.

- **Transformaciones de Instancias (Model Instance Mapping):** Identifica elementos específicos del MIP que deben ser transformados de una manera particular, dada una plataforma determinada. Esto se puede conseguir mediante marcas.

Una marca representa un concepto del MEP, y se aplica a un elemento del MIP para indicar como debe ser transformado. Las marcas, al ser específicas de la plataforma, no son parte del MIP. El desarrollador marca el MIP para dirigir o controlar la transformación a una plataforma determinada.

Ejemplo: Para la plataforma EJB, se podría tener la marca Entity aplicable a clases de un MIP; si está activa, esta marca indicará que la clase será transformada a un EJB Entity en el MEP.

La mayoría de las definiciones de transformación consistirán en alguna combinación de los dos enfoques. Una transformación de tipos solo es capaz de expresar transformaciones en términos de reglas sobre elementos de un tipo en el MIP que se transforman en elementos de uno o más tipos en el MEP. Sin embargo, si no se puede marcar el modelo con información adicional para dirigir la transformación, la transformación siempre será determinística, y se basará íntegramente en información independiente de la plataforma para generar el MEP.

Asimismo, toda transformación de instancias del modelo tiene restricciones implícitas de tipo que deben cumplirse al marcar el modelo para que la transformación tenga sentido. Implícitamente a cada tipo de elemento del MIP solo pueden aplicarse determinadas marcas, que indican que tipo de elemento se generará en el MEP. Las transformaciones basadas en marcas deben establecer qué marcas son aplicables a qué tipos del MIP.

CARACTERÍSTICAS DESEABLES DE LAS TRANSFORMACIONES.

Para poder llevar a cabo el enfoque propuesto por la MDA, conviene que el proceso de transformación disponga de una serie de propiedades o características. En el (Kleppe, et al, 2003) se exponen las características deseables que deben tener las transformaciones en MDA. Estas propiedades son, por orden de importancia, la posibilidad de ajustar las transformaciones, la trazabilidad, la consistencia incremental y al bidireccionalidad. A continuación se explica cada una de ellas.

1. Ajustar las Transformaciones.

El requisito principal que debería cumplir toda herramienta de transformación es permitir que el desarrollador tenga cierto control sobre el proceso de transformación, esto puede lograrse de varias maneras:

- **Control Manual:** implica un control directo del usuario sobre la transformación. El usuario puede definir manualmente qué elementos del modelo van a ser transformados por qué reglas de transformación. Esta es la solución más flexible, pero es propensa a errores y complica la tarea del desarrollador, por lo que es el método menos utilizado.

- **Condiciones en las Transformaciones:** el desarrollador asigna una condición a cada regla de transformación, la cual describe cuando debe aplicarse la regla. Este enfoque puede combinarse con el control manual. Un ejemplo podría ser: (todas las clases con el estereotipo “persistent” se transforman en...).
- **Parámetros de Transformación:** las definiciones de transformaciones pueden parametrizarse para permitir al desarrollador cambiar el “estilo” de las transformaciones. Por ejemplo, cuando se transforma un atributo publico en uno privado con métodos get y set, los prefijos exactos (normalmente get y set) podrían definirse como parámetros de la transformación. Otro parámetro típico de la transformación podría ser la longitud de tipos de datos de tamaño indefinido. Por ejemplo, en lugar de transformar un string en el MIP en un VARCHAR(40) en un MEP relacional, podríamos definir la longitud del dato como un parámetro de transformación. Los parámetros de transformación suponen el principal método de control sobre el proceso de transformación. Las marcas vistas en el apartado anterior pueden verse como parámetros de transformación para instancias del modelo.

2. Trazabilidad.

La trazabilidad implica que pueda conocerse el elemento origen a partir del cual se ha generado cualquier elemento del modelo destino.

Dentro del marco de MDA, la trazabilidad es una característica muy útil en muchas situaciones. Supóngase que se cambia el nombre de la una operación en el MEP que ha sido generada directamente a partir del MIP. Sería deseable que ese cambio se reflejase también el MIP. Esto no es posible si la herramienta no dispone de un mecanismo para conocer el origen en el MIP de la operación modificada.

La Trazabilidad también es útil en la búsqueda y corrección de errores. Las partes de código erróneas pueden encontrarse buscando los elementos del MIP, que presentan la funcionalidad defectuosa y siguiendo su traza hasta el código.

3. Consistencia Incremental.

Normalmente cuando se genera un modelo destino, éste necesita algún trabajo extra, como rellenar el código de una operación o refinar la interfaz de usuario. Si se regenera de nuevo el modelo destino, debido a cambios en el modelo origen, se quiere que ese trabajo extra se conserve. Esto es lo que se llama consistencia incremental.

Cuando se produce un cambio en el modelo origen, el proceso de transformación sabe qué elementos en el modelo destino necesitan cambiarse también. Un proceso de transformación incremental puede reemplazar los elementos viejos con los nuevos, mientras mantiene la información extra del modelo destino en su sitio. Esto significa que cambios en el modelo origen tienen el mínimo impacto en el modelo destino.

4. Bidireccionalidad.

La Bidireccionalidad implica que las transformaciones pueden operar en ambas direcciones. Esta propiedad, aunque interesante, tiene menor prioridad que las anteriores. Las transformaciones bidireccionales pueden lograrse de dos formas:

- Ambas transformaciones se ejecutan de acuerdo con una única definición de transformación.
- Una transformación y su inversa se especifican mediante dos definiciones de transformación diferentes.

Es muy complicado definir transformaciones bidireccionales, principalmente porque es muy difícil asegurar que una definición de transformación es la inversa de otra. DE hecho, en muchas ocasiones la bidireccionalidad es imposible de conseguir. Por ejemplo, cuando se transforma un modelo de negocio en un modelo relacional, solo se transforma la información estructural del modelo origen, ignorando toda la información dinámica. Esto hace imposible regenerar el modelo de negocio completo a partir del modelo relacional.

HERRAMIENTAS DE TRANSFORMACIÓN O HERRAMIENTAS CASE QUE SOPORTAN MDA.

Ninguna de las promesas de MDA sería posible sin herramientas que den soporte a este nuevo framework: transformación de modelos, verificación de modelos, generación de código, etc. La mayor parte del peso recae sobre estas nuevas herramientas de construcción de software.

El soporte para MDA puede darse en diferentes variantes. La generación de código a partir de modelos ya se lleva haciendo desde hace más de una década, y encaja bien dentro del marco de MDA. Además de la generación de código, se necesitan otros tipos de transformaciones, fundamentalmente transformaciones de modelos, para dar soporte completo a MDA. Estas

transformaciones pueden ser implementadas por distintas herramientas, que se resumirán a continuación.

- **Herramientas de Transformación de MIP a MEP:** Permiten transformar un MIP de alto nivel a uno o varios MEP.
- **Herramientas de Transformación de MEP a Código:** El soporte más popular para MDA se encuentran en las herramientas que transforman MEP a código. Poseen una definición de transformación integrada que toman un tipo predefinido de modelo de entrada (MEP) y producen otro tipo predefinido como salida (modelo de código). Las herramientas CASE tradicionales siguen este patrón para la generación de código. Muchas de estas herramientas mantienen la relación entre el MEP y el código, permitiendo que los cambios realizados en cualquiera de los modelos (MEP o código) se refleje inmediatamente en el otro. Esto es posible porque el MEP y el código están muy cerca, el uno del otro, y poseen casi el mismo nivel de abstracción.
- **Herramientas de Transformación de MIP a Código:** Otro tipo de herramienta es aquella que soporta las dos transformaciones anteriores de MIP a MEP y de MEP a Código. En ocasiones el usuario simplemente verá una transformación directa de MIP a Código, quedando implícito el MEP. La herramienta ArcStyler podría enmarcarse en este grupo, ya que no existe MEP explícito, sino que se usa un MIP con marcas para generar directamente el código.
- **Herramientas de Transformación Ajustables:** Las herramientas de MDA deberían permitir refinar o ajustar las transformaciones. Normalmente no se permite acceder a las definiciones de las transformaciones para ajustarlas a tus propios requisitos. En el mejor de los casos, las herramientas proporcionan sus definiciones de transformaciones en lenguajes de script específicos de la herramienta, y resulta muy engorroso hacer cambios en dichos lenguajes. Mientras no se estandarice un lenguaje de definición de transformaciones, estas seguirán siendo específicas de cada herramienta. Incluso la mayoría de las herramientas trabajan con un lenguaje para MIP predefinidos, que suele ser una variante restrictiva de UML, e internamente usan también su propia definición de UML. Esto obliga a los usuarios de la herramienta a aprender la definición específica de UML de la herramienta para poder escribir definiciones de transformaciones.
- **Herramientas de Definición de Transformaciones:** El último tipo de herramientas lo constituirían aquellas herramientas que nos permiten crear y modificar transformaciones sobre modelos, imprescindible si se necesita usar transformaciones distintas a las disponibles en las herramientas. Las únicas herramientas disponibles de este tipo usan

lenguajes de script específicos. La complejidad de la definición de transformaciones lleva de nuevo a la necesidad de un lenguaje estándar de definición de transformaciones (QVT) y a disponer de herramientas mejor preparadas para esta tarea. Por desgracia, todavía queda algo de tiempo para que esto se convierta en realidad.

- **Otras Herramientas:** Las herramientas de transformación son el verdadero corazón de un desarrollo con MDA, pero no son las únicas que se necesitan. En MDA EXplained (19) se detalla la funcionalidad más relevante para lograr un entorno completo de desarrollo en MDA, que se resumirá a continuación.
 - **Editor de Código:** Realiza las tareas comunes proporcionadas por un entorno de desarrollo interactivo (Interactive Development Environment o IDE), como depuración, compilación y edición de código.
 - **Ficheros de Código:** Aunque se puede considerar al código como un modelo, usualmente se almacena en forma de ficheros de texto. Los ficheros de texto no son un formato que otras herramientas puedan entender. Por consiguiente se necesitan los siguientes dos elementos:
 - **Analizador Gramatical de Ficheros de Texto:** que lea un fichero de texto y almacene el código en forma de modelo en el repositorio de modelos, de manera que otras herramientas lo puedan usar.
 - **Generador de Ficheros de Texto:** que lea el código del repositorio de modelos y produzca un fichero de código basado en texto.
 - **Repositorio de Modelos:** Es la base de datos para los modelos, donde los modelos se almacenan y se recuperan usando XMI, JMI y IDL.
 - **Editor de Modelos:** herramienta CASE para construir y modificar modelos.
 - **Verificador de Modelos:** Los modelos usados para la generación de otros modelos deben estar extremadamente bien definidos. Estas herramientas pueden chequear los modelos para comprobar que cumplen unas determinadas reglas y asegurarse así que puedan ser transformados.
 - **Editor de Definiciones de Transformaciones:** para permitir crear y modificar definiciones de transformaciones.
 - **Repositorio de Definiciones de Transformaciones:** para almacenar las definiciones de transformaciones.

La figura N° 25, muestra un diagrama que resume la funcionalidad deseable en un entorno completo de desarrollo MDA.

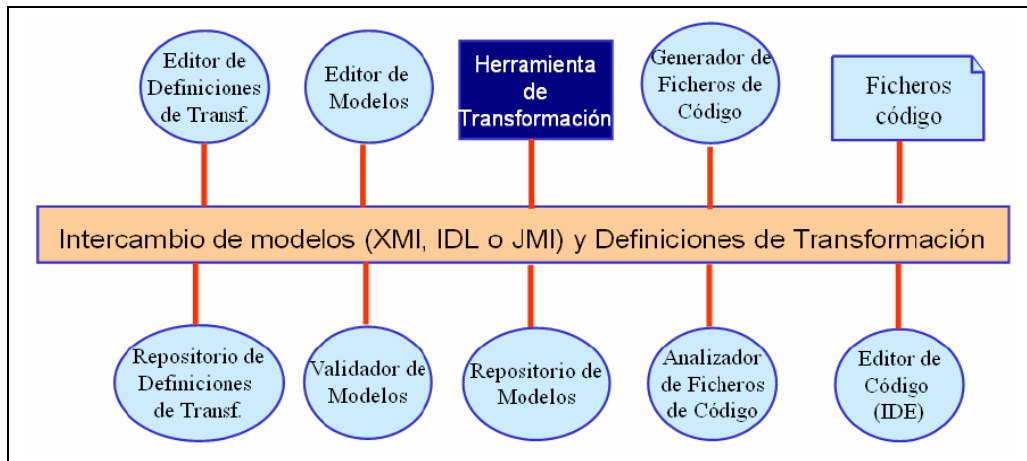


Figura N° 25. Funcionalidad en un Entorno de Desarrollo MDA.
Fuente: (OMG, MOF, 2002)

La mayoría de las herramientas actuales combinan, con mayor o menor medida, un número de estas funciones. Las herramientas CASE tradicionales proporcionan un editor de modelos y un repositorio de modelos. Un generador de código adherido a una herramienta CASE provee de la herramienta de transformación y del editor de definiciones de transformaciones.

Como puede observarse a lo largo de la sección de MDA, la concepción de una nueva forma de construir software requiere de manera indispensable el uso de herramientas de transformación, que son las mismas herramientas CASE, pero soportando el enfoque MDA. Si se toma en cuenta que la Ingeniería de software busca la construcción de productos de calidad, se debe garantizar que todo el proceso de desarrollo exhiba características de calidad, por lo que las herramientas CASE que soporten MDA también deben ser productos de calidad. Entonces pareciera que las herramientas CASE - MDA como producto de software deben ser desarrolladas de una manera impecable, de tal forma que garanticen soportar el desarrollo de un SI, de todo esto surge una gran interrogante que es, ¿las herramientas CASE que soportan MDA son un producto de calidad?, ¿quién mide la calidad de una herramienta CASE - MDA?, ¿los compradores de herramientas CASE - MDA tienen al alcance algún mecanismo para comprobar la calidad de la herramienta CASE adquirida?

Se hace difícil encontrar respuestas a los planteamientos anteriores, por ello en la próxima sección se tratará el tema del Modelo de Calidad para seleccionar Herramientas CASE (MC-CASE) el cual permite medir la calidad de este apreciado producto software.

2.3 MODELO DE CALIDAD PARA SELECCIONAR HERRAMIENTAS CASE (MC-CASE)

La Ingeniería del Software¹ tiene como objetivo principal producir un sistema, aplicación o producto de alta calidad. Para lograr este objetivo, los ingenieros del software deben aplicar métodos efectivos junto con herramientas modernas dentro del contexto de un proceso maduro de desarrollo del software. Además, un buen ingeniero del software (y buenos gestores de la ingeniería del software) deben medir si la alta calidad se va a llevar a cabo. [Pressman, 1998]

En función de poder adoptar productos de calidad Díaz A. en [Díaz, 2000] propone un Modelo de Calidad (MC-CASE) el cual tiene por objetivo evaluar y seleccionar Herramientas CASE; a continuación se presenta la estructura arquitectural del MC-CASE.

2.3.1 ESTRUCTURA ARQUITECTURAL DEL MC-CASE.

De acuerdo con Díaz A en [Díaz., 2000] el modelo se divide en tres grandes **Categorías** que son: Internos, Usuario y Externos; luego cada una de estas categorías esta conformada por un grupo de **Características**, las características se desglosan en **Subcaracterísticas** y éstas a su vez en **Subcaracterísticas Atómica**; finalmente cada subcaracterística atómica posee un grupo de métricas las cuales servirán para medir un aspecto específico de la Herramienta CASE. Esta estructura se puede observar en la Figura N° 26, que se muestra a continuación.

¹ La Ingeniería de Software no trata solamente del desarrollo de productos, ella envuelve la producción de una manera efectiva en cuanto a costos; el desafío para los ingenieros de software es producir software de alta calidad con una cantidad finita de recursos y dentro de una planificación preestablecida. [Sommerville, 1998]

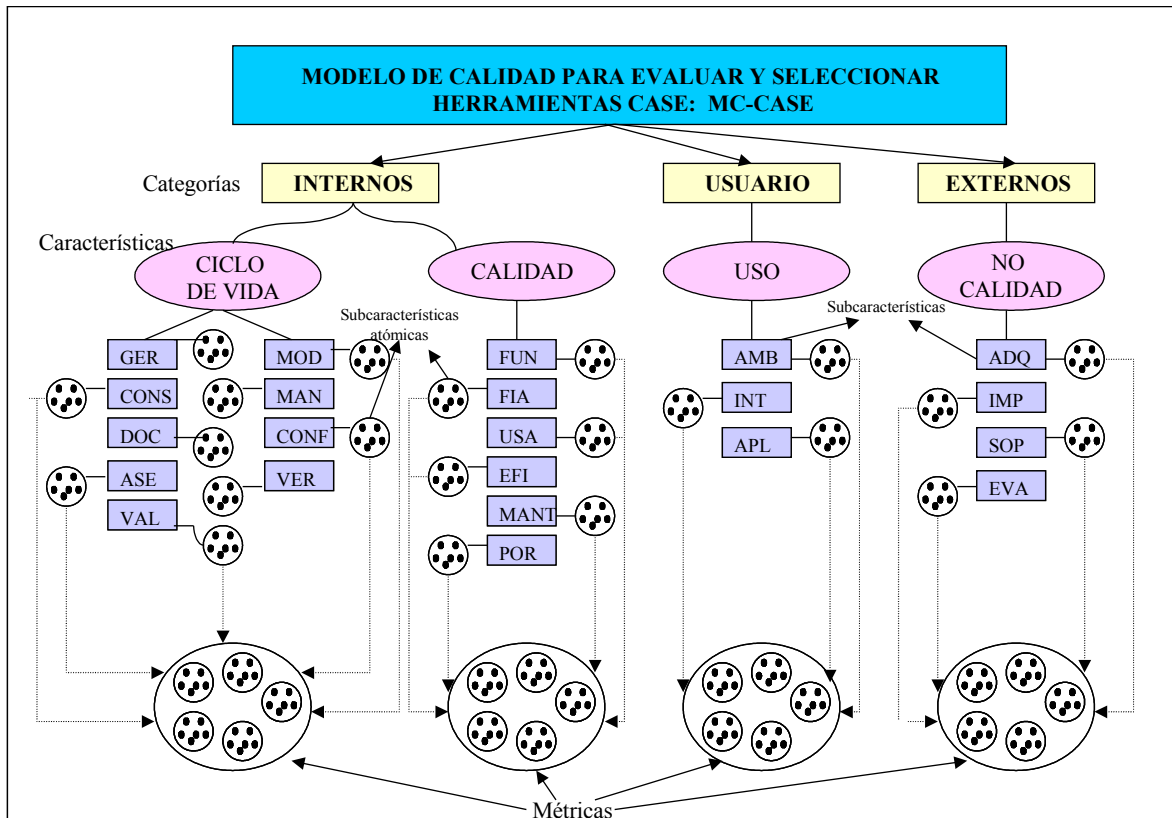


Figura N° 26. Arquitectura del MC-CASE
Fuente: [Díaz, 2000]

Es importante resaltar que las Categorías Internos y Externos son totalmente independientes de las necesidades del usuario, por lo que para este modelo es de vital importancia categorizar dentro de la plataforma base arquitectural los aspectos que dependen exclusivamente de los usuarios.

Dentro de la **Categoría Internos**, se observa que los aspectos propios y exclusivos de la herramienta son las **características relacionadas con el proceso del Ciclo de Vida**, esta característica viene definida por un conjunto de atributos que soportan la existencia de una serie de funciones y sus propiedades para soportar el uso de la herramienta CASE y su relación con los procesos y actividades del ciclo de vida de la ingeniería del software; y las **características relacionadas con la Calidad** describen la calidad de la herramienta en términos de la ISO/IEC 9126. Dentro de las características relacionadas con el Proceso del Ciclo de Vida se encuentran las siguientes Subcaracterísticas que se definen a continuación:

Proceso de Gerencia (GER): Una serie de atributos que se basan en la existencia de un conjunto de funciones y sus propiedades específicas para soportar las actividades del proceso de gerencia;

Proceso de Desarrollo-Modelado (MOD): agrupa un conjunto de atributos que miden las propiedades específicas para soportar las actividades de modelado las cuales pueden ser parte del proceso de desarrollo;

Proceso de Desarrollo-Construcción (CONS): conjunto de atributos que evalúan las propiedades específicas para soportar las actividades de construcción las cuales pueden ser parte del proceso de desarrollo.

Proceso de Mantenimiento (MAN): atributos que se basan sobre la existencia de un conjunto de funciones y sus propiedades específicas para soportar las actividades del proceso de mantenimiento.

Proceso de Documentación (DOC): conjunto de atributos que se basan en la existencia de un conjunto de funciones y sus propiedades específicas para soportar las actividades del proceso de documentación.

Proceso de Gerencia de configuración (CONF): conjunto de atributos que miden las actividades del proceso de gerencia de la configuración.

Proceso de Aseguramiento de la Calidad (ASE): conjunto de atributos que evalúan las actividades del proceso de aseguramiento de la calidad.

Proceso de Verificación (VER): conjunto de atributos que se basan en la existencia de una serie de funciones y sus propiedades específicas para soportar las actividades del proceso de verificación.

Proceso de Validación (VAL): conjunto de atributos que miden propiedades específicas para soportar las actividades del proceso de validación.

Y dentro de las características relacionadas con **Calidad** se encuentran las Subcaracterísticas siguientes:

Funcionalidad (FUN): conjunto de atributos que miden si la herramienta hace lo que especifican sus proveedores.

Fiabilidad (FIA): conjunto de atributos que miden la capacidad del software para mantener su nivel de ejecución bajo condiciones específicas para un periodo de tiempo determinado.

Usabilidad (USA): conjunto de atributos que miden la capacidad del software de ser fácil de usar.

Eficiencia (EFI): conjunto de atributos que se basan en la relación entre en nivel de ejecución del software y la cantidad de los recursos usados, bajo condiciones establecidas.

Mantenibilidad (MANT): conjunto de atributos que se basan en la necesidad de hacer modificaciones.

Portabilidad (POR): conjunto de atributos que se basan en la habilidad del software de ser transferido de un ambiente a otro.

La **Categoría Externos** esta relacionada con aquellas **características de No Calidad** de la Herramienta CASE que no dependen de la herramienta ni del usuario, sino que dependen de agentes externos a ella, entre estos se encuentran las siguientes Subcaracterísticas:

Proceso de Adquisición (ADQ): conjunto de atributos que se basan en la revisión de los procesos necesarios en caso de que la herramienta CASE sea seleccionada por adopción.

Implementación (IMP): conjunto de atributos que evalúan las entregas de la herramienta, instalación y operación.

Indicadores de Soporte (SOP): conjunto de atributos que miden la habilidad del vendedor para proporcionar soporte de la herramienta.

Evaluación o Certificación (EVA): atributos que se basan en la evaluación o certificación del desarrollador o del producto.

Por otro lado la **Categoría Usuario**, esta relacionada con las necesidades propias de cada usuario, esta categoría esta formada por la característica Uso y su cumplimiento es determinante para la adopción de la CASE. Entre las Subcaracterísticas que la conforman están:

Ambiente en el cual opera la herramienta CASE (AMB): conjunto de atributos que se basan en la relación entre la herramienta CASE y su ambiente operacional.

Integrabilidad de la herramienta CASE (INT): conjunto de atributos que miden la habilidad de la herramienta CASE para integrarse e interoperar con otros productos de software, en su ambiente operacional.

Aspectos de la aplicación de la herramienta CASE (APL): conjunto de atributos que se basan en la relación entre la herramienta CASE y los proyectos a los cuales es aplicada, incluyendo el ambiente de sus productos y las características de estos productos.

Las tablas subsiguientes mostrarán, de acuerdo a los aspectos arquitecturales descritos anteriormente, las métricas que se proponen en el Modelo MC-CASE. Cabe mencionar que las definiciones operacionales de cada una de las métricas no se exponen en este trabajo por efectos de dimensiones del mismo, por lo cual se muestran en el Apéndice N° I.

2.3.2. MÉTRICAS DE LA CATEGORÍA INTERNOS.

Las Tablas N° 14 a la N° 22, contienen las métricas que corresponden a la **característica Proceso del Ciclo de Vida**.

1) Subcaracterística: Proceso de Gerencia.

Subcaracterísticas atómicas	Métricas
Estimando costos y planificación	1. Estimación de costos del proyecto 2. Planificación del proyecto

	<p>3. Cantidad de métodos que posee para la estimación de costos y planificación</p> <p>Permite ingresar datos referentes a la estimación de costos y planificación</p> <p>4. Permite modificar los datos referentes a la estimación de costos y la planificación</p>
Planificación	<p>1. Grado de satisfacción de la planificación hecha por la herramienta</p> <p>2. Registro de entregables por tareas</p> <p>3. Registro de tareas o actividades</p> <p>4. Registro de tiempos por tareas</p> <p>5. Registro de recursos asignados por tareas</p>
Seguimiento del proyecto	<p>1. Permite hacer seguimiento al proyecto</p> <p>2. Permite registrar los recursos consumidos</p> <p>3. Número de documentos generados para el seguimiento</p> <p>4. Líneas de código desarrolladas</p> <p>5. Número de casos de pruebas completadas</p> <p>6. Número de defectos detectados</p> <p>7. Número de defectos corregidos</p> <p>Grado de satisfacción del seguimiento del proyecto</p>
Análisis y reportes del estatus del proyecto	<p>1. Emite reportes del seguimiento del proyecto, en formato definido por el usuario</p> <p>2. Emite reportes de la estimación de costos del proyecto</p> <p>3. Emite reportes de los costos reales del proyecto</p> <p>4. Emite reportes de la planificación del proyecto</p> <p>5. Emite reportes de la gerencia del proyecto</p> <p>6. Grado de satisfacción de la generación de reportes del seguimiento del proyecto.</p>
Gerencia del proceso	<p>1. Definición de entradas para la gerencia del proceso</p> <p>2. Definición de recursos</p> <p>3. Definición de fechas límites</p>

Tabla N° 14. Métricas para medir el Proceso de Gerencia
Fuente: [Díaz, 2000]

2) Subcaracterística: Proceso de Desarrollo – Modelado.

Subcaracterísticas atómicas	Métricas
Desarrollo de diagramas	<p>1. Facilidad de diagramación.</p> <p>2. Calidad de la edición gráfica.</p> <p>3. Grado de facilidad de edición.</p> <p>4. Cantidad de tipos de diagramas</p> <p>5. Realiza diagramas</p> <p>6. Grado de satisfacción del desarrollo de diagramas</p> <p>7. Grado de completitud de la caja de herramientas para realizar los diagramas</p> <p>8. Permite importar objetos entre diagramas</p>
Análisis de diagramas	<p>1. Permite variaciones de visión de documentos</p> <p>Facilidad para detectar inconsistencia</p> <p>2. Herramienta de análisis</p> <p>3. Verifica consistencia de los diagramas</p> <p>4. Verifica reglas semánticas entre los diagramas</p> <p>5. Permite el almacenamiento de la información de diseño</p> <p>6. Verifica reglas de diagramación</p> <p>7. Permite la impresión de los diagramas</p> <p>8. Emite reportes sobre los diagramas</p>
Soporte de especificación de requerimientos	<p>1. Posee un lenguaje formal para representar los requerimientos de datos</p> <p>2. Emite reportes de los requerimientos especificados</p> <p>3. Permite la clasificación de los requerimientos</p> <p>4. Grado de satisfacción de la especificación de requerimientos</p>
Soporte de especificación de diseño	<p>1. Grado de consistencia de la diagramación respecto a la metodología adoptada por la organización.</p> <p>2. Grado de ajuste a las metodologías en la etapa de diseño</p> <p>3. Facilidad para detectar inconsistencia.</p>

	<ol style="list-style-type: none"> 4. Posee un lenguaje formal para representar el diseño de datos 5. Emite reportes de los datos del diseño 6. Clasifica los tipos de datos del diseño 7. Grado de satisfacción de las especificaciones del diseño
Modelamiento de especificaciones de estructura de los procesos	<ol style="list-style-type: none"> 1. Permite moldear procesos 2. Permite modelar la estructura de datos asociada a los procesos 3. Grado de satisfacción de la edición de modelos de especificación de procesos
Simulación	<ol style="list-style-type: none"> 1. Permite hacer simulaciones de la utilidad operacional. 2. Permite hacer simulaciones de la interfaz con el operador 3. Permite hacer simulaciones del tiempo de respuesta 4. Permite hacer simulaciones del desempeño 5. Grado de satisfacción con respecto a las simulaciones hechas por la herramienta.
Prototipo	<ol style="list-style-type: none"> 1. Permite la realización de prototipo. 2. Nivel de generación de prototipo 3. Grado de satisfacción de la generación de prototipo
Modelamiento de la interfaz humana	<ol style="list-style-type: none"> 1. Permite modelar las interfaces. (menús, ventanas, reportes, etc.) 2. Grado de satisfacción con respecto al modelamiento de interfaces. 3. Facilidad para modelar interfaces 4. Riqueza de la paleta

Tabla N° 15. Métricas para medir el Proceso de Desarrollo - Modelado
Fuente: [Díaz, 2000]

3) Subcaracterística: Proceso de Desarrollo – Construcción.

Subcaracterísticas atómicas	Métricas
Generación de código	<ol style="list-style-type: none"> 1. Grado de completitud del código generado 2. Calidad del código generado 3. Cantidad de lenguajes en que genera código 4. Generación de códigos 5. Ejecutables para distintas plataformas 6. Tasa de cobertura del código generado
Generación de esquemas de bases de datos	<ol style="list-style-type: none"> 1. Generación de los esquemas de bases de datos (definir esquemas). 2. Grado de satisfacción de los esquemas de bases de datos generados por la herramienta. 3. Número de manejadores de bases de datos en que genera esquemas 4. Tasa de cobertura de la generación de esquemas
Generación de pantallas	<ol style="list-style-type: none"> 1. Generador de pantallas. 2. Bondad de diseño visual. 3. Genera pantallas 4. Grado de satisfacción de la generación de pantallas 5. Facilidad con que se generan las pantallas
Generación de reportes	<ol style="list-style-type: none"> 1. Permite generar reportes 2. Permite definir reportes con formatos del usuario 3. Calidad de los reportes que genera 4. Grado de completitud de los reportes
Compilación	<ol style="list-style-type: none"> 1. Permite compilar código fuente 2. Cantidad de lenguajes en que compila código fuente 3. Grado de satisfacción del proceso de compilación realizado por la herramienta 4. Tasa de cobertura de los lenguajes en que compila código fuente 5. Emite reportes acerca de la compilación
Edición dirigida por la sintaxis	<ol style="list-style-type: none"> 1. Provee soporte de sintaxis para la edición del código fuente. (en varios lenguajes de programación). 2. Grado de satisfacción del soporte de sintaxis de la herramienta 3. Cantidad de lenguajes en que soporta la sintaxis del código fuente
Depurador	<ol style="list-style-type: none"> 1. La herramienta identifica y aísla los errores detectados por el compilador 2. Permite definir policias en el código a depurar 3. Permite seleccionar variables para observar los valores que toma 4. Grado de satisfacción de la identificación de los errores

Tabla N° 16. Métricas para medir el Proceso de Desarrollo - Construcción
Fuente: [Díaz, 2000]

4) Subcaracterística: Proceso de Mantenimiento.

Subcaracterísticas atómicas	Métricas
Entendimiento del problema	<ol style="list-style-type: none"> 1. Emite reportes de fallas 2. Emite reportes de defectos 3. Restringe las operaciones cuando se presenta una falla 4. Tiene la capacidad de determinar si los problemas han sido resueltos 5. Tiene la capacidad de aprender de los defectos y fallas encontradas
Localización	<ol style="list-style-type: none"> 1. Identifica el problema que requiera modificación. 2. Identifica la ubicación del problema 3. Grado de satisfacción de la localización de errores
Análisis de impacto	<ol style="list-style-type: none"> 1. Identifica las consecuencias de hacer cambios 2. Grado de satisfacción de la identificación de las consecuencias de hacer cambios 3. Difunde los cambios 4. Grado de satisfacción con los mecanismos de difusión utilizados por la herramienta
Ingeniería de reverso de datos	<ol style="list-style-type: none"> 1. Proceso de ingeniería de reverso de datos 2. Calidad del producto de la Ingeniería de reverso de datos 3. Tasa de cobertura de los lenguajes de ingeniería de reverso de datos 4. Cantidad de lenguajes soportados por la ingeniería de reverso de datos 5. Permite hacer ingeniería de reverso de datos
Ingeniería de reverso de procesos/procedimientos	<ol style="list-style-type: none"> 1. Proceso de ingeniería de reverso de procesos 2. Calidad del producto de la Ingeniería de reverso de procesos 3. Tasa de cobertura de los lenguajes de ingeniería de reverso de procesos 4. Cantidad de lenguajes soportados por la ingeniería de reverso de procesos 5. Permite hacer ingeniería de reverso de procesos
Reestructuración del código fuente	<ol style="list-style-type: none"> 1. Modifica el formato y estructura del código fuente en uno o mas lenguajes. (reducir el tamaño del código, reducir el tiempo de ejecución). 2. Grado de satisfacción de la modificación y reestructuración del código fuente 3. Número de lenguajes en que reestructura el código fuente 4. Tasa de cobertura de la reestructuración de código fuente
Traducción del código fuente	<ol style="list-style-type: none"> 1. Permite traducir código fuente de un lenguaje a otro 2. Número de lenguajes en que traduce el código fuente 3. Tasa de cobertura de traducción del código fuente 4. Grado de satisfacción de la traducción del código fuente

Tabla N° 17. Métricas para medir el Proceso de Mantenimiento
Fuente: [Díaz, 2000]

5) Subcaracterística: Proceso de Documentación.

Subcaracterísticas atómicas	Métricas
Edición de texto	<ol style="list-style-type: none"> 1. Facilidad de documentación 2. Facilidad para editar texto 3. Grado de satisfacción de la edición de texto 4. Discrimina instrucciones de variables 5. Permite la edición automática de texto
Edición gráfica	<ol style="list-style-type: none"> 1. Grado de satisfacción de la edición gráfica 2. Calidad de la edición gráfica 3. Grado de facilidad de edición gráfica 4. Grado de completitud de la caja de herramientas para editar datos gráficamente
Edición basada en formas	<ol style="list-style-type: none"> 1. Permite la edición a través de formas 2. Permite la edición a través de formas definidas por el usuario 3. Grado de satisfacción de la edición basada en formas
Publicidad	<ol style="list-style-type: none"> 1. Cuenta con mecanismos tales como e-mail, groupware, para difundir la documentación

	2. Grado de satisfacción de la difusión de la documentación
Soporte de hipertexto	1. La herramienta soporta la edición de hipertexto 2. La herramienta soporta todas las funciones de hipertexto 3. Grado de satisfacción del soporte de hipertexto.
Manejando variantes	1. Facilidad de cambio en producto terminado 2. Permite identificar las diferencias entre las versiones de un documento 3. Permite llevar un registro para cada versión
Extracción de datos automática y generación de documentos	1. Permite la recuperación de las especificaciones de datos textuales y gráficos. 2. Genera documentos de las especificaciones de los datos 3. Grado de satisfacción de la generación de documentos 4. Número de especificaciones de datos textuales que recupera 5. Número de especificaciones de datos gráficos que recupera 6. Permite personalizar los documentos 7. Grado de completitud de la recuperación de los datos

Tabla N° 18. Métricas para medir el Proceso de Documentación
Fuente: [Díaz, 2000]

6) Subcaracterística: Proceso de Gerencia de la Configuración.

Subcaracterísticas atómicas	Métricas
Control de acceso	1. Permite establecer restricciones para acceder los elementos de datos 2. Permite llevar registros de las modificaciones realizadas a los elementos de datos 3. Permite diferenciar las autoridades 4. Grado de satisfacción del control sobre el acceso que posee la herramienta
Búsqueda de modificaciones	1. Permite llevar registros de las modificaciones realizadas al sistema en general 2. Permite realizar búsquedas de las modificaciones de acuerdo a diferentes criterios 3. Grado de satisfacción de la búsqueda de modificaciones
Definición y gerencia de versiones múltiples	1. Facilidad de integración de versiones 2. Permite el registro de versiones 3. Facilidad de búsqueda de las versiones
Estatus de configuración	1. Genera reportes donde se resume la información de las configuraciones gerenciadas en una sesión de trabajo 2. Grado de completitud de los reportes de configuración generados 3. Grado de satisfacción de los reportes de configuración generados
Generación de Versiones	1. Permite la creación de versiones de un producto a partir de otro 2. Grado de satisfacción de la generación de versiones 3. Registra las versiones creadas 4. Facilidad de búsqueda de versiones
Capacidad de archivo	1. Graba automáticamente los productos generados 2. Facilidad de búsqueda de los productos generados

Tabla N° 19. Métricas para medir el Proceso de Gerencia de la Configuración
Fuente: [Díaz, 2000]

7) Subcaracterística: Proceso de Aseguramiento de la Calidad.

Subcaracterísticas atómicas	Métricas
Gerencia de la calidad de los datos	1. Proporciona planes de auditoria y revisiones de resultados o productos 2. Proporciona revisiones de resultados o productos
Gerencia de riesgos	1. Permite definir y modificar los riesgos del proyecto 2. Permite elaborar listas de chequeo de riesgos 3. Permite clasificar los riesgos 4. Emite reportes de los riesgos del proyecto 5. Grado de satisfacción de la gerencia de los riesgos

Tabla N° 20. Métricas para medir el Proceso de Aseguramiento de la Calidad
Fuente: [Díaz, 2000]

8) Subcaracterística: Proceso de Verificación.

Subcaracterísticas atómicas	Métricas
Análisis de especificación de trazabilidad	<ol style="list-style-type: none"> 1. Facilidad para detectar inconsistencia 2. Verifica consistencia entre los modelos 3. Grado de satisfacción de la verificación de consistencia realizada 4. Permite trazar los cambios de un modelo a otro 5. Facilidad para trazar los cambios de un modelo a otro
Análisis de especificación	<ol style="list-style-type: none"> 1. Emite reportes de los modelos de análisis y sus elementos 2. Emite reportes de los modelos de diseño y sus elementos
Análisis de código fuente	<ol style="list-style-type: none"> 1. Permite la definición de métricas de análisis de código fuente 2. Emite reportes sobre las métricas de análisis de código fuente 3. Grado de satisfacción de los reportes sobre métricas para analizar el código fuente

Tabla N° 21. Métricas para medir el Proceso de Verificación

Fuente: [Díaz, 2000]

9) Subcaracterística: Proceso de Validación.

Subcaracterísticas atómicas	Métricas
Prueba de técnicas de correctitud	<ol style="list-style-type: none"> 1. Permite constatar si se cumplen las restricciones (precondiciones y postcondiciones del software)
Análisis de fallas	<ol style="list-style-type: none"> 1. Frecuencia de fallas por tiempo que presenta el software 2. Rapidez de recuperación después de haberse presentado la falla 3. Lleva un registro de las fallas presentadas 4. Lleva un registro de la causa que genera la falla 5. Emite reportes de las fallas presentadas
Análisis de defectos	<ol style="list-style-type: none"> 1. Número de defectos encontrados 2. Lleva un registro de los defectos encontrados 3. Lleva un registro de las consecuencias de las fallas 4. Lleva un registro de las consecuencias de los defectos
Caso de prueba y espera del resultado de entrada	<ol style="list-style-type: none"> 1. Facilidad para ingresar casos de prueba 2. Facilidad para comparar con los resultados esperados 3. Permite llevar un registro de los casos de prueba y sus resultados 4. Grado de satisfacción de la comparación con los resultados esperados
Caso de prueba y generación de resultados esperados	<ol style="list-style-type: none"> 1. Genera automáticamente casos de prueba 2. Genera automáticamente los resultados esperados 3. Grado de satisfacción de la generación de los casos de prueba 4. Grado de satisfacción de los resultados esperados
Prueba de trazabilidad	<ol style="list-style-type: none"> 1. Verifica consistencia entre las pruebas y los datos de prueba 2. Grado de satisfacción de la verificación de consistencia entre las pruebas y los datos de prueba
Instrumentación de código fuente	<ol style="list-style-type: none"> 1. Ejecuta el código fuente 2. Grado de completitud de la ejecución del código fuente 3. Grado de satisfacción de la ejecución del código fuente
Captura de la entrada y repetición	<ol style="list-style-type: none"> 1. Guarda datos que hayan sido editados para casos de prueba 2. Permite reutilizar los datos de prueba
Manejo de pruebas	<ol style="list-style-type: none"> 1. Permite repetir casos de prueba previamente ejecutados 2. Permite realizar búsqueda de pruebas 3. Permite hacer planificación de pruebas
Análisis en tiempo de corrida	<ol style="list-style-type: none"> 1. Genera reportes del desempeño del programa durante la ejecución del mismo 2. Permite ingresar indicadores de desempeño de ejecución del programa 3. Grado de satisfacción de los reportes de desempeño generados
Análisis de fiabilidad	<ol style="list-style-type: none"> 1. Permite definir indicadores de confiabilidad 2. Genera reportes de los indicadores de confiabilidad
Análisis de alcance de pruebas	<ol style="list-style-type: none"> 1. Genera reportes del alcance de las pruebas 2. Grado de satisfacción del alcance de las pruebas
Gerencia del proceso de prueba	<ol style="list-style-type: none"> 1. Permite formular un plan para la ejecución de las pruebas 2. Definición y modificación de prioridades en la ejecución de las pruebas

	<ol style="list-style-type: none"> 3. Definición y modificación de actividades de las pruebas 4. Definición y modificación de tiempos en los cuales se realizaran las pruebas 5. Asignación de recursos a las pruebas 6. Grado de satisfacción de la gerencia del proceso de prueba
Prueba de regresión	<ol style="list-style-type: none"> 1. Compara los resultados de diferentes casos de prueba 2. Genera reportes de las pruebas de regresión 3. Grado de completitud de los reportes de las pruebas de regresión 4. Grado de satisfacción con los reportes de las pruebas de regresión
Chequeo automático de resultados	<ol style="list-style-type: none"> 1. Chequeo o comparación de los resultados de las pruebas con los resultados esperados 2. Generación de reportes del chequeo de los resultados de las pruebas 3. Grado de satisfacción del chequeo de los resultados de las pruebas 4. Grado de completitud de los reportes del chequeo de los resultados de las pruebas
Análisis de pruebas estadísticas	<ol style="list-style-type: none"> 1. Permite definir pruebas estadísticas de los resultados 2. Genera reportes estadísticos de los resultados de las pruebas 3. Grado de satisfacción de las pruebas estadísticas de los resultados
Simulación en ambiente de operaciones	<ol style="list-style-type: none"> 1. Genera simulaciones en el ambiente real de operación 2. Permite definir las pautas de la simulación 3. Grado de satisfacción de las simulaciones en ambiente real
Prueba de integración	<ol style="list-style-type: none"> 1. Permite registrar las actividades de integración de software 2. Registra los tiempos de las actividades de integración 3. Registra la secuencia de las actividades de integración 4. Emite reportes de las pruebas de integración 5. Grado de satisfacción de los reportes de las pruebas de integración. 6. Grado de completitud de los reportes de las pruebas de integración

Tabla N° 22. Métricas para medir el Proceso de Validación

Fuente: [Díaz, 2000]

Próximamente se presentan las Tablas N° 23 a la N° 28, que contienen las métricas relacionadas con la **Características Calidad**.

1) Subcaracterística: Funcionalidad.

Subcaracterísticas atómicas	Métricas
Seguridad	<ol style="list-style-type: none"> 1. Mecanismo de manejo de niveles de seguridad
Exactitud	<ol style="list-style-type: none"> 1. Grado de claridad de los mensajes 2. Localización rápida de opciones 3. Grado de versatilidad de la navegación
Normativas regulatorias	<ol style="list-style-type: none"> 1. Genera productos acordes con las normas legales del país 2. Grado de completitud en que los productos responden a las normas legales del país
Normativas técnicas	<ol style="list-style-type: none"> 1. Genera productos acordes con las normas técnicas del país

Tabla N° 23. Métricas para medir la Funcionalidad

Fuente: [Díaz, 2000]

2) Subcaracterística: Fiabilidad.

Subcaracterísticas atómicas	Métricas
Integridad de los datos	<ol style="list-style-type: none"> 1. Permite accesos no autorizados 2. Permite jerarquizar los accesos 3. Tiene mecanismos de "backup" o replicación de los datos 4. Permite recuperar los datos almacenados
Backup automático	<ol style="list-style-type: none"> 1. Realiza copias de seguridad automáticamente. 2. Grado de satisfacción de las copias de seguridad realizadas por la herramienta
Manejo de errores	<ol style="list-style-type: none"> 1. Capacidad de detectar comportamientos anormales 2. Guarda los trabajos en el momento de alguna interrupción

	3. Los mensajes de error que genera están acordes a los errores presentados
Tolerancia de fallas	<ol style="list-style-type: none"> 1. Conservación del nivel de desempeño al presentarse fallas (hardware, software, red, etc.) 2. Número de fallas presentadas durante una jornada de trabajo 3. Frecuencia con la que se presentan las fallas 4. Grado de satisfacción de la tolerancia a fallas de la h. CASE
Recuperabilidad	<ol style="list-style-type: none"> 1. Recupera su nivel de desempeño después de presentada una falla 2. Recupera los datos afectados durante la falla 3. Tiempo que requiere para recuperar el nivel de desempeño

Tabla N° 24. Métricas para medir la Fiabilidad

Fuente: [Díaz, 2000]

3) Subcaracterística: Usabilidad.

Subcaracterísticas atómicas	Métricas
Amigabilidad del usuario	<ol style="list-style-type: none"> 1. La herramienta es intuitiva 2. Facilidad de uso
Guía del usuario	<ol style="list-style-type: none"> 1. Presencia de metáforas 2. Rapidez del sistema de ayuda
Homogeneidad	<ol style="list-style-type: none"> 1. Facilidad de detectar inconsistencia
Adaptabilidad	<ol style="list-style-type: none"> 1. Capacidad de parametrización <p>La interfaz de la herramienta se adapta a diferentes hábitos y modos culturales</p> <ol style="list-style-type: none"> 2. Permite la parametrización o personalización
Claridad de control	<ol style="list-style-type: none"> 1. Grado de versatilidad de la navegación. 2. Localización rápida de opciones 3. Sensibilidad de contexto
Manejo de errores	<ol style="list-style-type: none"> 1. Grado de claridad de los mensajes. 2. Permite la corrección de los errores evitando datos incorrectos y cambios en los procesos
Concisión	<ol style="list-style-type: none"> 1. Presencia de metáforas. 2. Sensibilidad de contexto
Fácil de aprender	<ol style="list-style-type: none"> 1. Horas de entrenamiento requeridas para lograr la experticia necesaria para el uso completo de la herramienta, en una persona que desconozca la misma. 2. La herramienta es intuitiva
Calidad de documentación de la herramienta	<ol style="list-style-type: none"> 1. Grado de completitud del material de apoyo según la funcionalidad de la herramienta. 2. Calidad de la clasificación de los temas. 3. Profundidad en el tratamiento de los temas. 4. Consistencia del material. 5. Tasa de cobertura del material de apoyo en cuanto a su presentación. 6. Tasa de cobertura del material de apoyo en cuanto a los usuarios a los que esta dirigido. 7. Variedad del material de apoyo en cuanto a los usuarios a los que esta dirigido. 8. Variedad del material de apoyo de acuerdo a su presentación.
Fácil de instalar	<ol style="list-style-type: none"> 1. Facilidad para instalar la herramienta 2. Tiempo real de instalación 3. Tasa del tiempo promedio de instalación 4. Se requiere al proveedor para la instalación 5. Se requiere de material de apoyo para la instalación

Tabla N° 25. Métricas para medir la Usabilidad

Fuente: [Díaz, 2000]

4) Subcaracterística: Eficiencia.

Subcaracterísticas atómicas	Métricas
Tiempo de respuesta aceptable	<ol style="list-style-type: none"> 1. Grado de satisfacción del tiempo de respuesta.
Requerimientos de almacenamiento de datos	<ol style="list-style-type: none"> 1. Cantidad de memoria secundaria necesaria para almacenar a la herramienta y a las bases de datos requeridas y generadas por ella

	2. La plataforma de hardware de la organización cuenta con la cantidad de memoria secundaria necesaria
Capacidad de memoria aceptable	1. Cantidad de memoria RAM requerida para cargar y operar la herramienta satisfactoriamente 2. Grado de desempeño de la herramienta con respecto a la cantidad de memoria RAM disponible en el ambiente de trabajo de la organización
Velocidad de procesamiento aceptable	1. Tipo y velocidad del procesador requerido para operar con la herramienta de manera satisfactoria 2. Aceptabilidad del desempeño de la herramienta durante la ejecución del proyecto 3. Grado de rapidez con que se accesan las opciones dentro de la herramienta en ejecución

Tabla N° 26. Métricas para medir la Eficiencia
Fuente: [Díaz, 2000]

5) Subcaracterística: Mantenibilidad.

Subcaracterísticas atómicas	Métricas
Soporte del vendedor	1. Soporte técnico permanente. 2. Soportes adicionales (web, mail, etc.). 3. Grado de satisfacción del tiempo promedio de respuesta ante una solicitud de respuesta 4. Tasa de cobertura del horario de atención del soporte técnico. 5. Horas semanales de soporte técnico.
Habilidad de la herramienta para hacer cambios en la metodología	1. El vendedor puede modificar la herramienta para mantener el soporte a las metodologías
Actualización	1. Grado de mejoramiento de la herramientas
Expansibilidad	1. El desarrollador de la H. CASE genera "upgrade" de la herramienta 2. Grado de facilidad para obtener los "upgrade" de la herramienta 3. Frecuencia con la cual el proveedor lanza al mercado los "upgrade"

Tabla N° 27. Métricas para medir la Mantenibilidad
Fuente: [Díaz, 2000]

6) Subcaracterística: Portabilidad.

Subcaracterísticas atómicas	Métricas
Portabilidad a diferentes plataformas de hardware	1. Portabilidad a otras plataformas de hardware Los productos de la h. CASE se pueden ejecutar para distintas plataformas de hardware
Compatibilidad con diferentes sistemas operativos	1. La herramienta puede ejecutarse sobre varias versiones de un mismo sistema operativo 2. Se ejecuta sobre varios sistemas operativos 3. Cantidad de sistemas operativos sobre los cuales opera la herramienta 4. Tasa de cobertura de los sistemas operativos requeridos por la organización
Habilidad para mover datos entre versiones de la herramienta	1. La herramienta puede usar datos provenientes de otra versión de la herramienta 2. Permite guardar como otra versión de la h. CASE
Portabilidad con sistemas Windows	1. La herramienta se ejecuta bajo plataformas Windows 2. Versiones de Windows que soportan a la h. CASE

Tabla N° 28. Métricas para medir la Portabilidad
Fuente: [Díaz, 2000]

2.3.3. MÉTRICAS DE LA CATEGORÍA USUARIO.

Las Tablas N° 29 a la N° 31, que se presentan a continuación resumen la adaptación y la propuesta de indicadores correspondientes a la **Característica Funcionalidad – Uso de la Herramienta CASE**

1) Subcaracterística: Ambiente en el cual opera la herramienta CASE.

Subcaracterísticas atómicas	Métricas
Características de hardware requeridas por la herramienta	1. La organización o el usuario posee la plataforma de hardware necesaria para soportar a la h. CASE
Ambiente de software requerido por la herramienta	1. Cantidad de plataformas de software que soportan a la herramienta. 2. Tasa de la capacidad de la plataforma de la organización para soportar la herramienta 3. Cantidad de plataformas de software que posee la organización
Repositorio de software	1. Repositorio común
Ambiente físico de la herramienta	1. Requiere actualizar un repositorio en múltiples sitios 2. Permite distribuir geográficamente los nodos de una red 3. Número de usuarios concurrentes 4. Grado de satisfacción en el manejo de las concurrencias 5. Tasa de cobertura de la concurrencia requerida

Tabla N° 29. Métricas para medir el Ambiente en el cual opera la Herramienta CASE

Fuente: [Díaz, 2000]

2) Subcaracterística: Integrabilidad de la herramienta CASE.

Subcaracterísticas atómicas	Métricas
Compatibilidad con los elementos del ambiente	1. Grado de portabilidad. 2. Portabilidad a otras plataformas.
Integración de data	1. Calidad de la integración. 2. Es una herramienta integrada. 3. Integración de data.
Integración de control	1. Integración de actividades.
Integración de presentación	1. Integración de interface.
Acceso a metadatos	1. Repositorio común.

Tabla N° 30. Métricas para medir la Integrabilidad de la Herramienta CASE

Fuente: [Díaz, 2000]

3) Subcaracterística: Aspectos de la aplicación de la herramienta CASE.

Subcaracterísticas atómicas	Métricas
Ambiente de hardware y software de los productos de la herramienta	1. Los productos de la herramienta requieren de un ambiente de hardware y software especial 2. La organización posee la plataforma de hardware que requiere la h. CASE 3. La organización posee la plataforma de software que requiere la h. CASE 4. Grado de satisfacción del desempeño de los productos de la herramienta en el ambiente de software 5. Grado de satisfacción del desempeño de los productos de la herramienta en el ambiente de hardware
Conformidad de estándares – Productos de la herramienta-	1. La herramienta genera productos que están acordes con los estándares establecidos por la organización 2. La herramienta permite definir los estándares que deben llevar los productos de la organización 3. Total de etapas soportadas 4. Tasa de cobertura de las etapas requeridas 5. Grado de consistencia de la diagramación respecto a la metodología adoptada por la organización
Dominio de la aplicación	1. Soporta el tipo de aplicación que el usuario desee desarrollar 2. Grado de satisfacción del soporte de la herramienta con respecto a la aplicación requerida por la organización
Tamaño de la aplicación soportada	1. Soporta el tamaño de la aplicación que el usuario desea desarrollar
Lenguajes soportados	1. Cantidad de lenguajes en que genera código 2. Tasa de cobertura del código generado

	3. Cantidad de lenguajes soportados por la ingeniería de reverso 4. Tasa de cobertura de los lenguajes de ingeniería de reverso
Bases de datos soportadas	1. Tasa de la manipulación de los manejadores de bases de datos requeridos por la organización. 2. Total de manejadores de bases de datos que maneja.
Metodologías soportadas	1. Tasa de cobertura de las metodologías requeridas. 2. Total de metodologías soportadas.
Internacionalización	1. Varía la filosofía de trabajo de acuerdo con la cultura o país 2. Grado de satisfacción de la variación de filosofía de trabajo 3. Cambia el idioma de acuerdo a la cultura o país

Tabla N° 31. Métricas para medir los Aspectos de la aplicación de la Herramienta CASE
Fuente: [Díaz, 2000]

2.3.4. MÉTRICAS DE LA CATEGORÍA EXTERNOS.

Por último se presentan las Tablas N° 32, 33, 34 y 35, que contienen las métricas para la **Característica de No Calidad**.

1) Subcaracterística: Proceso de Adquisición.

Subcaracterísticas atómicas	Métricas
Costo de implementación de la herramienta	1. Costos de adquisición. 2. Costos de entrenamiento. 3. Tasa de cobertura del costo de adquisición de la herramienta. 4. Costo de hardware 5. Costo de software 6. Tasa de cobertura de costo de hardware 7. Tasa de cobertura de costo de software 8. Costos de instalación 9. Costo de mantenimiento inicial 10. Tasa de cobertura de los costos de instalación 11. Tasa de cobertura de los costos de mantenimiento inicial
Políticas de licencias	1. El proveedor establece políticas de licencias con respecto a la herramienta. 2. El proveedor otorga el derecho a copias 3. El proveedor restringe algún uso de la herramienta. La herramienta tiene garantía
Restricciones de exportación	1. El proveedor restringe la exportación de la herramienta

Tabla N° 32. Métricas para medir el Proceso de Adquisición
Fuente: [Díaz, 2000]

2) Subcaracterística: Implementación.

Subcaracterísticas atómicas	Métricas
Efectividad del costo	1. Es la herramienta rentable para la organización. (análisis costo – beneficio)
Restricciones de desarrollo/entregas	1. La herramienta es intuitiva. 2. Tiene tutorial 3. Grado de satisfacción del servicio del tutorial 4. La herramienta permite ir más allá del producto requerido por la organización
Actividades requeridas por los usuarios de la organización	1. La herramienta soporta actividades especiales con respecto al ambiente del usuario
Necesidades de infraestructura	1. Requiere de condiciones de infraestructura especiales para su operación. (Espacio del piso, condiciones eléctricas, condiciones de temperatura, etc.)

Tabla N° 33. Métricas para medir el Proceso de Mantenimiento
Fuente: [Díaz, 2000]

3) Subcaracterística: Indicadores de Soporte.

Subcaracterísticas atómicas	Métricas
Perfil del vendedor	<ol style="list-style-type: none"> 1. Grado de bienestar financiero de la empresa desarrolladora. 2. Tiempo de la empresa desarrolladora en el mercado con esa herramienta. 3. Participación en el mercado con esta herramienta.
Perfil del producto	<ol style="list-style-type: none"> 1. Publicaciones expertas positivas. 2. Desempeño de la herramienta en otras empresas. 3. Experiencia positiva con otras empresas
Disponibilidad de entrenamiento	<ol style="list-style-type: none"> 1. Cantidad de cursos básicos al año. 2. Cantidad de cursos avanzados al año. 3. Cantidad de tipos de cursos. 4. Cobertura de cursos básicos al año. 5. Cobertura de cursos avanzados al año.

Tabla N° 34. Métricas para medir los Indicadores de Soporte

Fuente: [Díaz, 2000]

4) Subcaracterística: Evaluación o Certificación.

Subcaracterísticas atómicas	Métricas
Evaluación del desarrollador o Certificación	<ol style="list-style-type: none"> 1. Posee certificaciones 2. La organización que evalúa o certifica es reconocida
Certificación del producto	<ol style="list-style-type: none"> 1. La herramienta posee certificaciones 2. La organización que certificó es reconocida

Tabla N° 35. Métricas para medir la Evaluación o Certificación

Fuente: [Díaz, 2000]

Con esta sección se culmina la descripción del **Modelo de Calidad para Evaluar y Seleccionar Herramientas CASE (MC-CASE)**, en su primera versión.

Sobre la base de la descripción hecha del Modelo de Calidad para Evaluar Herramientas CASE se puede observar que este modelo carece de los aspectos explícitos que permitan evaluar si la herramienta CASE es una herramienta MDA. Si bien la Característica Ciclo de Vida contempla unas subcaracterísticas como Modelado y Construcción, las métricas que de ella se desprenden no parecen ser suficientes para considerar que evalúan a una herramienta MDA, por lo tanto esto sirve de inspiración para obtener el objetivo central del presente trabajo de investigación, el cual es proponer un Modelo de Calidad para Evaluar Herramientas CASE – MDA. En el próximo capítulo se describe la metodología que permitirá lograr la concepción de dicho modelo.

CAPÍTULO III

MARCO METODOLÓGICO

Una vez que se han desarrollado las bases teóricas sobre las cuales se apoya esta Investigación, es preciso seleccionar los distintos métodos y técnicas que harán posible obtener la información requerida. Según Francois Chazel et al., citado en [Balestrini, 1998], “...*tales métodos no son fórmulas rígidas susceptibles de ser aplicadas con mayor o menor fortuna, sino que forman parte de un razonamiento... De este modo las conclusiones de orden teórico parecen como inseparables del método que ha permitido establecerlas; constituyen la prueba de su fecundidad y de sus límites...*”.

En toda investigación científica se hace necesario que los hechos estudiados, así como las relaciones que se establecen entre estos, los resultados obtenidos y las evidencias significativas encontradas en relación al problema investigado, además de los nuevos conocimientos que es posible situar; reúnan las condiciones de confiabilidad, objetividad y validez interna; para lo cual se requiere delimitar los pasos de orden metodológico a través de los cuales se intenta dar respuesta a las interrogantes objeto de la investigación.

En consecuencia, el Marco Metodológico de la presente investigación donde se quiere proponer un Meta-Modelo de Calidad para Evaluar Herramientas CASE con soporte a MDA, es la instancia sobre la cual se construyó el conjunto de pasos que se emplearon en el proceso de recolección de los datos requeridos en la investigación. En tal sentido, se conceptualizará al tipo de estudio o investigación que en este caso se trata de una **investigación Básica**; para luego dar paso a un esquema donde se resumen las actividades que se siguieron para realizar la presente investigación.

3.1. INVESTIGACIÓN DE TIPO BÁSICA

De acuerdo con el objetivo de esta investigación y según Zorrilla [Zorrilla 1993], La investigación básica denominada también pura o fundamental, busca el progreso científico, acrecentar los conocimientos teóricos, sin interesarse directamente en sus posibles aplicaciones o consecuencias prácticas; es más formal y persigue las generalizaciones con vistas al desarrollo de una teoría basada en principios y leyes.

3.2. METODOLOGÍA APLICADA

Sobre la base teórica relativa a la investigación de tipo Básica, en el presente trabajo se aplicaron los pasos que se muestran en la Figura N° 27.

Los pasos que se seguirán para desarrollar este trabajo de investigación se describirán gráficamente a través de un diagrama de Actividad de UML, el cual permite mostrar de manera dinámica las actividades que deben ejecutarse para completar el objetivo central del trabajo de investigación.

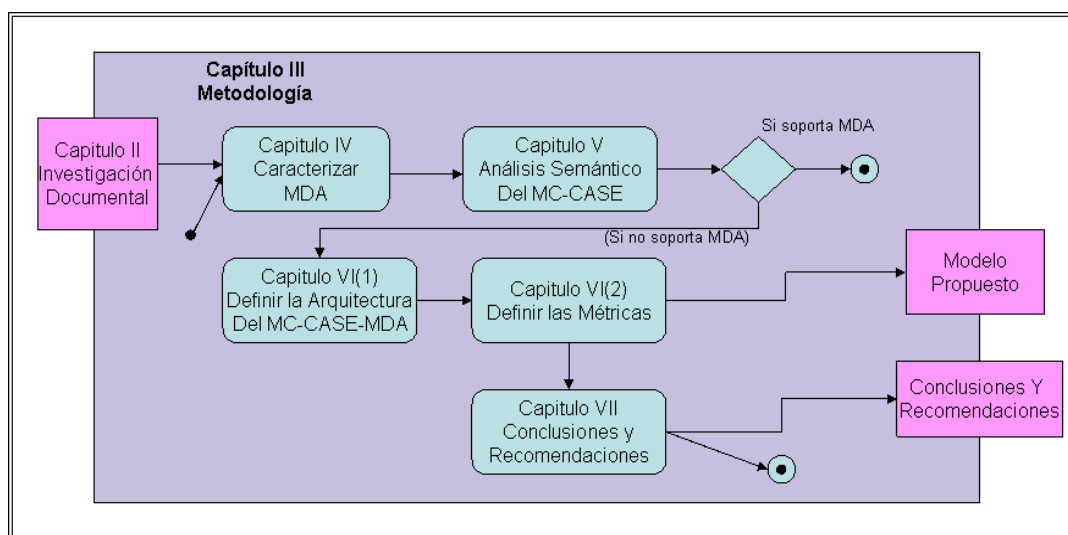


Figura N° 27. Aspectos metodológicos para desarrollar el Trabajo de Investigación
Fuente: Elaboración propia

A continuación se describen cada uno de los pasos que conforman el marco metodológico en el orden que indica la Figura N° 27.

3.2.1. INVESTIGACIÓN DOCUMENTAL Y BIBLIOGRÁFICA

Para Trujillo [Trujillo, 1990], “*representa la explicación científica para comprender la naturaleza del hecho investigado. Es la facultad que tiene la propia investigación de sostenerse ella misma a través de los elementos siguientes: antecedentes, bases teóricas, hechos, variables, hipótesis (de ser necesario), indicadores y definiciones básicas*”. Por su parte Sabino [Sabino, 1978], citado en [Trujillo, 1990], expresa que a través del marco teórico “*se trata de elaborar un conjunto de proposiciones que sirvan de referencia al tema concreto de estudio. Al estar estas proposiciones coherentemente ligadas entre sí irán a constituir el ámbito teórico dentro del cual formularemos nuestras proposiciones específicas, describiremos e interpretaremos los hechos que nos interesan*”.

Tomando como base las afirmaciones anteriores, a través de la revisión bibliográfica y electrónica sobre las herramientas CASE, la Arquitectura Dirigida por Modelos y el Modelo de Calidad para seleccionar herramientas CASE, se conformó el marco teórico referencial, que corresponde al **Capítulo II** del presente Trabajo de Investigación.

3.2.2. CARACTERIZACIÓN DE MDA

Tomando como base la información documental del Capítulo II, se procede a buscar todas las formas que hasta ahora existen de caracterizar a una herramienta que soporte MDA, con el objetivo de establecer un marco referencial para analizar el Modelo de Calidad para Evaluar Herramientas CASE MC-CASE. La caracterización de las herramientas MDA conforman el **Capítulo IV** de esta investigación.

3.2.3. ANÁLISIS SEMÁNTICO DEL MC-CASE

Tomando como insumos primarios el Modelo de Calidad para Evaluar Herramientas CASE MC-CASE y la caracterización realizada en el capítulo IV, se procede a revisar semánticamente el significado de cada elemento presente en el MC-CASE con respecto a los aspectos que se deben considerar presentes en una Herramienta de Desarrollo de Software que soporte MDA, por lo cual aquí se establece el nivel de cobertura que tiene el MC-CASE para soportar la evaluación de una herramienta MDA. Este análisis se encuentra especificado en el **Capítulo V**.

3.2.4. PROPUESTA DEL MODELO

En el **Capítulo VI** se propone el Modelo de Calidad para Evaluar Herramientas CASE con soporte a MDA, la propuesta del Modelo implica la definición de toda la Arquitectura que lo compone, tomando como base el MC-CASE y la definición técnica de cada una de las métricas que deban proponerse para conformar el modelo.

3.2.5. CONCLUSIONES Y RECOMENDACIONES

De acuerdo con Hernández et al. [Hernández et al., 1997], *“en esta parte se derivan conclusiones, se hacen recomendaciones para otras investigaciones, se analizan las implicaciones de la investigación y si se cumplieron o no los objetivos”*. Por otro lado, Trujillo [Trujillo, 1990] indica que las conclusiones *“tienen como objetivo permitir una visión de los resultados obtenidos y generalmente están ligados a los objetivos planteados en la investigación”*.

Tomando en cuenta las afirmaciones de los autores citados anteriormente, se redactaron las conclusiones, las cuales están desarrolladas en el **Capítulo VII**, de tal manera que se facilite la toma de decisiones respecto al tema tratado. Finalmente, se hicieron algunas recomendaciones con respecto a futuras investigaciones relacionadas.

En los capítulos siguientes se describen cada una de estas actividades o pasos, haciendo énfasis en los resultados obtenidos como producto de su aplicación.

CAPÍTULO IV

CARACTERIZACIÓN DE HERRAMIENTAS MDA

Este capítulo tiene por objetivo presentar la caracterización de las herramientas MDA que han hecho los diferentes autores que se han encargado de estudiar este tópico. El análisis de éstas características será el insumo que se usará para estudiar semánticamente el Modelo de Calidad MC – CASE y determinar su competencia para evaluar herramientas MDA. Entre los autores que se estudian en este capítulo están: García y Rodríguez (2004), Herrera, Matteo y Díaz, (2007), Cervantes y Riba (2007), y Bollati, et al, (2008).

4.1 CARACTERIZACIÓN DE ACUERDO CON GARCÍA Y RODRÍGUEZ

Según García y Rodríguez (2004), durante la etapa de realización de su proyecto en la que se trabajó en la identificación de las dimensiones y criterios de evaluación de herramientas MDA, tuvieron conocimiento de la publicación de los resultados de un interesante trabajo (King's, 2003) en el que se había evaluado la herramienta OptimalJ y se habían identificado unos criterios completos y bien definidos que debe cumplir una herramienta con soporte completo para MDA. Dada la calidad de los criterios presentados en dicho estudio, decidieron usar la misma metodología para esta evaluación, adaptando ligeramente algunos de los criterios y añadiendo otros nuevos. A continuación se muestra la tabla N° 36, con las propiedades evaluadas en este estudio.

Código	Propiedad	Descripción
P01	Soporte para MIP	La herramienta permite que se especifique un sistema mediante un modelo independiente de cualquier plataforma (MIP)
P02	Soporte para MEP	La herramienta permite construir modelos del sistema que capturan los aspectos esenciales de una tecnología de implementación determinada (MEP)
P03	Permite varias implementaciones	La herramienta posibilita la generación de varias implementaciones diferentes a partir del mismo MIP, utilizando MEP, marcas u otros mecanismos. También se tendrá en cuenta que puedan añadirse nuevas implementaciones a las disponibles en la herramienta.
P04	Integración de Modelos	Permite integrar diferentes modelos para producir una única aplicación, principalmente mediante la generación de los “puentes” apropiados para comunicar las distintas partes entre si.
P05	Interoperabilidad	La herramienta permite importar y exportar información a otras herramientas.
P06	Acceso a la Definición de las Transformaciones	La herramienta provee de un mecanismo de definición de transformaciones entre modelos y permite al usuario crear nuevas transformaciones o modificar las existentes para

		satisfacer sus requisitos específicos.
P07	Verificador de Modelos	Incluye algún mecanismo para chequear la corrección de los modelos, incluidos MIP y MEP.
P08	Expresividad de los Modelos	La herramienta tiene un lenguaje para representar MIP y MEP lo suficientemente expresivo como para capturar de forma precisa la estructura y funcionalidad en los distintos niveles de abstracción
P09	Uso de Patrones	La herramienta aplica o permite aplicar patrones de diseño en la construcción de MIP, MEP y Código, y pueden definirse otros nuevos o modificar los existentes.
P10	Soporte para la Regeneración de Modelos	La herramienta proporciona soporte para rehacer modelos, por ejemplo, regenerar el MIP a partir de los MEP y viceversa. También debe permitir conservar los cambios efectuados manualmente tanto a nivel de modelo como de código.
P11	Transformaciones Intra-Modelo	Provee soporte para transformar un MEP a otros MEP, o un MIP a otros MIP.
P12	Trazabilidad	Incorpora un mecanismo para seguir el rastro de determinadas transformaciones desde su origen hasta su destino.
P13	Ciclo de Vida	Incluye la mayor parte del ciclo de vida de un desarrollo con MDA, esto es, el análisis, el diseño, la implementación, el ensamblado, el despliegue y el mantenimiento.
P14	Estandarización	La herramienta utiliza los estándares básicos de MDA. Por ejemplo, expresa sus modelos en UML, es capaz de importar y exportar modelos en XMI y de guardarlos en un repositorio MOF.
P15	Control y Refinamiento de las Transformaciones	La aplicación permite dirigir o controlar las transformaciones entre modelos, entre MIP y MEP o entre MEP y Código. Por ejemplo, dispone de parámetros en las transformaciones, permite seleccionar los elementos a ser transformados o establecer condiciones para las transformaciones.
P16	Calidad del Código Generado	La aplicación genera código de calidad, bien documentado, legible y que puede ser adaptado o extendido fácilmente por el desarrollador.
P17	Herramientas de Soporte	Además de las herramientas de transformación, la aplicación incluye otras herramientas para dar soporte completo a MDA: editor de código, editor de modelos, herramientas para prueba y despliegue, etc.

Tabla N° 36. Propiedades de las Herramientas MDA

Fuente: (García y Rodríguez, 2004)

Para evaluar con detalle cada herramienta se puntuará del 0 al 4 cada propiedad de la tabla anterior, según el siguiente criterio.

Puntuación	Descripción
0	Soporte nulo de la propiedad
1	Soporte mínimo de la propiedad
2	Soporte medio de la propiedad
3	Buen soporte de la propiedad
4	Excelente soporte de la propiedad

Tabla N° 37. Puntuación para las Propiedades

Fuente: (García y Rodríguez, 2004)

En la caracterización hecha por García y Rodríguez no se tomó en cuenta la construcción del MIC, y en consecuencia tampoco se referenció la transformación del MIC al MIP. Sin embargo la

manera de colocar la puntuación a las métricas es mucho más precisa que simplemente decir si posee o no la propiedad.

4.2 CARACTERIZACIÓN DE ACUERDO CON HERRERA, MATTEO Y DÍAZ

De acuerdo con (Herrera, Matteo y Díaz, 2007), en los últimos años se han propuesto diversos criterios para la clasificación de herramientas MDA (Czarnecki y Helsen, 2003), (Jezequel, 2005), (Kent, 2002), (Modelbaset.net) y (Tariq y Akhter, 2005). De especial interés son las propuestas de Stuart Kent, Jean-Marc Jezequel, Czarnecki y Helsen y Naveed Ahsan Tariq Naeem Akhter. La elección de estos trabajos se debió a que permiten determinar el nivel de apoyo que puede ofrecer la herramienta MDA a la comunidad de desarrolladores e identificar el enfoque de transformación utilizado por las mismas. En su conjunto, los criterios propuestos por estos autores conforman un marco consistente para la caracterización de herramientas MDA de código abierto. Se trata con esto de establecer sus principales características y posibilitar su comparación de tal manera de facilitar la elección de estas herramientas, en términos del uso que se requiera dar a las mismas

4.2.1. Clasificación de Stuart Kent

La clasificación propuesta por Kent se muestra en la Tabla N° 38. El criterio utilizado por este autor está orientado a establecer las funcionalidades que deberían ofrecer estas herramientas, agrupándolas según el tipo de apoyo que ofrecen a los desarrolladores. Sin embargo, esta clasificación tiene como limitante que no considera los diferentes enfoques de transformación que utilizan estas herramientas.

Código	Característica
1.1	Orientadas a verificar que los modelos están bien formados
1.2	Permiten trabajar con instancias de modelos
1.3	Ofrecen soporte para la transformación entre modelos
1.4	Verificación orientada a modelos
1.5	Dedicadas al control de versiones y el trabajo distribuido
1.6	Apoyan los procesos de desarrollo de software dirigido por modelos

Tabla N° 38. Características según Stuart Kent

Fuente: (Herrera, Matteo y Díaz, 2007)

4.2.2. Clasificación de Jean-Marc Jezequel

Jezequel esboza una caracterización según la técnica de transformación de modelos que aplica la herramienta. El autor expresa en cada categoría la forma cómo las transformaciones de modelos son soportadas, dependiendo de la herramienta y el enfoque de transformación utilizado para ello.

Código	Característica
2.1	Lenguajes de Programación de Propósito General
2.2	Herramienta de Transformación Genérica
2.3	Herramientas CASE de Lenguajes Interpretados
2.4	Dedicadas a la Transformación de Modelos
2.5	De Meta-Meta-modelado

Tabla N° 39. Características según Jean-Marc Jezequel

Fuente: (Herrera, Matteo y Díaz, 2007)

4.2.3. Clasificación de Czarnecki y Helsen

Czarnecki y Helsen establecen una taxonomía para la clasificación de las herramientas MDA según el enfoque de transformación de modelos que utilizan. Establecen dos enfoques de alto nivel: transformación de modelo a código y transformación de modelo a modelo. Como parte de estos enfoques se agrupan otros tal como se muestra en la Tabla N° 40.

Código	Característica
3.1	Enfoque Modelo a Código
3.1.1	Enfoque basado en el patrón "visitor"
3.1.2	Enfoque basado en plantillas
3.2	Enfoque Modelo a Modelo
3.2.1	Enfoque de manipulación directa
3.2.2	Enfoque relacional
3.2.3	Enfoque basado en transformación de grafos
3.2.4	Enfoque centrado en la estructura
3.2.5	Enfoque híbridos

Tabla N° 40 Características según Czarnecki y Helsen

Fuente: (Herrera, Matteo y Díaz, 2007)

4.2.4. Clasificación de Naveed Ahsan Tariq y Naeem Akhter

El aspecto más importante del trabajo de estos autores es la elaboración de un estudio formal de las características esenciales que deben tener las herramientas MDA. Sin embargo, este marco sólo ha sido aplicado a herramientas de código propietario. La Tabla N° 41 muestra un subconjunto de características propuestas por Tariq y Akhter. Este subconjunto se ha seleccionado considerando aquellas características que expresan los aspectos más relevantes de la transformación entre modelos.

Código	Característica
4.1	Soporte de MIC
4.2	Soporte de MIP
4.3	Modelos MOF compatible
4.4	Soporte de Diagrama de Caso de Uso
4.5	Soporte de Diagrama de Clase
4.6	Soporte de Diagrama de Secuencia
4.7	Soporte para AS UML
4.8	Soporte para XMI-XML
4.9	Soporte Query-View-Transformation
4.10	Soporte OCL

4.11	Transformación MIP a MEP a Código
4.12	Transformación Código a MEP a MIP
4.13	Transformación directa MIP a Código
4.14	Transformación directa Código a MIP
4.15	Transformación basada en marcas
4.16	Transformación basada en meta-modelos
4.17	Transformación basada en modelos
4.18	Transformación basada en patrones
4.19	Transformación basada en perfiles UML
4.20	Registro de transformaciones
4.21	Trazabilidad de transformaciones
4.22	Combina 2 o más MIC en 1
4.23	Combina 2 o más MIP en 1

Tabla N° 41. Características según Naveed Ahsan Tariq y Naeem Akhter
Fuente: (Herrera, Matteo y Díaz, 2007)

4.2.5. Otras Características propuestas por Herrera, Matteo y Díaz

Según (Herrera, Matteo y Díaz, 2007), una etapa común en cualquier método de desarrollo de software es la especificación de los requisitos del sistema los cuales son expresados, generalmente, de forma textual. A partir del modelo de especificación se construye el modelo de análisis usando un lenguaje de modelado. Este proceso se hace generalmente de forma manual. No obstante, ya se han propuesto algunas herramientas que pretenden dar soporte automatizado a este proceso (Díaz et al, 2005). Estas herramientas permiten la transformación de un documento o texto (MIC: Modelo Independiente de la Computación) para obtener un Modelo Independiente de Plataforma (MIP) que exprese el resultado del análisis de los requisitos del sistema. La identificación de herramientas que apoyen este tipo de transformaciones y la determinación del alcance de las mismas son entonces necesidades a las que se enfrentan actualmente los desarrolladores. Ninguno de los marcos de evaluación consultados permite establecer si una herramienta MDA satisface estas necesidades. De aquí que estos autores hayan propuesto las características que se enuncian en la Tabla N° 42.

Código	Característica
5.1	Transformación MIC a MIP
5.2	Transformación MIP a MIC
5.3	Transformación MIC a MIP a MEP
5.4	Transformación MEP a MIP a MIC
5.5	Transformación Texto a MOF
5.6	Transformación MOF a Texto
5.7	Transformación de Texto a MIC
5.8	Transformación de MIC a Texto

Tabla N° 42. Características según Herrera, Matteo y Díaz
Fuente: (Herrera, Matteo y Díaz, 2007)

La caracterización hecha por Herrera y sus amigos, amplía de manera importante las características que se le deben evaluar a una herramienta MDA, en esta investigación se tomó en cuenta al MIC, y

todas las transformaciones que se pueden hacer con los modelos que propone la arquitectura del MDA.

4.3 CARACTERIZACIÓN DE ACUERDO CON CERVANTES Y RIBA

Cervantes y Riba hicieron un estudio comparativo de un grupo de herramientas MDA entre las cuales estaban: UVM-QVT, OpenMDX, AndroMDA, EMF y Acceleo, los puntos que se comparan en cada herramienta tienen que ver con aspectos importantes que debe ofrecer una herramienta de desarrollo MDA para considerarse como MDA, además de aspectos como su extensibilidad y facilidad de uso. Las características consideradas para la evaluación se presentan a continuación, éstas se definen en base a la especificación de MDA de la OMG.

a) **MEP a código de cualquier plataforma.** Se revisa que la herramienta sea capaz de generar código para cualquier plataforma, sin estar atada a una plataforma en particular, esto es porque un aspecto importante de MDA es que, cuando surgen nuevas tecnologías, la herramienta debe adaptarse a estas de forma natural.

b) **MIP a MEP.** Es importante que la herramienta cuente con una forma de transformar modelos independientes de la plataforma a modelos en distintas plataformas, de manera que teniendo un solo MIP se pueda regenerar una aplicación una y otra vez.

c) **Editor de metamodelos.** Un aspecto básico de MDA es que los modelos que se generan deben de alguna forma ser validados con respecto a una serie de reglas que se establezcan para cada dominio en particular. La única manera de realizar este proceso de validación es mediante la definición de un metamodelo. Esto puede hacerse vía un editor visual o textual.

d) **Editor de reglas de transformación.** La herramienta debe proveer una manera simple de definir qué elementos de un modelo corresponderán a qué elementos de otro modelo, de manera que se pueda realizar la transformación entre uno y otro. Esto puede hacerse vía un editor visual o textual.

e) **Facilidad adaptación a un dominio en particular.** La herramienta debe permitir de manera simple adaptar metamodelos, reglas de transformación y código generado para cualquier dominio o plataforma existente o no existente aún.

f) **Comercial.** Es importante que la herramienta no tenga un precio excesivo, de manera que pueda ser accesible a la mayoría de los proyectos de desarrollo de software.

g) **Dificultad de uso y aprendizaje.** Un problema con la mayor parte de las herramientas en el mercado es que son difíciles de utilizar y comenzar a sacarles provecho puede llegar a ser una tarea complicada. Esto provoca que las herramientas y los procesos MDA sean rechazados por la mayor parte de los proyectos de desarrollo. En éste trabajo de investigación se realizaron pruebas con las herramientas para determinar de una manera más real el nivel de dificultad que presenta el utilizarlas.

Esta investigación a parte de tomar en cuenta ciertos aspectos internos de la herramienta que la cataloguen como MDA, toma en cuenta aspectos externos referentes a la adaptación a un dominio en particular, que tan comercial es la herramienta y la facilidad de uso y de aprendizaje de la misma. También es importante resaltar que todas las métricas se midieron bajo el valor de Si o No, es decir, si lo tiene o no lo tiene, a diferencia de la última métrica “dificultad de uso y aprendizaje” donde se usó una escala de liker: alta, media, baja.

4.4 CARACTERIZACIÓN DE ACUERDO CON BOLLATI Y OTROS.

Posteriormente en el año 2008, se encuentra una publicación titulada “Una revisión de herramientas MDA”, por Bollati, Vara, Vela y Marcos, en la cual hacen una categorización de características para evaluar una herramienta MDA.

Bollati y sus colaboradores hicieron una revisión de los trabajos relacionados con la evaluación de herramientas MDA, como (García et al, 2004), (King’s, 2003), (Tariq y Akhter, 2005), (Quintero y Anaya, 2007). Luego de su análisis, se basaron puntualmente en el trabajo (García et al, 2004), y en el trabajo de (Quintero y Anaya, 2007). En el de (García et al, 2004), se realiza un estudio comparativo de dos herramientas MDA comerciales (*OptimalJ* y *ArcStyler*). En dicho estudio se han evaluado las herramientas realizando un análisis de propiedades extraídas de la especificación de MDA y desarrollando para ello un caso de estudio, con el fin de detectar cuál de las dos se ajusta mejor a MDA. En el de (Quintero y Anaya, 2007), se presenta un marco de referencia para la evaluación de herramientas MDA y se evalúan diez herramientas de éstas, logrando realizar una taxonomía de las mismas. Para ello, se propone, una clasificación de herramientas, así como los criterios de evaluación. Estos últimos se basan en características de MDA, criterios de calidad, de entorno y generales.

De acuerdo con (Bollati et al, 2008), para la selección de las características que se han estudiado en las diferentes herramientas, se han tenido en cuenta las propuestas en los trabajos citados anteriormente, completándose los mismos con otras características extraídas de la especificación de MDA. Éstas se han agrupado en funcionales, técnicas y de calidad, indicándose para cada una de las características el grado de cumplimiento que debería tener la herramienta de acuerdo a ciertas necesidades: obligatoria (*Obl*) o deseable (*Des*).

4.4.1. Características funcionales

En esta categoría se agrupan todos aquellos requisitos que se correspondan con los aspectos funcionales que debería cumplir una herramienta MDA.

- *Niveles que cubre (Obl)*: qué niveles de MDA cubre (MIC, MIP, MEP).
- *Grado de generación de código (Obl)*: en qué medida permite la transformación de los MEPs a código. Por ello se debe evaluar si las herramientas generan código y si dicho código puede ser implementado de forma directa o si es necesario realizar modificaciones en el mismo. Además se debe verificar si la generación de código se realiza desde el nivel MEP o desde el nivel MIP.
- *Transformaciones (Obl)*: verificar el grado de automatización de las transformaciones.
- *Interacción con el usuario (Des)*: verificar el grado de participación del usuario en el proceso de transformación.
- *Tipo de transformaciones (Obl)*: evaluar, si permite realizar transformaciones verticales y/u horizontales. Se entiende por transformaciones verticales, las que se realizan entre los diferentes niveles de abstracción (MIC, MIP y MEP) y, por horizontales, las que se realizan entre los diferentes modelos de un mismo nivel de abstracción.

4.4.2. Características técnicas

En esta categoría se agrupan todos aquellos requisitos que se correspondan con las características técnicas que debería cumplir una herramienta MDA:

- *Lenguaje de almacenamiento y gestión de modelos (Obl)*: para realizar las transformaciones de los modelos entre los distintos niveles de abstracción es necesario un lenguaje que permita el almacenamiento y la gestión de estos modelos de forma que éstos se puedan intercambiar entre los distintos niveles (MIC, MIP y MEP).

- *Plataformas y tecnologías soportadas (Des)*: verificar qué plataformas de desarrollo soporta la herramienta.
- *Ámbito de aplicación (Des)*: se debe determinar el tipo de desarrollo en el que se centran, en particular para el artículo de Bollati y otros, interesa determinar si las herramientas están orientadas hacia el desarrollo de SI Web o hacia desarrollos orientados a servicios, o ambos.

4.4.3. Características de calidad

En esta categoría se agrupan todos aquellos requisitos que se correspondan con las características de calidad que debería cumplir una herramienta MDA.

- *Uso de Estándares (Des)*: es importante determinar si las herramientas estudiadas hacen uso de estándares como UML, XML y MOF, para la definición de los modelos. Además se tendrá en cuenta si permite definir o personalizar perfiles propios del usuario para UML.
- *Extensibilidad (Obl)*: capacidad que tiene la herramienta de adaptarse a nuevos requisitos.
- *Usabilidad (Des)*: determinar la medida en el que la herramienta puede ser usada por los usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado.
- *Interoperabilidad entre herramientas (Des)*: ver en qué grado la herramienta se puede integrar con otras herramientas con el objetivo de obtener nuevas funcionalidades.

Como puede observarse el estudio hecho por Bollati y sus compañeros categoriza una serie de aspectos que permiten caracterizar a una herramienta MDA, sin embargo estas investigaciones no están bajo el marco completo de un Modelo de Calidad que garantice de alguna forma, la evaluación integral de la herramienta MDA, es por eso, que en el próximo capítulo se presenta el Análisis Semántico del Modelo de Calidad para Evaluar Herramientas CASE MC-CASE propuesto por (Díaz, 2001) con respecto a todas estas investigaciones revisadas en el presente capítulo, esto con el objetivo de determinar el nivel de cobertura del MC-CASE para evaluar herramientas MDA.

CAPÍTULO V

ANÁLISIS SEMÁNTICO DEL MODELO DE CALIDAD PARA EVALUAR HERRAMIENTAS CASE (MC-CASE)

El objetivo central de este capítulo es hacerle un análisis minucioso al Modelo de Calidad para Evaluar Herramientas CASE MC-CASE, en cuanto a cada uno de los aspectos que él contempla, es decir, categorías, características, subcaracterísticas, subcaracterísticas atómicas y las métricas, y determinar si éste modelo soporta la evaluación de herramientas de desarrollo de software MDA; e inclusive estudiar la forma de aplicar el modelo a las herramientas CASE objetos de evaluación. En conclusión poder determinar el nivel de cobertura del MC-CASE para evaluar MDA. El análisis semántico que se desarrollará en éste capítulo se hará a través de tres perspectivas diferentes, la primera esta relacionada con el nivel arquitectural del MC-CASE, la segunda se refiere a la aplicación del modelo como tal y la tercera, relacionará el MC-CASE con las diferentes caracterizaciones de MDA que se desarrollaron en el capítulo IV.

5.1 ANÁLISIS SEMÁNTICO DEL MC-CASE A NIVEL ARQUITECTURAL.

Para realizar este análisis se partirá del esquema gráfico que propone Díaz en (Díaz, 2000), tal como se muestra en la Figura N° 9, y se ira revisando elemento por elemento de arriba hacia abajo, es decir, se realizará un análisis Top – Down al MC-CASE.

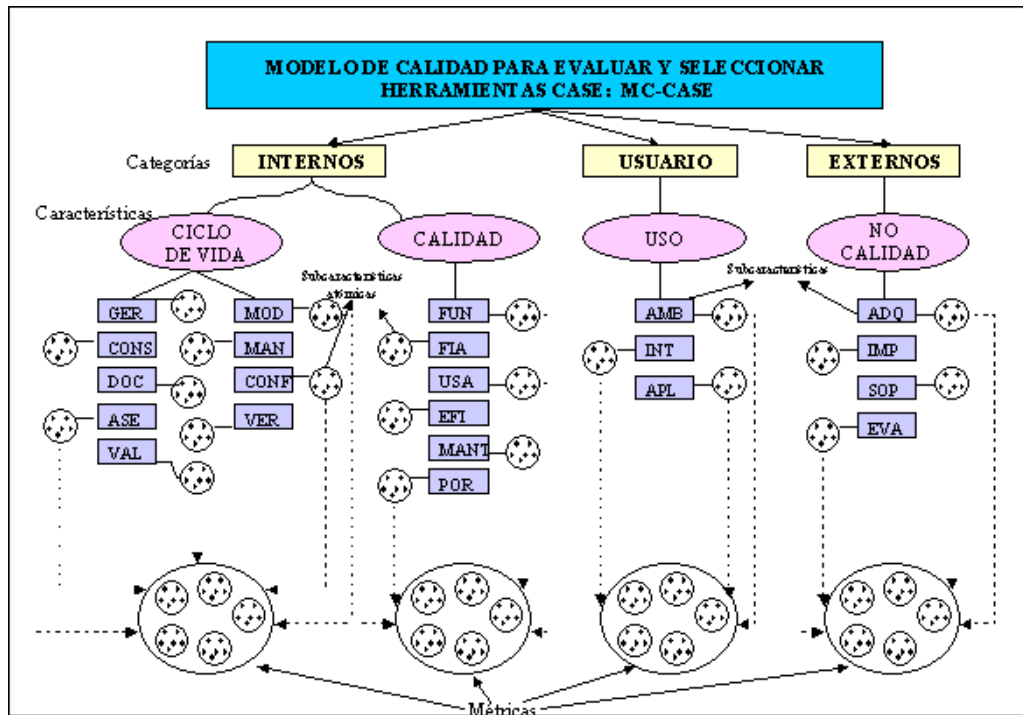


Figura N° 9. Arquitectura del MC-CASE
Fuente: [Díaz, 2000]

5.1.1 ANÁLISIS SEMÁNTICO DEL NIVEL DE CATEGORÍAS.

El Modelo de Calidad MC-CASE indica que se deben evaluar tres grandes **Categorías**: Internos, Usuario y Externos, partiendo de la *categoría Internos* que son todos aquellos aspectos relacionados con la herramienta CASE como tal, la *categoría Externos*, son todos los aspectos que no están relacionados directamente con la herramienta, y la *categoría Usuario* esta relacionado con el cliente quien desea adoptar una herramienta CASE. Éste trabajo de investigación sugiere mantener estas tres categorías por considerarlas suficientes y necesarias para abordar un proceso de evaluación de calidad de una HDS.

5.1.2 ANÁLISIS SEMÁNTICO DEL NIVEL DE CARACTERÍSTICAS.

En el segundo rubro de arriba hacia abajo se encuentran las **Características** donde se ubican las características *Ciclo de Vida* y *Calidad*, dependiendo de la categoría Internos, la característica *Uso* dependiendo de la Categoría Usuario, y la Característica *No Calidad* dependiendo de la Categoría Externos. En cuanto a la Característica *No Calidad*, se considera que no es el nombre ideal o significativo, para los aspectos que representa como son: el proceso de adquisición, la entrega de la

herramienta, su instalación, el soporte técnico que puede ofrecer el proveedor y la certificación del desarrollador y del producto; en ésta investigación se sugiere que en lugar de *No Calidad*, sería un poco más significativo llamarla **Comercialización**.

5.1.3 ANÁLISIS SEMÁNTICO DEL NIVEL DE SUBCARACTERÍSTICAS.

En el tercer nivel de profundidad del MC-CASE, se encuentran las **Subcaracterísticas**, éstas se presentan clasificadas de acuerdo a la Característica a la que pertenecen, tal como se muestra en la Tabla N° 43.

Nivel 1 Categoría	Nivel 2 Características	Nivel 3 Subcaracterísticas
Internos	Ciclo de Vida	Proceso de Gerencia Proceso de Desarrollo-Modelado Proceso de Desarrollo-Construcción Proceso de Mantenimiento Proceso de Documentación Proceso de Gerencia de Configuración Proceso de Aseguramiento de la Calidad Proceso de Verificación Proceso de Validación
	Calidad	Funcionalidad Fiabilidad Usabilidad Eficiencia Mantenibilidad Portabilidad
Usuario	Uso	Ambiente en el cual opera la herramienta CASE Integrabilidad de la herramienta CASE Aspectos de la aplicación de la herramienta
Externos	No Calidad	Proceso de Adquisición Implementación Indicadores de Soporte Evaluación o Certificación

Tabla N° 43. Nivel 1, 2 y 3 del MC-CASE

Fuente: Elaboración Propia

En cuanto a las Subcaracterísticas que pertenecen a la **Característica Ciclo de Vida**, considero que se deben **separar los procesos medulares del Ciclo de vida de los procesos de apoyo**, en tal sentido los procesos medulares serían: Proceso de Desarrollo-Modelado, Proceso de Desarrollo-Construcción, Proceso de Verificación, Proceso de Validación y Proceso de Mantenimiento; y dentro de los procesos de apoyo, se encontrarían: Proceso de Gerencia, Proceso de Documentación, Proceso de Gerencia de la Configuración y Proceso de Aseguramiento de la calidad.

Por otro lado, dentro de estas subcaracterísticas se deja a un lado el Modelado del Negocio y la Ingeniería de Requisitos, los cuales debido a su importancia dentro del proceso de desarrollo de un software deben tomarse en cuenta de manera separada.

Con respecto a las Subcaracterísticas asociadas a la **Característica Calidad**, se considera que están muy bien fundamentadas basándose en el estándar ISO 9126, por lo que en estas subcaracterísticas no se tienen observaciones que conduzcan a una modificación.

En cuanto a las Subcaracterísticas relacionadas con la **Característica Uso**, se tiene que la subcaracterística, *Ambiente en el cual opera la Herramienta CASE*, se refiere a los requisitos no funcionales que requiere la herramienta CASE para poder operar satisfactoriamente, por lo que se sugiere que esta subcaracterística se encuentre dentro de los aspectos relacionados con el cliente como proveedor. Con respecto a la Subcaracterística, *Integrabilidad de la herramienta CASE*, por ser la capacidad que tiene la herramienta de integrarse en el medio de desarrollo en el cual se pondrá en funcionamiento, se considera que esta bien ubicada en esta Característica. Y la Subcaracterística, *Aspectos de la Aplicación de la herramienta CASE*, son todos los requisitos que puede tener el cliente que desea adoptar una herramienta CASE, por lo tanto cada evaluación que se le haga a un cliente distinto, estos son los aspectos que van a variar, el resto permanecería igual, claro está, siempre y cuando sea la misma herramienta y la misma versión.

Ahora bien, las Subcaracterísticas pertenecientes a la **Característica No Calidad**, como son *Proceso de Adquisición, Implementación, Indicadores de Soporte y Evaluación o Certificación*, son todas Subcaracterísticas que dependen del proveedor de la herramienta y su proceso de comercialización.

A continuación se hará el análisis semántico para el nivel cuatro o nivel de Subcaracterísticas Atómicas.

5.1.4 ANÁLISIS SEMÁNTICO DEL NIVEL DE SUBCARACTERÍSTICAS ATÓMICAS.

Por las dimensiones del modelo MC-CASE, el análisis semántico de este nivel se ha dividido en tablas de acuerdo a cada nivel por separado.

Nivel 1 Categoría	Nivel 2 Características	Nivel 3 Subcaracterísticas	Nivel 4 Subcaracterísticas Atómicas
	Ciclo de Vida	Proceso de Gerencia	Estimando costos y planificación Planificación Seguimiento del proyecto Análisis y reportes del estatus del proyecto
		Proceso de	Desarrollo de Diagramas Análisis de diagramas Soporte de especificación de requisitos Soporte de especificación de diseño

Internos	Desarrollo-Modelado	Modelado de especificaciones de la estructura de los procesos Simulación Prototipo Modelado de la interfaz humana
	Proceso de Desarrollo- Construcción	Generación de código Generación de esquemas de bases de datos Generación de pantallas Generación de reportes Compilación Edición dirigida por la sintaxis Depurador
	Proceso de Mantenimiento	Entendimiento del problema Localización Análisis de impacto Ingeniería de reverso de datos Ingeniería de reverso de procesos-procedimientos Reestructuración del código fuente Traducción del código fuente
	Proceso de Documentación	Edición de texto Edición gráfica Edición basada en formas Publicidad Soporte de hipertexto Manejador de variantes Extracción automática de datos y generación de documentos
	Proceso de Gerencia de Configuración	Control de acceso Búsqueda de modificaciones Definición y gerencia de versiones múltiples Estatus de configuración Generación de versiones Capacidad de archivo
	Proceso de Aseguramiento de la Calidad	Gerencia de la calidad de los datos Gerencia de riesgos
	Proceso de Verificación	Análisis de especificación de trazabilidad Análisis de especificación Análisis de código fuente
	Proceso de Validación	Prueba técnica de correctitud Análisis de fallas Análisis de defectos Casos de prueba y espera del resultado de entrada Casos de prueba y generación de resultados esperados Prueba de trazabilidad Instrumentación de código fuente Captura de la entrada y repetición Manejo de pruebas Análisis en tiempo de corrida Análisis de fiabilidad Análisis de alcance de pruebas Gerencia del proceso de pruebas

**Tabla N° 44. Subcaracterísticas atómicas de la categoría Internos, parte I.
Fuente: Elaboración propia**

De acuerdo con la información suministrada en la Tabla N° 44, se puede decir que en primer lugar las subcaracterísticas y subcaracterísticas atómicas resaltadas son las que están relacionadas con el enfoque MDA, es decir, todas las subcaracterísticas atómicas del proceso de desarrollo modelado y

construcción, y esto conduce, a que deberían estar aquí contenidas los aspectos que permitan medir si una herramienta CASE soporta o no a MDA.

Una vez analizadas las subcaracterísticas atómicas que cubren las subcaracterísticas de: proceso de gerencia, proceso de mantenimiento, proceso de documentación, proceso de gerencia de la configuración, proceso de aseguramiento de la calidad, proceso de verificación y proceso de validación, se observa que contemplan todos los aspectos asociados a la Subcaracterística que representan.

Ahora bien, con respecto a las Subcaracterísticas atómicas del **Proceso de Desarrollo-Modelado**, se detalla lo siguiente; en cuanto a las subcaracterísticas atómicas: desarrollo de diagramas, análisis de diagramas, simulación, prototipo y modelado de la interfaz humana, se tienen que revisar minuciosamente las métricas asociadas a ellas para establecer su completitud con respecto a la subcaracterística atómica que representa y el soporte que tienen a MDA, por lo tanto hasta ahora no se puede emitir un juicio. Y con respecto a las subcaracterísticas atómicas: Soporte de especificación de requisitos, soporte de especificación de diseño están relacionadas con la ingeniería de requisitos, paso previo para comenzar a abordar el diseño de la aplicación. Y la subcaracterística atómica Modelado de especificaciones de la estructura de los procesos, pudiera estar relacionada con el modelado del negocio, que es la primera etapa del proceso de desarrollo de software.

Analizando las Subcaracterísticas atómicas pertenecientes al **Proceso de Desarrollo-Construcción** se deduce lo siguiente; hace una buena diferenciación entre código, datos e interfaz, si la herramienta a evaluar no fuera MDA con estas subcaracterísticas pareciera suficiente, pero si la herramienta evaluada es una MDA, hay que considerar revisar exhaustivamente cada una de las métricas de estas subcaracterísticas atómicas para determinar si cubren el enfoque MDA o no.

A continuación se presenta la Tabla N° 45 que contiene las subcaracterísticas atómicas de la característica Calidad.

Nivel 1 Categoría	Nivel 2 Características	Nivel 3 Subcaracterísticas	Nivel 4 Subcaracterísticas Atómicas
	Calidad	Funcionalidad	Seguridad Exactitud Normativas regulatorias Normativas técnicas
			Integridad de los datos Backup automático

Internos	Fiabilidad	Manejo de errores Tolerancia de fallas Recuperabilidad
	Usabilidad	Amigabilidad del usuario Guía del usuario Homogeneidad Adaptabilidad Claridad de control Manejo de errores Concisión Fácil de aprender Calidad de documentación de la herramienta Fácil de instalar
	Eficiencia	Tiempo de respuesta aceptable Requerimientos de almacenamiento de datos Capacidad de memoria aceptable Velocidad de procesamiento aceptable
	Mantenibilidad	Soporte del vendedor Habilidad de la herramienta para hacer cambios en la metodología Actualización Expansibilidad
	Portabilidad	Portabilidad a diferentes plataformas de hardware Compatibilidad con diferentes sistemas operativos Habilidad para mover datos entre versiones de la herramienta Portabilidad con los sistemas Windows

Tabla N° 45. Subcaracterísticas atómicas de la categoría Internos, parte II.
Fuente: Elaboración propia

Con respecto a las **Subcaracterísticas atómicas** referentes a la **Característica Calidad**, no se tienen grandes observaciones puesto que provienen del estándar internacional ISO 9126, estas subcaracterísticas cambiarían radicalmente si se cambia el modelo bajo el cual se establecen las subcaracterísticas. Ahora bien si hay una observación puntual en cuanto a las subcaracterísticas *normativas regulatorias* y *normativas técnicas*, las cuales deben depender de un dominio de desarrollo particular. Y en cuanto a la subcaracterística atómica *portabilidad con los sistemas Windows*, en nuestro país, por una ordenanza del gobierno venezolano la plataforma operativa debe ser Linux, por lo que ésta métrica debe cambiar o adicionarse una nueva.

Ahora se tiene la Tabla N° 46 que contiene las subcaracterísticas atómicas de la categoría Usuario.

Nivel 1 Categoría	Nivel 2 Características	Nivel 3 Subcaracterísticas	Nivel 4 Subcaracterísticas Atómicas
Usuario	Funcionalidad	Ambiente en el cual opera la herramienta CASE	Características de hardware requeridas por la herramienta Ambiente de software requerido por la herramienta Repositorio de software Ambiente físico de la herramienta
		Integrabilidad de la herramienta CASE	Compatibilidad con los elementos del ambiente Integración de data Integración de control Integración de presentación Acceso a metadatos
			Ambiente de hardware y software de los productos de la

	-Uso	Aspectos de la aplicación de la herramienta CASE	herramienta Conformidad de estándares-productos de la herramienta Dominio de la aplicación Tamaño de la aplicación soportada Lenguajes soportados Bases de datos soportadas Metodologías soportadas Internacionalización
--	------	--	---

Tabla N° 46. Subcaracterísticas atómicas de la categoría Usuario.
Fuente: Elaboración propia

Todas las subcaracterísticas atómicas asociadas a la *Subcaracterística Ambiente en el cual opera la herramienta CASE*, es información que suministra el proveedor de la herramienta CASE a ser evaluada, esta información solo cambiará si cambia la versión de la herramienta, de lo contrario una vez determinados los valores de sus métricas, estas permanecerán igual.

Al igual que las subcaracterísticas atómicas anteriores, las que pertenecen a la *Subcaracterística Integrabilidad de la herramienta CASE*, tampoco van a variar a menos que cambie la versión de la herramienta, y están íntimamente relacionadas con la capacidad que tiene la herramienta de desarrollo de software para integrarse con otros productos software.

Ahora bien las Subcaracterísticas atómicas relacionadas con la *Subcaracterística Aspectos de la Aplicación de la herramienta CASE*, son los **aspectos críticos de la evaluación**, porque aquí se evaluará la herramienta CASE con respecto a los requisitos que tenga el usuario, por lo tanto, cada vez que se haga la evaluación a un cliente, deben evaluarse todas estas subcaracterísticas atómicas y determinar si le conviene o no adquirir o adoptar esta herramienta de desarrollo de software.

A continuación se analizarán las subcaracterísticas atómicas pertenecientes a la categoría Externos, observe la Tabla N° 47.

Nivel 1 Categoría	Nivel 2 Características	Nivel 3 Subcaracterísticas	Nivel 4 Subcaracterísticas Atómicas
Externos	No Calidad	Proceso de adquisición	Costo de implementación de la herramienta Políticas de licencias Restricciones de exportación
		Implementación	Efectividad del costo Restricciones de desarrollo-entregas Actividades requeridas por los usuarios de la organización Necesidades de infraestructura
		Indicadores de Soporte	Perfil del vendedor Perfil del producto Disponibilidad de entrenamiento
		Evaluación o Certificación	Evaluación del desarrollador o certificación Certificación del producto

Tabla N° 47. Subcaracterísticas atómicas de la categoría Externos.

Fuente: Elaboración propia

Al analizar cada una de las subcaracterísticas atómicas de esta tabla, se observa que están acordes con las subcaracterísticas que representan, sin embargo, se hace necesario analizar por separado las métricas que permiten medirlas para poder hacer un análisis más profundo.

5.1.5 ANÁLISIS SEMÁNTICO DEL NIVEL DE MÉTRICAS.

El análisis semántico a nivel de métricas es el análisis de más bajo nivel que posee el modelo MC-CASE, y debido a las dimensiones del modelo es el más extenso de todos, por lo que se ha dividido en tablas de acuerdo con la Subcaracterística que representan. Este análisis consistirá en tomar cada métrica y hacer una reflexión sobre su contenido a nivel operacional, en primer lugar su relación con el enfoque MDA, y en segundo lugar determinar si existe alguna condición o restricción que la haga dependiente de “algo”. En el caso de considerarse bien fundamentada y actualmente valida se le colocará la nota de “no hay observación”, de lo contrario se hará el comentario que sea necesario.

Para hacer este análisis también se respetará el orden en el cual se presenta la arquitectura del MC-CASE, en consecuencia se tienen las tablas N° 48 a la N° 69:

CATEGORÍA: INTERNOS. CARACTERÍSTICA: CICLO DE VIDA.

Subcaracterística: Proceso de Gerencia.

Subcaracterística Atómica: Estimando costos y planificación	
Métricas	Observación
Estimación de costos del proyecto	No hay observación
Planificación del proyecto	No hay observación
Cantidad de métodos que posee para la estimación de costos y planificación	No hay observación
Permite ingresar datos referentes a la estimación de costos y planificación	No hay observación
Permite modificar los datos referentes a la estimación de costos y la planificación	No hay observación
Subcaracterística Atómica: Planificación	
Grado de satisfacción de la planificación hecha por la herramienta	No hay observación
Registro de entregables por tareas	No hay observación
Registro de tareas o actividades	No hay observación
Registro de tiempos por tareas	No hay observación
Registro de recursos asignados por tareas	No hay observación
Subcaracterística Atómica: Seguimiento del proyecto	
Permite hacer seguimiento al proyecto	No hay observación
Permite registrar los recursos consumidos	No hay observación
Número de documentos generados para el seguimiento	No hay observación
Líneas de código desarrolladas	No hay observación

Número de casos de pruebas completadas	No hay observación
Número de defectos detectados	No hay observación
Número de defectos corregidos	No hay observación
Grado de satisfacción del seguimiento del proyecto	No hay observación
Subcaracterística Atómica: Análisis y reportes del estatus del proyecto	
Emite reportes del seguimiento del proyecto, en formato definido por el usuario	No hay observación
Emite reportes de la estimación de costos del proyecto	No hay observación
Emite reportes de los costos reales del proyecto	No hay observación
Emite reportes de la planificación del proyecto	No hay observación
Emite reportes de la gerencia del proyecto	No hay observación
Grado de satisfacción de la generación de reportes del seguimiento del proyecto.	No hay observación
Subcaracterística Atómica: Gerencia del proceso	
Definición de entradas para la gerencia del proceso	No hay observación
Definición de recursos	No hay observación
Definición de fechas límites	No hay observación

Tabla N° 48. Análisis Semántico de la Subcaracterística Proceso de Gerencia
Fuente: Elaboración Propia

Estas métricas deben ser evaluadas obligatoriamente si la herramienta de desarrollo de software es una herramienta para hacer gestión y control de proyectos.

Subcaracterística: Proceso de Desarrollo-Modelado.

Subcaracterística Atómica: Desarrollo de diagramas	
Métricas	Observación
Facilidad de diagramación.	No hay observación
Calidad de la edición gráfica.	No hay observación
Grado de facilidad de edición.	No hay observación
Cantidad de tipos de diagramas	Que realice uno, dos, cinco o diez no resulta significativo, a menos que se clasifiquen, como por ejemplo estáticos, y dinámicos, o estructurados u orientados a objeto
Realiza diagramas	Si la respuesta es sí o no, no dice mucho, a menos que se quiera determinar si es una herramienta para diagramar o para codificación.
Grado de satisfacción del desarrollo de diagramas	No hay observación
Grado de completitud de la caja de herramientas para realizar los diagramas	No hay observación
Permite importar objetos entre diagramas	No hay observación
Subcaracterística Atómica: Análisis de diagramas	
Permite variaciones de visión de documentos	No hay observación
Facilidad para detectar inconsistencia	No hay observación
Herramienta de análisis	No hay observación
Verifica consistencia de los diagramas	Solo hace referencia a consistencia entre diagramas, no entre modelos.
Verifica reglas semánticas entre los diagramas	Debe especificarse con respecto a que lenguaje de modelado
Permite el almacenamiento de la información de diseño	No hay observación
Verifica reglas de diagramación	Debe especificarse con respecto a que lenguaje de modelado
Permite la impresión de los diagramas	No hay observación
Emite reportes sobre los diagramas	No hay observación
Subcaracterística Atómica: Soporte de especificación de requerimientos	
Posee un lenguaje formal para representar los requerimientos	Todas estas métricas deben estar ubicadas dentro

de datos	de otra Subcaracterística que sea Ingeniería de Requisitos.
Emite reportes de los requerimientos especificados	
Permite la clasificación de los requerimientos	
Grado de satisfacción de la especificación de requerimientos	
Subcaracterística Atómica: Soporte de especificación de diseño	
Grado de consistencia de la diagramación respecto a la metodología adoptada por la organización.	No hay observación
Grado de ajuste a las metodologías en la etapa de diseño	No hay observación
Facilidad para detectar inconsistencia.	No hay observación
Posee un lenguaje formal para representar el diseño de datos	No hay observación
Emite reportes de los datos del diseño	No hay observación
Clasifica los tipos de datos del diseño	No hay observación
Grado de satisfacción de las especificaciones del diseño	No hay observación
Subcaracterística Atómica: Modelado de especificaciones de estructura de los procesos	
Permite modelar procesos	Es conveniente especificar si son procesos dinámicos del desarrollo del software como son por ejemplo en OO: el diagrama de secuencia, de comunicación, de actividad. O en cuanto a la segunda métrica el diagrama de estructura compuesta. Si no se especifica puede ser que el evaluador lo considere como procesos del modelado del negocio.
Permite modelar la estructura de datos asociada a los procesos	
Grado de satisfacción de la edición de modelos de especificación de procesos	
Subcaracterística Atómica: Simulación	
Permite hacer simulaciones de la utilidad operacional.	No hay observación
Permite hacer simulaciones de la interfaz con el operador	No hay observación
Permite hacer simulaciones del tiempo de respuesta	No hay observación
Permite hacer simulaciones del desempeño	No hay observación
Grado de satisfacción con respecto a las simulaciones hechas por la herramienta.	No hay observación
Subcaracterística Atómica: Prototipo	
Permite la realización de prototipo.	No hay observación
Nivel de generación de prototipo	No hay observación
Grado de satisfacción de la generación de prototipo	No hay observación
Subcaracterística Atómica: Modelado de la interfaz humana	
Permite modelar las interfaces. (menús, ventanas, reportes, etc.)	No hay observación
Grado de satisfacción con respecto al modelado de interfaces.	No hay observación
Facilidad para modelar interfaces	No hay observación
Riqueza de la paleta	No hay observación. Depende totalmente de la valoración que haga el evaluador.

Tabla N° 49. Análisis Semántico de la Subcaracterística Proceso de Desarrollo - Modelado
Fuente: Elaboración Propia

Al analizar todas las métricas de la Tabla N° 49, se observa que no existe dentro de esta estructura un conjunto de métricas que permita determinar si la herramienta CASE es verdaderamente una herramienta MDA o no, es decir, existe una total ausencia de métricas que indique si hay transformación de modelos bajo el esquema MDA, esto quiere decir, MIC, MIP, MEP y Código.

Subcaracterística: Proceso de Desarrollo – Construcción.

Subcaracterística Atómica: Generación de código	
Métricas	Observación
<i>Grado de completitud del código generado</i>	No hay observación
<i>Calidad del código generado</i>	No hay observación

<i>Cantidad de lenguajes en que genera código</i>	No solo la cantidad, sino también cuales.
<i>Generación de códigos</i>	¿A partir de qué genera el código?
<i>Ejecutables para distintas plataformas</i>	No hay observación
Tasa de cobertura del código generado	Dependen de la organización. Es una métrica crítica
Subcaracterística Atómica: Generación de esquemas de bases de datos	
<i>Generación de los esquemas de bases de datos (definir esquemas).</i>	¿A partir de qué genera el esquema?
Grado de satisfacción de los esquemas de bases de datos generados por la herramienta.	No hay observación
<i>Número de manejadores de bases de datos en que genera esquemas</i>	No hay observación
Tasa de cobertura de la generación de esquemas	Dependen de la organización. Es una métrica crítica
Subcaracterística Atómica: Generación de pantallas	
<i>Generador de pantallas.</i>	No hay observación
Bondad de diseño visual.	No hay observación
<i>Genera pantallas</i>	No hay observación
Grado de satisfacción de la generación de pantallas	No hay observación
<i>Facilidad con que se generan las pantallas</i>	No hay observación
Subcaracterística Atómica: Generación de reportes	
Permite generar reportes	No hay observación
Permite definir reportes con formatos del usuario	No hay observación
Calidad de los reportes que genera	No hay observación
Grado de completitud de los reportes	No hay observación
Subcaracterística Atómica: Compilación	
<i>Permite compilar código fuente</i>	No hay observación
<i>Cantidad de lenguajes en que compila código fuente</i>	No hay observación
Grado de satisfacción del proceso de compilación realizado por la herramienta	No hay observación
Tasa de cobertura de los lenguajes en que compila código fuente	Dependen de la organización. Es una métrica crítica
Emite reportes acerca de la compilación	No hay observación
Subcaracterística Atómica: Edición dirigida por la sintaxis	
Provee soporte de sintaxis para la edición del código fuente. (en varios lenguajes de programación).	No hay observación
Grado de satisfacción del soporte de sintaxis de la herramienta	No hay observación
Cantidad de lenguajes en que soporta la sintaxis del código fuente	No hay observación
Subcaracterística Atómica: Depurador	
La herramienta identifica y aísla los errores detectados por el compilador	No hay observación
Permite definir policias en el código a depurar	No hay observación
Permite seleccionar variables para observar los valores que toma	No hay observación
Grado de satisfacción de la identificación de los errores	No hay observación

Tabla N° 50. Análisis Semántico de la Subcaracterística Proceso de Desarrollo - Construcción
Fuente: Elaboración Propia

Las métricas resaltadas con letra en cursiva son las que tienden a indicar soporte a MDA pero solo lo que respecta de MEP a Código, claro está sin ser específico que es a partir de un verdadero MEP, lo cierto es que es “algo” que se transforma a código. También se tienen tres métricas tipo tasas de cobertura que dependen de la organización como cliente que desea adoptar o adquirir la herramienta de desarrollo de software objeto de la evaluación.

Subcaracterística: Proceso de Mantenimiento.

Subcaracterística Atómica: Entendimiento del problema	
Métricas	Observación
Emite reportes de fallas	No hay observación
Emite reportes de defectos	No hay observación
Restringe las operaciones cuando se presenta una falla	No hay observación
Tiene la capacidad de determinar si los problemas han sido resueltos	No hay observación
Tiene la capacidad de aprender de los defectos y fallas encontradas	No hay observación
Subcaracterística Atómica: Localización	
Identifica el problema que requiera modificación.	No hay observación
Identifica la ubicación del problema	No hay observación
Grado de satisfacción de la localización de errores	No hay observación
Subcaracterística Atómica: Análisis de impacto	
Identifica las consecuencias de hacer cambios	No hay observación
Grado de satisfacción de la identificación de las consecuencias de hacer cambios	No hay observación
Difunde los cambios	No hay observación
Grado de satisfacción con los mecanismos de difusión utilizados por la herramienta	No hay observación
Subcaracterística Atómica: Ingeniería de reverso de datos	
Proceso de ingeniería de reverso de datos	Una vez revisada la forma es que es medida esta métrica (ver Apéndice N° 1) y se observa que esta relacionada con la completitud del proceso de ingeniería de reverso de datos, por lo tanto se sugiere que el nombre más significativo para el lector, es: Completitud del proceso de ingeniería de reverso de datos.
Calidad del producto de la Ingeniería de reverso de datos	No hay observación
Tasa de cobertura de los lenguajes de ingeniería de reverso de datos	Dependen de la organización. Es una métrica crítica
Cantidad de lenguajes soportados por la ingeniería de reverso de datos	No hay observación
Permite hacer ingeniería de reverso de datos	No hay observación
Subcaracterística Atómica: Ingeniería de reverso de procesos/procedimientos	
Proceso de ingeniería de reverso de procesos	Una vez revisada la forma es que es medida esta métrica (ver Apéndice N° 1) y se observa que esta relacionada con la completitud del proceso de ingeniería de reverso de procesos, por lo tanto se sugiere que el nombre más significativo para el lector, es: Completitud del proceso de ingeniería de reverso de procesos.
Calidad del producto de la Ingeniería de reverso de procesos	No hay observación
Tasa de cobertura de los lenguajes de ingeniería de reverso de procesos	Dependen de la organización. Es una métrica crítica
Cantidad de lenguajes soportados por la ingeniería de reverso de procesos	No hay observación
Permite hacer ingeniería de reverso de procesos	No hay observación
Subcaracterística Atómica: Reestructuración del código fuente	
Modifica el formato y estructura del código fuente en uno o más lenguajes. (reducir el tamaño del código, reducir el tiempo de ejecución).	No hay observación
Grado de satisfacción de la modificación y reestructuración del código fuente	No hay observación
Número de lenguajes en que reestructura el código fuente	No hay observación
Tasa de cobertura de la reestructuración de código fuente	Dependen de la organización. Es una métrica crítica

Subcaracterística Atómica: Traducción del código fuente	
Permite traducir código fuente de un lenguaje a otro	No hay observación
Número de lenguajes en que traduce el código fuente	No hay observación
Tasa de cobertura de traducción del código fuente	Dependen de la organización. Es una métrica crítica
Grado de satisfacción de la traducción del código fuente	No hay observación

Tabla N° 51. Análisis Semántico de la Subcaracterística Proceso de Mantenimiento
Fuente: Elaboración Propia

De acuerdo con la Tabla N° 51, es importante señalar que cuando se habla de **ingeniería de reverso**, también es necesario hacer ciertas especificaciones para poder determinar si sigue la cadena propuesta por MDA.

Subcaracterística: Proceso de Documentación

Subcaracterística Atómica: Edición de texto	
Métricas	Observación
Facilidad de documentación	No hay observación
Facilidad para editar texto	No hay observación
Grado de satisfacción de la edición de texto	No hay observación
Discrimina instrucciones de variables	No hay observación
Permite la edición automática de texto	No hay observación
Subcaracterística Atómica: Edición gráfica	
Grado de satisfacción de la edición gráfica	No hay observación
Calidad de la edición gráfica	No hay observación
Grado de facilidad de edición gráfica	No hay observación
Grado de completitud de la caja de herramientas para editar datos gráficamente	No hay observación
Subcaracterística Atómica: Edición basada en formas	
Permite la edición a través de formas	No hay observación
Permite la edición a través de formas definidas por el usuario	No hay observación
Grado de satisfacción de la edición basada en formas	No hay observación
Subcaracterística Atómica: Publicidad	
Cuenta con mecanismos tales como e-mail, groupware, para difundir la documentación	No hay observación
Grado de satisfacción de la difusión de la documentación	No hay observación
Subcaracterística Atómica: Soporte de hipertexto	
La herramienta soporta la edición de hipertexto	No hay observación
La herramienta soporta todas las funciones de hipertexto	No hay observación
Grado de satisfacción del soporte de hipertexto.	No hay observación
Subcaracterística Atómica: Manejando variantes	
Facilidad de cambio en producto terminado	No hay observación
Permite identificar las diferencias entre las versiones de un documento	No hay observación
Permite llevar un registro para cada versión	No hay observación
Subcaracterística Atómica: Extracción de datos automática y generación de documentos	
Permite la recuperación de las especificaciones de datos textuales y gráficos.	No hay observación
Genera documentos de las especificaciones de los datos	No hay observación
Grado de satisfacción de la generación de documentos	No hay observación
Número de especificaciones de datos textuales que recupera	No hay observación
Número de especificaciones de datos gráficos que recupera	No hay observación
Permite personalizar los documentos	No hay observación
Grado de completitud de la recuperación de los datos	No hay observación

Tabla N° 52. Análisis Semántico de la Subcaracterística Proceso de Documentación
Fuente: Elaboración Propia

Subcaracterística: Proceso de Gerencia de la Configuración

Subcaracterística Atómica: Control de acceso	
Métricas	Observación
Permite establecer restricciones para acceder los elementos de datos	No hay observación
Permite llevar registros de las modificaciones realizadas a los elementos de datos	No hay observación
Permite diferenciar las autoridades	Resulta más significativo el nombre: Permite diferenciar los niveles de acceso.
Grado de satisfacción del control sobre el acceso que posee la herramienta	No hay observación
Subcaracterística Atómica: Búsqueda de modificaciones	
Permite llevar registros de las modificaciones realizadas al sistema en general	No hay observación
Permite realizar búsquedas de las modificaciones de acuerdo a diferentes criterios	No hay observación
Grado de satisfacción de la búsqueda de modificaciones	No hay observación
Subcaracterística Atómica: Definición y gerencia de versiones múltiples	
Facilidad de integración de versiones	No hay observación
Permite el registro de versiones	No hay observación
Facilidad de búsqueda de las versiones	No hay observación
Subcaracterística Atómica: Estatus de configuración	
Genera reportes donde se resuma la información de las configuraciones gerenciadas en una sesión de trabajo	No hay observación
Grado de completitud de los reportes de configuración generados	No hay observación
Grado de satisfacción de los reportes de configuración generados	No hay observación
Subcaracterística Atómica: Generación de Versiones	
Permite la creación de versiones de un producto a partir de otro	No hay observación
Grado de satisfacción de la generación de versiones	No hay observación
Registra las versiones creadas	No hay observación
Facilidad de búsqueda de versiones	No hay observación
Subcaracterística Atómica: Capacidad de archivo	
Graba automáticamente los productos generados	No hay observación
Facilidad de búsqueda de los productos generados	No hay observación

Tabla N° 53. Análisis Semántico de la Subcaracterística Gerencia de la Configuración
Fuente: Elaboración Propia

Subcaracterística: Proceso de Aseguramiento de la Calidad.

Subcaracterística Atómica: Gerencia de la calidad de los datos	
Métricas	Observación
Proporciona planes de auditoria y revisiones de resultados o productos	No hay observación
Proporciona revisiones de resultados o productos	No hay observación
Subcaracterística Atómica: Gerencia de riesgos	
Permite definir y modificar los riesgos del proyecto	No hay observación
Permite elaborar listas de chequeo de riesgos	No hay observación
Permite clasificar los riesgos	No hay observación
Emite reportes de los riesgos del proyecto	No hay observación
Grado de satisfacción de la gerencia de los riesgos	No hay observación

Tabla N° 54. Análisis Semántico de la Subcaracterística Proceso de Aseguramiento de la Calidad

Fuente: Elaboración Propia

Subcaracterística: Proceso de Verificación.

Subcaracterística Atómica: Análisis de especificación de trazabilidad	
Métricas	Observación
Facilidad para detectar inconsistencia	No hay observación
Verifica consistencia entre los modelos	No hay observación
Grado de satisfacción de la verificación de consistencia realizada	No hay observación
Permite trazar los cambios de un modelo a otro	No hay observación
Facilidad para trazar los cambios de un modelo a otro	No hay observación
Subcaracterística Atómica: Análisis de especificación	
Emite reportes de los modelos de análisis y sus elementos	No hay observación
Emite reportes de los modelos de diseño y sus elementos	No hay observación
Subcaracterística Atómica: Análisis de código fuente	
Permite la definición de métricas de análisis de código fuente	No hay observación
Emite reportes sobre las métricas de análisis de código fuente	No hay observación
Grado de satisfacción de los reportes sobre métricas para analizar el código fuente	No hay observación

Tabla N° 55. Análisis Semántico de la Subcaracterística Proceso de Verificación

Fuente: Elaboración Propia

Subcaracterística: Proceso de Validación.

Subcaracterística Atómica: Prueba de técnicas de correctitud	
Métricas	Observación
1. Permite constatar si se cumplen las restricciones (precondiciones y postcondiciones del software)	No hay observación
Subcaracterística Atómica: Análisis de fallas	
Frecuencia de fallas por tiempo que presenta el software	No hay observación
Rapidez de recuperación después de haberse presentado la falla	No hay observación
Lleva un registro de las fallas presentadas	No hay observación
Lleva un registro de la causa que genera la falla	No hay observación
Emite reportes de las fallas presentadas	No hay observación
Subcaracterística Atómica: Análisis de defectos	
Número de defectos encontrados	No hay observación
Lleva un registro de los defectos encontrados	No hay observación
Lleva un registro de las consecuencias de las fallas	No hay observación
Lleva un registro de las consecuencias de los defectos	No hay observación
Subcaracterística Atómica: Caso de prueba y espera del resultado de entrada	
Facilidad para ingresar casos de prueba	No hay observación
Facilidad para comparar con los resultados esperados	No hay observación
Permite llevar un registro de los casos de prueba y sus resultados	No hay observación
Grado de satisfacción de la comparación con los resultados esperados	No hay observación
Subcaracterística Atómica: Caso de prueba y generación de resultados esperados	
Genera automáticamente casos de prueba	No hay observación
Genera automáticamente los resultados esperados	No hay observación
Grado de satisfacción de la generación de los casos de prueba	No hay observación
Grado de satisfacción de los resultados esperados	No hay observación
Subcaracterística Atómica: Prueba de trazabilidad	
Verifica consistencia entre las pruebas y los datos de prueba	No hay observación
Grado de satisfacción de la verificación de consistencia entre	No hay observación

las pruebas y los datos de prueba	
Subcaracterística Atómica: Instrumentación de código fuente	
Ejecuta el código fuente	No hay observación
Grado de completitud de la ejecución del código fuente	No hay observación
Grado de satisfacción de la ejecución del código fuente	No hay observación
Subcaracterística Atómica: Captura de la entrada y repetición	
Guarda datos que hayan sido editados para casos de prueba	No hay observación
Permite reutilizar los datos de prueba	No hay observación
Subcaracterística Atómica: Manejo de pruebas	
Permite repetir casos de prueba previamente ejecutados	No hay observación
Permite realizar búsqueda de pruebas	No hay observación
Permite hacer planificación de pruebas	No hay observación
Subcaracterística Atómica: Análisis en tiempo de corrida	
Genera reportes del desempeño del programa durante la ejecución del mismo	No hay observación
Permite ingresar indicadores de desempeño de ejecución del programa	No hay observación
Grado de satisfacción de los reportes de desempeño generados	No hay observación
Subcaracterística Atómica: Análisis de fiabilidad	
Permite definir indicadores de confiabilidad	No hay observación
Genera reportes de los indicadores de confiabilidad	No hay observación
Subcaracterística Atómica: Análisis de alcance de pruebas	
Genera reportes del alcance de las pruebas	No hay observación
Grado de satisfacción del alcance de las pruebas	No hay observación
Subcaracterística Atómica: Gerencia del proceso de prueba	
Permite formular un plan para la ejecución de las pruebas	No hay observación
Definición y modificación de prioridades en la ejecución de las pruebas	No hay observación
Definición y modificación de actividades de las pruebas	No hay observación
Definición y modificación de tiempos en los cuales se realizaran las pruebas	No hay observación
Grado de satisfacción de la gerencia del proceso de prueba	No hay observación
Subcaracterística Atómica: Prueba de regresión	
Compara los resultados de diferentes casos de prueba	No hay observación
Genera reportes de las pruebas de regresión	No hay observación
Grado de completitud de los reportes de las pruebas de regresión	No hay observación
Grado de satisfacción con los reportes de las pruebas de regresión	No hay observación
Subcaracterística Atómica: Chequeo automático de resultados	
Chequeo o comparación de las resultados de las pruebas con los resultados esperados	No hay observación
Generación de reportes del chequeo de los resultados de las pruebas	No hay observación
Grado de satisfacción del chequeo de los resultados de las pruebas	No hay observación
Grado de completitud de los reportes del chequeo de los resultados de las pruebas	No hay observación
Subcaracterística Atómica: Análisis de pruebas estadísticas	
Permite definir pruebas estadísticas de los resultados	No hay observación
Genera reportes estadísticos de los resultados de las pruebas	No hay observación
Grado de satisfacción de las pruebas estadísticas de los resultados	No hay observación
Subcaracterística Atómica: Simulación en ambiente de operaciones	
Genera simulaciones en el ambiente real de operación	No hay observación
Permite definir las pautas de la simulación	No hay observación
Grado de satisfacción de las simulaciones en ambiente real	No hay observación
Subcaracterística Atómica: Prueba de integración	
Permite registrar las actividades de integración de software	No hay observación
Registra los tiempos de las actividades de integración	No hay observación

Registra la secuencia de las actividades de integración	No hay observación
Emite reportes de las pruebas de integración	No hay observación
Grado de satisfacción de los reportes de las pruebas de integración.	No hay observación
Grado de completitud de los reportes de las pruebas de integración	No hay observación

Tabla N° 56. Análisis Semántico de la Subcaracterística Proceso de Validación

Fuente: Elaboración Propia

Como puede observarse en las tablas anteriores los procesos de documentación, proceso de gerencia de la configuración, proceso de aseguramiento de la calidad, proceso de verificación y proceso de validación prácticamente no tienen observaciones, lo cual implica que de actualizarse el MC-CASE, estas subcaracterísticas pudieran permanecer igual.

CATEGORÍA: INTERNOS. CARACTERÍSTICA: CALIDAD.

Subcaracterística: Funcionalidad.

Subcaracterística Atómica: Seguridad	
Métricas	Observación
Mecanismo de manejo de niveles de seguridad	No hay observación
Subcaracterística Atómica: Exactitud	
Grado de claridad de los mensajes	No hay observación
Localización rápida de opciones	No hay observación
Grado de versatilidad de la navegación	No hay observación
Subcaracterística Atómica: Normativas regulatorias	
Genera productos acordes con las normas legales del país	El valor de esta variable estará sujeto al dominio en el que se este trabajando
Grado de completitud en que los productos responden a las normas legales del país	El valor de esta variable estará sujeto al dominio en el que se este trabajando
Subcaracterística Atómica: Normativas técnicas	
Genera productos acordes con las normas técnicas del país	El valor de esta variable estará sujeto con el dominio en el que se este trabajando

Tabla N° 57. Análisis Semántico de la Subcaracterística Funcionalidad

Fuente: Elaboración Propia

Subcaracterística: Fiabilidad.

Subcaracterística Atómica: Integridad de los datos	
Métricas	Observación
Permite accesos no autorizados	No hay observación
Permite jerarquizar los accesos	No hay observación
Tiene mecanismos de "backup" o replicación de los datos	No hay observación
Permite recuperar los datos almacenados	No hay observación
Subcaracterística Atómica: Backup automático	
Realiza copias de seguridad automáticamente.	No hay observación
Grado de satisfacción de las copias de seguridad realizadas por la herramienta	No hay observación
Subcaracterística Atómica: Manejo de errores	
Capacidad de detectar comportamientos anormales	No hay observación
Guarda los trabajos en el momento de alguna interrupción	No hay observación
Los mensajes de error que genera están acordes a los errores presentados	No hay observación
Subcaracterística Atómica: Tolerancia de fallas	

Conservación del nivel de desempeño al presentarse fallas (hardware, software, red, etc.)	No hay observación
Número de fallas presentadas durante una jornada de trabajo	No hay observación
Frecuencia con la que se presentan las fallas	No hay observación
Grado de satisfacción de la tolerancia a fallas de la h. CASE	No hay observación
Subcaracterística Atómica: Recuperabilidad	
Recupera su nivel de desempeño después de presentada una falla	No hay observación
Recupera los datos afectados durante la falla	No hay observación
Tiempo que requiere para recuperar el nivel de desempeño	No hay observación

Tabla N° 58. Análisis Semántico de la Subcaracterística Fiabilidad
Fuente: Elaboración Propia

Subcaracterística: Usabilidad.

Subcaracterística Atómica: Amigabilidad del usuario	
Métricas	Observación
La herramienta es intuitiva	No hay observación
Facilidad de uso	No hay observación
Subcaracterística Atómica: Guía del usuario	
Presencia de metáforas	No hay observación
Rapidez del sistema de ayuda	No hay observación
Subcaracterística Atómica: Homogeneidad	
Facilidad de detectar inconsistencia	No hay observación. Inconsistencia en cuanto al uso
Subcaracterística Atómica: Adaptabilidad	
Capacidad de parametrización	No hay observación
La interfaz de la herramienta se adapta a diferentes hábitos y modos culturales	No hay observación
Permite la parametrización o personalización	No hay observación
Subcaracterística Atómica: Claridad de control	
Grado de versatilidad de la navegación.	No hay observación
Localización rápida de opciones	No hay observación
Sensibilidad de contexto	No hay observación
Subcaracterística Atómica: Manejo de errores	
Grado de claridad de los mensajes.	No hay observación
Permite la corrección de los errores evitando datos incorrectos y cambios en los procesos	No hay observación
Subcaracterística Atómica: Concisión	
Presencia de metáforas.	No hay observación
Sensibilidad de contexto	No hay observación
Subcaracterística Atómica: Fácil de aprender	
Horas de entrenamiento requeridas para lograr la experticia necesaria para el uso completo de la herramienta, en una persona que desconozca la misma.	No hay observación
La herramienta es intuitiva	No hay observación
Subcaracterística Atómica: Calidad de documentación de la herramienta	
Grado de completitud del material de apoyo según la funcionalidad de la herramienta.	No hay observación
Calidad de la clasificación de los temas.	No hay observación
Profundidad en el tratamiento de los temas.	No hay observación
Consistencia del material.	No hay observación
Tasa de cobertura del material de apoyo en cuanto a su presentación.	No hay observación
Tasa de cobertura del material de apoyo en cuanto a los usuarios a los que esta dirigido.	No hay observación
Variedad del material de apoyo en cuanto a los usuarios a los que esta dirigido.	No hay observación

Variedad del material de apoyo de acuerdo a su presentación.	No hay observación
Subcaracterística Atómica: Fácil de instalar	
Facilidad para instalar la herramienta	No hay observación
Tiempo real de instalación	No hay observación
Tasa del tiempo promedio de instalación	No hay observación
Se requiere al proveedor para la instalación	No hay observación
Se requiere de material de apoyo para la instalación	No hay observación

Tabla N° 59. Análisis Semántico de la Subcaracterística Usabilidad
Fuente: Elaboración Propia

Subcaracterística: Eficiencia.

Subcaracterística Atómica: Tiempo de respuesta aceptable	
Métricas	Observación
Grado de satisfacción del tiempo de respuesta.	No hay observación
Subcaracterística Atómica: Requerimientos de almacenamiento de datos	
Cantidad de memoria secundaria necesaria para almacenar a la herramienta y a las bases de datos requeridas y generadas por ella	No hay observación
La plataforma de hardware de la organización cuenta con la cantidad de memoria secundaria necesaria	No hay observación
Subcaracterística Atómica: Capacidad de memoria aceptable	
Cantidad de memoria RAM requerida para cargar y operar la herramienta satisfactoriamente	No hay observación
Grado de desempeño de la herramienta con respecto a la cantidad de memoria RAM disponible en el ambiente de trabajo de la organización	No hay observación
Subcaracterística Atómica: Velocidad de procesamiento aceptable	
Tipo y velocidad del procesador requerido para operar con la herramienta de manera satisfactoria	No hay observación
Aceptabilidad del desempeño de la herramienta durante la ejecución del proyecto	No hay observación
Grado de rapidez con que se accesan las opciones dentro de la herramienta en ejecución	No hay observación

Tabla N° 60. Análisis Semántico de la Subcaracterística Eficiencia
Fuente: Elaboración Propia

Subcaracterística: Mantenibilidad.

Subcaracterística Atómica: Soporte del vendedor	
Métricas	Observación
Soporte técnico permanente.	No hay observación
Soportes adicionales (web, mail, etc.).	No hay observación
Grado de satisfacción del tiempo promedio de respuesta ante una solicitud de respuesta	No hay observación
Tasa de cobertura del horario de atención del soporte técnico.	Dependen de la organización. Es una métrica crítica. Porque relaciona el horario de atención que proporciona la empresa que venda la herramienta con el horario de atención que requiere el cliente que desea adquirir la herramienta.
Horas semanales de soporte técnico.	No hay observación
Subcaracterística Atómica: Habilidad de la herramienta para hacer cambios en la metodología	
El vendedor puede modificar la herramienta para mantener el soporte a las metodologías	No hay observación
Subcaracterística Atómica: Actualización	

Grado de mejoramiento de la herramientas	No hay observación
Subcaracterística Atómica: Expansibilidad	
El desarrollador de la H. CASE genera "upgrade" de la herramienta	No hay observación
Grado de facilidad para obtener los "upgrade" de la herramienta	No hay observación
Frecuencia con la cual el proveedor lanza al mercado los "upgrade"	No hay observación

Tabla N° 61. Análisis Semántico de la Subcaracterística Mantenibilidad
Fuente: Elaboración Propia

Subcaracterística: Portabilidad.

Subcaracterística Atómica: Portabilidad a diferentes plataformas de hardware	
Métricas	Observación
Portabilidad a otras plataformas de hardware	No hay observación
Los productos de la h. CASE se pueden ejecutar para distintas plataformas de hardware	No hay observación
Subcaracterística Atómica: Compatibilidad con diferentes sistemas operativos	
La herramienta puede ejecutarse sobre varias versiones de un mismo sistema operativo	No hay observación
Se ejecuta sobre varios sistemas operativos	No hay observación
Cantidad de sistemas operativos sobre los cuales opera la herramienta	No hay observación
Tasa de cobertura de los sistemas operativos requeridos por la organización	Dependen de la organización. Es una métrica crítica. Porque relaciona los sistemas operativos en que opera la CASE con los sistemas operativos que requiera el cliente en que opere la herramienta.
Subcaracterística Atómica: Habilidad para mover datos entre versiones de la herramienta	
La herramienta puede usar datos provenientes de otra versión de la herramienta	No hay observación
Permite guardar como otra versión de la h. CASE	No hay observación
Subcaracterística Atómica: Portabilidad con sistemas Windows	
La herramienta se ejecuta bajo plataformas Windows	Esta métrica es necesario que se actualice, se tiene que ampliar para otras plataformas operativas
Versiones de Windows que soportan a la h. CASE	Se tiene que ampliar por ejemplo para Linux.

Tabla N° 62. Análisis Semántico de la Subcaracterística Portabilidad
Fuente: Elaboración Propia

De las tablas N° 57 a la 62, se puede concluir que las subcaracterísticas fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad prácticamente no tienen cambios radicales en su estructura, salvo las métricas del tipo tasas de cobertura que deben actualizarse de acuerdo al cliente y las métricas que de manera natural deben adaptarse a las plataformas operativas de hoy en día, como son las relacionadas con portabilidad con sistemas Windows. Y en el caso de la subcaracterística funcionalidad, donde las métricas relacionadas con las normativas regulatorias y técnicas deben actualizarse con respecto al dominio para el cual se requiera la herramienta CASE.

CATEGORÍA: USUARIO. CARACTERÍSTICA: USO.

Subcaracterística: Ambiente en el cual opera la Herramienta CASE.

Subcaracterística Atómica: Características de hardware requeridas por la herramienta	
Métricas	Observación
La organización o el usuario posee la plataforma de hardware necesaria para soportar a la h. CASE	No hay observación
Subcaracterística Atómica: Ambiente de software requerido por la herramienta	
Cantidad de plataformas de software que soportan a la herramienta.	No hay observación
Tasa de la capacidad de la plataforma de la organización para soportar la herramienta	Dependen de la organización. Es una métrica crítica. Porque relaciona las plataformas de software necesarias para hacer funcionar la herramienta contra las plataformas que posee la organización como usuaria.
Cantidad de plataformas de software que posee la organización	No hay observación
Subcaracterística Atómica: Repositorio de software	
Repositorio común	No hay observación
Subcaracterística Atómica: Ambiente físico de la herramienta	
Requiere actualizar un repositorio en múltiples sitios	No hay observación
Permite distribuir geográficamente los nodos de una red	No hay observación
Número de usuarios concurrentes	No hay observación
Grado de satisfacción en el manejo de las concurrencias	No hay observación
Tasa de cobertura de la concurrencia requerida	Dependen de la organización. Es una métrica crítica. Porque relaciona la concurrencia que puede tener la herramienta contra la concurrencia esperada por la organización como usuaria.

Tabla N° 63. Análisis Semántico de la Subcaracterística Ambiente en el cual opera la herramienta CASE

Fuente: Elaboración Propia

Subcaracterística: Integrabilidad de la herramienta CASE.

Subcaracterística Atómica: Compatibilidad con los elementos del ambiente	
Métricas	Observación
Grado de portabilidad.	No hay observación
Portabilidad a otras plataformas.	No hay observación
Subcaracterística Atómica: Integración de data	
Calidad de la integración.	No hay observación
Es una herramienta integrada.	No hay observación
Integración de data.	No hay observación
Subcaracterística Atómica: Integración de control	
Integración de actividades.	No hay observación
Subcaracterística Atómica: Integración de presentación	
Integración de interface.	No hay observación
Subcaracterística Atómica: Acceso a metadata	
Repositorio común.	No hay observación

Tabla N° 64. Análisis Semántico de la Subcaracterística Integrabilidad de la herramienta CASE

Fuente: Elaboración Propia

Subcaracterística: Aspectos de la aplicación de la herramienta CASE.

Subcaracterística Atómica: Ambiente de hardware y software de los productos de la herramienta	
Métricas	Observación
Los productos de la herramienta requieren de un ambiente de hardware y software especial	Dependen de la organización. Es una métrica crítica
La organización posee la plataforma de hardware que requiere la h. CASE	Dependen de la organización. Es una métrica crítica

La organización posee la plataforma de software que requiere la h. CASE	Dependen de la organización. Es una métrica crítica
Grado de satisfacción del desempeño de los productos de la herramienta en el ambiente de software	No hay observación
Grado de satisfacción del desempeño de los productos de la herramienta en el ambiente de hardware	No hay observación
Subcaracterística Atómica: Conformidad de estándares –Productos de la herramienta-	
La herramienta genera productos que están acordes con los estándares establecidos por la organización	Dependen de la organización. Es una métrica crítica
La herramienta permite definir los estándares que deben llevar los productos de la organización	Dependen de la organización. Es una métrica crítica
Total de etapas soportadas	No hay observación
Tasa de cobertura de las etapas requeridas	Dependen de la organización. Es una métrica crítica
Grado de consistencia de la diagramación respecto a la metodología adoptada por la organización	No hay observación
Subcaracterística Atómica: Dominio de la aplicación	
Soporta el tipo de aplicación que el usuario desee desarrollar	Dependen de la organización. Es una métrica crítica
Grado de satisfacción del soporte de la herramienta con respecto a la aplicación requerida por la organización	No hay observación
Subcaracterística Atómica: Tamaño de la aplicación soportada	
Soporta el tamaño de la aplicación que el usuario desea desarrollar	Dependen de la organización. Es una métrica crítica
Subcaracterística Atómica: Lenguajes soportados	
Cantidad de lenguajes en que genera código	No hay observación
Tasa de cobertura del código generado	Dependen de la organización. Es una métrica crítica
Cantidad de lenguajes soportados por la ingeniería de reverso	No hay observación
Tasa de cobertura de los lenguajes de ingeniería de reverso	Dependen de la organización. Es una métrica crítica
Subcaracterística Atómica: Bases de datos soportadas	
Tasa de la manipulación de los manejadores de bases de datos requeridos por la organización.	Dependen de la organización. Es una métrica crítica
Total de manejadores de bases de datos que maneja.	No hay observación
Subcaracterística Atómica: Metodologías soportadas	
Tasa de cobertura de las metodologías requeridas.	Dependen de la organización. Es una métrica crítica
Total de metodologías soportadas.	No hay observación
Subcaracterística Atómica: Internacionalización	
Varia la filosofía de trabajo de acuerdo con la cultura o país	No hay observación
Grado de satisfacción de la variación de filosofía de trabajo	No hay observación
Cambia el idioma de acuerdo a la cultura o país	No hay observación

Tabla N° 65. Análisis Semántico de la Subcaracterística Aspectos de la Aplicación de la Herramienta CASE

Fuente: Elaboración Propia

Las métricas de la Tabla N° 65 están relacionadas con el uso que se le dará a la herramienta CASE dentro de la organización que la está adquiriendo. Por lo que el usuario pudiera querer saber si cubre MDA o no y en que proporción cubre: del MIC a MIP, o del MIP a MEP, o de MEP a Código, o de MIC a MIP a MEP a Código, o de MIC a MIP a MEP, o de MIP a MEP a Código y viceversa. Por otro lado, cuando las métricas hablan de **dominio** pudiera parecer algún tipo de soporte al **MIP**, pero aun así se tiene que ser muy claro en la definición de la métrica. Ahora bien, con respecto a la subcaracterística Integridad de la herramienta CASE, no se tienen

observaciones radicales, solo se observa que no existe presencia de métricas relacionadas con la interoperabilidad de la herramienta CASE con otros productos software o con otras CASE.

En cuanto a la subcaracterística Ambiente en el cual opera la herramienta CASE, estas métricas deben evaluarse de acuerdo al cliente y al uso que le quiera dar a la herramienta, porque la mayoría de sus métricas determina la compatibilidad de los requisitos de hardware y software con exige la herramienta contra o versus la plataforma que posee la organización donde será implantada la CASE.

CATEGORÍA: EXTERNOS. CARACTERÍSTICA: NO CALIDAD.

Subcaracterística: Proceso de Adquisición.

Subcaracterística Atómica: Costo de implementación de la herramienta	
Métricas	Observación
Costos de adquisición.	El valor de esta métrica está sujeta a la fecha en la cual se realice la evaluación, es decir, cada vez que se evalúe la herramienta se tiene que verificar con el proveedor el valor real de la métrica.
Costos de entrenamiento.	El valor de esta métrica está sujeta a la fecha en la cual se realice la evaluación, es decir, cada vez que se evalúe la herramienta se tiene que verificar con el proveedor el valor real de la métrica.
Tasa de cobertura del costo de adquisición de la herramienta.	Dependen de la organización. Es una métrica crítica
Costo de hardware	El valor de esta métrica está sujeta a la fecha en la cual se realice la evaluación, es decir, cada vez que se evalúe la herramienta se tiene que verificar con el proveedor el valor real de la métrica.
Costo de software	
Tasa de cobertura de costo de hardware	Dependen de la organización. Es una métrica crítica
Tasa de cobertura de costo de software	Dependen de la organización. Es una métrica crítica
Costos de instalación	El valor de estas métricas está sujeta a la fecha en la cual se realice la evaluación, es decir, cada vez que se evalúe la herramienta se tiene que verificar con el proveedor el valor real de la métrica.
Costo de mantenimiento inicial	
Tasa de cobertura de los costos de instalación	Dependen de la organización. Es una métrica crítica
Tasa de cobertura de los costos de mantenimiento inicial	Dependen de la organización. Es una métrica crítica
Subcaracterística Atómica: Políticas de licencias	
El proveedor establece políticas de licencias con respecto a la herramienta.	El valor de esta métrica está sujeta a la fecha en la cual se realice la evaluación, es decir, cada vez que se evalúe la herramienta se tiene que verificar con el proveedor el valor real de la métrica.
El proveedor otorga el derecho a copias	
El proveedor restringe algún uso de la herramienta.	
La herramienta tiene garantía	
Subcaracterística Atómica: Restricciones de exportación	
El proveedor restringe la exportación de la herramienta	El valor de esta métrica está sujeta a la fecha en la cual se realice la evaluación, es decir, cada vez que se evalúe la herramienta se tiene que verificar con el proveedor el valor real de la métrica.

Tabla N° 66. Análisis Semántico de la Subcaracterística Proceso de Adquisición

Fuente: Elaboración Propia

Subcaracterística: Implementación.

Subcaracterística Atómica: Efectividad del costo	
Métricas	Observación
Es la herramienta rentable para la organización. (análisis costo – beneficio)	El valor de esta métrica dependerá del cliente y del uso que le desee dar a la herramienta.
Subcaracterística Atómica: Restricciones de desarrollo/entregas	
La herramienta es intuitiva.	No hay observación
Tiene tutorial	No hay observación
Grado de satisfacción del servicio del tutorial	No hay observación
La herramienta permite ir más allá del producto requerido por la organización	El valor de esta métrica dependerá del cliente y del uso que le desee dar a la herramienta.
Subcaracterística Atómica: Actividades requeridas por los usuarios de la organización	
La herramienta soporta actividades especiales con respecto al ambiente del usuario	El valor de esta métrica dependerá del cliente y del uso que le desee dar a la herramienta.
Subcaracterística Atómica: Necesidades de infraestructura	
Requiere de condiciones de infraestructura especiales para su operación. (Espacio del piso, condiciones eléctricas, condiciones de temperatura, etc.)	Si la versión de la herramienta cambia, se tiene que verificar este valor nuevamente.

Tabla N° 67. Análisis Semántico de la Subcaracterística Implementación
Fuente: Elaboración Propia

Subcaracterística: Indicadores de Soporte.

Subcaracterística Atómica: Perfil del vendedor	
Métricas	Observación
Grado de bienestar financiero de la empresa desarrolladora.	El valor de esta métrica está sujeta a la fecha en la cual se realice la evaluación, es decir, cada vez que se evalúe la herramienta se tiene que verificar con el proveedor el valor real de la métrica.
Tiempo de la empresa desarrolladora en el mercado con esa herramienta.	
Participación en el mercado con esta herramienta.	
Subcaracterística Atómica: Perfil del producto	
Publicaciones expertas positivas.	El valor de esta métrica está sujeta a la fecha en la cual se realice la evaluación, es decir, cada vez que se evalúe la herramienta se tiene que verificar con el proveedor el valor real de la métrica.
Desempeño de la herramienta en otras empresas.	
Experiencia positiva con otras empresas	
Subcaracterística Atómica: Disponibilidad de entrenamiento	
Cantidad de cursos básicos al año.	El valor de esta métrica está sujeta a la fecha en la cual se realice la evaluación, es decir, cada vez que se evalúe la herramienta se tiene que verificar con el proveedor el valor real de la métrica.
Cantidad de cursos avanzados al año.	
Cantidad de tipos de cursos.	
Cobertura de cursos básicos al año.	
Cobertura de cursos avanzados al año.	

Tabla N° 68. Análisis Semántico de la Subcaracterística Indicadores de Soporte
Fuente: Elaboración Propia

Subcaracterística: Evaluación o Certificación.

Subcaracterística Atómica: Evaluación del desarrollador o Certificación	
Métricas	Observación
Posee certificaciones	El valor de esta métrica está sujeta a la fecha en la cual se realice la evaluación, es decir, cada vez que se evalúe la herramienta se tiene que verificar
La organización que evalúa o certifica es reconocida	

	con el proveedor el valor real de la métrica.
Subcaracterística Atómica: Certificación del producto	
La herramienta posee certificaciones	El valor de esta métrica está sujeta a la fecha en la cual se realice la evaluación, es decir, cada vez que se evalúe la herramienta se tiene que verificar con el proveedor el valor real de la métrica.
La organización que certificó es reconocida	

Tabla N° 69. Análisis Semántico de la Subcaracterística Evaluación o Certificación

Fuente: Elaboración Propia

Observe que la mayoría de las métricas que se encuentran en las tablas de la 66 a la 69, dependen del factor tiempo, es decir, cada evaluación que se le haga a la misma herramienta, pero en tiempos diferentes, debe implicar la evaluación de las métricas de la categoría Externos. También hay presencia de métricas críticas que dependen exclusivamente de las necesidades del cliente y métricas tipo tasa de cobertura.

5.2. ANÁLISIS SEMÁNTICO DE LA APLICACIÓN DEL MODELO MC-CASE.

Este Modelo de Calidad, cuenta con una serie de métricas distribuidas en toda la arquitectura, con una concentración considerable en la Subcaracterística *Aspectos de la Aplicación de la Herramienta CASE*; estas métricas tienen una particularidad muy importante, y es que determinan si la herramienta le conviene a la organización o no, basándose solamente en los requisitos que tenga el cliente, debido a que relacionan lo que desea la organización contra lo que la herramienta soporta o permite, por lo que a estas métricas se les puede atribuir una característica de criticidad, dentro del Modelo MC-CASE.

$$\text{Métrica Crítica} = \frac{\text{Proporción del requisito soportado}}{\text{Requisito de la organización}}$$

Por otro lado, también se considera una métrica crítica a aquella métrica que se construye de un requisito directo del cliente.

Existen otro grupo de métricas también muy importantes para el modelo que son los *grados de Conformidad o satisfacción* con respecto a un aspecto en particular. Estos grados de satisfacción representan unos coeficientes relativos con la percepción que tiene el evaluador de la herramienta de desarrollo de software. Por ejemplo, se tienen dos herramientas A y B, y las dos cubren el proceso de MDA, pero, con cuál el evaluador se siente más cómodo de acuerdo a su criterio, entonces allí entra en juego este coeficiente.

Y junto a estos coeficientes surge otro factor muy importante que es el evaluador como tal, debe recordarse que el evaluador será un sujeto preferiblemente con experiencia en todo lo relacionado con el proceso de desarrollo de software y que tenga cierta experticia con el uso de la herramienta CASE, por lo que se está incorporando en el modelo un grado de subjetividad, que se denominará Error del evaluador o error de experiencia.

Estas consideraciones no las tiene tomadas en cuenta el Modelo como tal, solamente están allí presentes, pero el modelo no dice que hacer con ellas. Tampoco presenta un algoritmo o un método a seguir de manera sistémica para ejecutar el Modelo. Por otro lado, el Modelo MC-CASE es solo un modelo teórico, y que para esta investigación le falta un soporte matemático que lo puede acompañar para que los resultados de la evaluación sean más objetivos que subjetivos. Todas estas consideraciones a parte de que se evidenció en este análisis que el Modelo MC-CASE no soporta evaluaciones bajo enfoques MDA y sumado a la importancia que tiene este Modelo, de ser hasta ahora un esquema suficientemente sólido y aceptado como modelo de calidad para evaluar herramientas CASE, lo cual hace un aporte importante a la Ingeniería de Software, se hace necesario su actualización o mantenimiento; es por lo que en el próximo capítulo se propondrá un Modelo de Calidad para evaluar herramientas de desarrollo de software que incluyan soporte a MDA. Pero antes de abordar el próximo capítulo, resulta interesante analizar comparativamente el MC-CASE con respecto a las caracterizaciones de MDA que se hicieron en el capítulo IV, la siguiente sección versará sobre este aspecto.

5.3 ANÁLISIS SEMÁNTICO – COMPARATIVO DEL MC-CASE CON LAS CARACTERIZACIONES DE MDA.

Tomando como base las caracterizaciones de herramientas de desarrollo de software MDA que se hizo en el capítulo IV del presente trabajo de investigación, se elaborará una comparación de cada una de ellas con respecto al modelo de Calidad MC-CASE con el objetivo de determinar hasta que punto el MC-CASE cubre los atributos tomados en cuenta por estos autores.

A continuación se presenta la Tabla N° 70, donde se establece el nivel de cobertura del Modelo MC-CASE con respecto a la caracterización hecha por García y Rodríguez en su investigación, y

sobre la base del significado que ellos le dieron a cada atributo, se ubicaron dentro del MC-CASE todas las métricas que cubrieran todo lo relativo al mismo.

Características según García y Rodríguez	Aspectos del MC-CASE que cubren la característica
Soporte para MIP	No posee
Soporte para MEP	No posee
Permite varias implementaciones	No posee
Integración de Modelos	Solo en un 10% con las métricas: calidad de la integración, integración de data, integración de actividades e integración de interfaces.
Interoperabilidad	La herramienta puede usar datos provenientes de otra versión de la herramienta
Acceso a la Definición de las Transformaciones	No posee
Verificador de Modelos	Facilidad para detectar inconsistencia Verifica consistencia entre los modelos Grado de satisfacción de la verificación de consistencia realizada Permite trazar los cambios de un modelo a otro Facilidad para trazar los cambios de un modelo a otro Emite reportes de los modelos de análisis y sus elementos Emite reportes de los modelos de diseño y sus elementos
Expresividad de los Modelos	No posee
Uso de Patrones	No posee
Soporte para la Regeneración de Modelos	No posee
Transformaciones Intra-Modelo	No posee
Trazabilidad	Facilidad para detectar inconsistencia Verifica consistencia entre los modelos Grado de satisfacción de la verificación de consistencia realizada Permite trazar los cambios de un modelo a otro Facilidad para trazar los cambios de un modelo a otro Emite reportes de los modelos de análisis y sus elementos Emite reportes de los modelos de diseño y sus elementos
Ciclo de Vida	Total de etapas soportadas Tasa de cobertura de las etapas requeridas
Estandarización	Solo estándares definidos por la organización: La herramienta genera productos que están acordes con los estándares establecidos por la organización La herramienta permite definir los estándares que deben llevar los productos de la organización
Control y Refinamiento de las Transformaciones	No posee
Calidad del Código Generado	Grado de completitud del código generado Calidad del código generado Cantidad de lenguajes en que genera código Generación de códigos Ejecutables para distintas plataformas Tasa de cobertura del código generado Modifica el formato y estructura del código fuente en uno o más lenguajes. (reducir el tamaño del código, reducir el tiempo de ejecución). Grado de satisfacción de la modificación y reestructuración del código fuente Número de lenguajes en que reestructura el código fuente Tasa de cobertura de la reestructuración de código fuente Permite traducir código fuente de un lenguaje a otro Número de lenguajes en que traduce el código fuente Tasa de cobertura de traducción del código fuente Grado de satisfacción de la traducción del código fuente

Herramientas de Soporte	El modelo MC-CASE si tiene métricas para evaluar si la herramienta hace modelado, pruebas, codificación, pero no una métrica explícita que establezca si la herramienta hace pruebas, por ejemplo.
-------------------------	--

Tabla N° 70. Nivel de Cobertura del MC-CASE con respecto a las Características de García y Rodríguez

Fuente: Elaboración Propia.

Como puede observarse casi el 60% de los atributos que tomaron en cuenta García y Rodríguez no son cubiertos por el Modelo MC-CASE, entre estos atributos están: Soporte para MIP, Soporte para MEP, Permite varias implementaciones, Acceso a la Definición de las Transformaciones, Expresividad de los Modelos, Uso de Patrones, Soporte para la Regeneración de Modelos, Transformaciones Intra-Modelo y Control y Refinamiento de las Transformaciones. Es importante señalar que todos estos atributos son los que están relacionados directamente con que la herramienta evaluada sea MDA.

Ahora bien de acuerdo con las características planteadas por Stuart Kent, tal como se muestra en la Tabla N° 71, el 83% de las características de Kent no son evaluadas por el MC-CASE y todas están relacionadas con el enfoque MDA.

Características según Kent	Aspectos del MC-CASE que cubren la característica
Orientadas a verificar que los modelos están bien formados	Posee métricas orientadas a la definición de diagramas pero no de modelos bajo el enfoque MDA
Permiten trabajar con instancias de modelos	No posee
Ofrecen soporte para la transformación entre modelos	No posee
Verificación orientada a modelos	No posee
Dedicadas al control de versiones y el trabajo distribuido	Tiene métricas para el control de versiones como por ejemplo: Facilidad de integración de versiones Permite el registro de versiones Facilidad de búsqueda de las versiones Permite la creación de versiones de un producto a partir de otro Grado de satisfacción de la generación de versiones Registra las versiones creadas
Apoyan los procesos de desarrollo de software dirigido por modelos	No posee

Tabla N° 71. Nivel de Cobertura del MC-CASE con respecto a las Características de Kent
Fuente: Elaboración Propia.

De las características propuestas por Jezequel, ninguna es cubierta por el MC-CASE (ver Tabla N° 72), es decir el nivel de cobertura es totalmente nulo.

Características según Jezequel	Aspectos del MC-CASE que cubren la característica
Lenguajes de Programación de Propósito General	No posee
Herramienta de Transformación Genérica	No posee

Herramientas CASE de Lenguajes Interpretados	No posee
Dedicadas a la Transformación de Modelos	No posee
De Meta-Meta-modelado	No posee

Tabla N° 72. Nivel de Cobertura del MC-CASE con respecto a las Características de Jezequel Fuente: Elaboración Propia.

Al igual que con Jezequel, el MC-CASE no cubre ningún aspecto de los propuestos por Czarnecki y Helsén, y todos están relacionados con las transformaciones de modelo a código y de modelo a modelo, aspectos muy importantes para el enfoque MDA. (Ver Tabla N° 73)

Características según Czarnecki y Helsén	Aspectos del MC-CASE que cubren la característica
Enfoque basado en el patrón "visitor"	No posee
Enfoque basado en plantillas	No posee
Enfoque de manipulación directa	No posee
Enfoque relacional	No posee
Enfoque basado en transformación de grafos	No posee
Enfoque centrado en la estructura	No posee
Enfoque híbridos	No posee

Tabla N° 73. Nivel de Cobertura del MC-CASE con respecto a las Características de Czarnecki y Helsén Fuente: Elaboración Propia.

Tal como se muestra en la Tabla N° 74 el nivel de cobertura del MC-CASE con respecto a las características propuestas por Tariq y Akhter, es del 0%, y todas son características muy importantes para indicar que una herramienta es MDA.

Características según Tariq y Akhter	Aspectos del MC-CASE que cubren la característica
Soporte de CIM	No posee
Soporte de PIM	No posee
Modelos MOF compatible	No posee
Soporte de Diagrama de Caso de Uso	No posee
Soporte de Diagrama de Clase	No posee
Soporte de Diagrama de Secuencia	No posee
Soporte para AS UML	No posee
Soporte para XMI-XML	No posee
Soporte Query-View-Transformation	No posee
Soporte OCL	No posee
Transformación PIM a PSM a Código	No posee
Transformación PIM a PSM a Código	No posee
Transformación directa PIM a Código	No posee
Transformación directa Código a PIM	No posee
Transformación basada en marcas	No posee
Transformación basada en meta-modelos	No posee
Transformación basada en modelos	No posee
Transformación basada en patrones	No posee
Transformación basada en perfiles UML	No posee
Registro de transformaciones	No posee

Trazabilidad de transformaciones	No posee
Combina 2 o más CIM en 1	No posee
Combina 2 o más PIM en 1	No posee

Tabla N° 74. Nivel de Cobertura del MC-CASE con respecto a las Características de Tariq y Akhter
Fuente: Elaboración Propia.

Una vez más el MC-CASE no cubre ningún aspecto, en este caso se hace referencia a los propuestos por Herrera, Matteo y Díaz.

Características según Herrera, Matteo y Díaz	Aspectos del MC-CASE que cubren la característica
Transformación CIM a PIM	No posee
Transformación PIM a CIM	No posee
Transformación CIM a PIM a PSM	No posee
Transformación PSM a PIM a CIM	No posee
Transformación Texto a MOF	No posee
Transformación MOF a Texto	No posee
Transformación de Texto a CIM	No posee
Transformación de CIM a Texto	No posee

Tabla N° 75. Nivel de Cobertura del MC-CASE con respecto a las Características de Herrera, Matteo y Díaz
Fuente: Elaboración Propia.

En cuanto a la propuesta de características hechas por Cervantes y Riba en su investigación, el MC-CASE no cubre ningún aspecto de los relacionados directamente con el enfoque MDA, observe la Tabla N° 76. Mientras que los aspectos externos a la herramienta como, que tan comercial es la herramienta y la facilidad de uso y de aprendizaje de la misma, si están totalmente cubiertos por el MC-CASE.

Características según Cervantes y Riba	Aspectos del MC-CASE que cubren la característica
PSM a código de cualquier plataforma	El Modelo MC-CASE verifica la transformación al código, pero no se especifica a partir de que, es decir, es a partir de un diagrama pero no dice si es un MIP o un MEP.
PIM a PSM	No posee
Editor de metamodelos	No posee
Editor de reglas de transformación	No posee
Facilidad adaptación a un dominio en particular	Lo cubre de forma muy parcial a través de la métrica: Soporta el tipo de aplicación que el usuario desee desarrollar Grado de satisfacción del soporte de la herramienta con respecto a la aplicación requerida por la organización
Comercial	Costos de adquisición. Tasa de cobertura del costo de adquisición de la herramienta.
Dificultad de uso y aprendizaje	La herramienta es intuitiva Facilidad de uso Horas de entrenamiento requeridas para lograr la experticia necesaria para el uso completo de la herramienta, en una persona que desconozca la misma.

Tabla N° 76. Nivel de Cobertura del MC-CASE con respecto a las Características de Cervantes y Riba
Fuente: Elaboración Propia.

Y para finalizar esta comparación se tienen las características propuestas por Bollati y sus compañeros, observe la Tabla N° 77, en la cual se aprecia como de manera reiterativa el MC-CASE no cubre los aspectos relacionados con el enfoque MDA de manera directa, y por el contrario si cubre lo relativo a generación de código, plataformas y tecnologías soportadas, ámbito de la aplicación, extensibilidad y usabilidad.

Características según Bollati et al	Aspectos del MC-CASE que cubren la característica
Niveles que cubre	No posee
Grado de generación de código	Grado de completitud del código generado Calidad del código generado Cantidad de lenguajes en que genera código Generación de códigos Ejecutables para distintas plataformas NOTA: pero no establece de donde proviene el código de manera específica
Transformaciones	No posee
Interacción con el usuario	No posee
Tipo de transformaciones	No posee
Lenguaje de almacenamiento y gestión de modelos	No posee
Plataformas y tecnologías soportadas	Cantidad de lenguajes en que genera código Número de manejadores de bases de datos en que genera esquemas Portabilidad a otras plataformas de hardware Los productos de la h. CASE se pueden ejecutar para distintas plataformas de hardware Se ejecuta sobre varios sistemas operativos Cantidad de sistemas operativos sobre los cuales opera la herramienta La herramienta se ejecuta bajo plataformas Windows Cantidad de plataformas de software que soportan a la herramienta. Tasa de la capacidad de la plataforma de la organización para soportar la herramienta Cantidad de plataformas de software que posee la organización Total de manejadores de bases de datos que maneja.
Ámbito de aplicación	Soporta el tipo de aplicación que el usuario desee desarrollar. Grado de satisfacción del soporte de la herramienta con respecto a la aplicación requerida por la organización.
Uso de Estándares	No posee
Extensibilidad	Grado de mejoramiento de la herramienta El desarrollador de la H. CASE genera "upgrade" de la herramienta. Grado de facilidad para obtener los "upgrade" de la herramienta. Frecuencia con la cual el proveedor lanza al mercado los "upgrade".
Usabilidad	La herramienta es intuitiva Facilidad de uso Horas de entrenamiento requeridas para lograr la experticia necesaria para el uso completo de la herramienta, en una persona que desconozca la misma.
Interoperabilidad entre herramientas	No posee

Tabla N° 77. Nivel de Cobertura del MC-CASE con respecto a las Características de Bollati et al
Fuente: Elaboración Propia.

Una vez hecho el análisis semántico exhaustivo al Modelo MC-CASE en cuanto a varias perspectivas, desde el punto de vista arquitectural, desde el punto de vista de la aplicación del

modelo y desde el punto de vista de la comparación con las distintas caracterizaciones que se han hecho hasta ahora acerca de herramientas de desarrollo de software MDA, se concluye que el MC-CASE necesita actualizarse para poder cubrir tan importante elemento dentro de la Ingeniería del Software, como lo son las herramientas de desarrollo de software. Para dar respuesta a esta necesidad se tiene en el siguiente capítulo la propuesta de un Modelo de Calidad que permita evaluar herramientas de desarrollo de software, incluyendo las herramientas MDA, Un modelo que se pretende presentar no solo a nivel teórico sino también a nivel matemático, lo cual le da mayor robustez al modelo y a su aplicación, haciendo del mismo un modelo mucho más objetivo que el actual.

CAPÍTULO VI

PROPUESTA DE UN META-MODELO DE CALIDAD PARA EVALUAR HERRAMIENTAS DE DESARROLLO DE SOFTWARE CON SOPORTE A MDA

Este capítulo centra su atención en plantear la propuesta de un **Meta-Modelo de Calidad para Evaluar herramientas de desarrollo de software incluyendo el soporte al enfoque MDA**. Este Meta-Modelo esta compuesto a su vez por tres modelos, el Modelo del Producto, el Modelo del Proceso y el Modelo Humano. El **Modelo del Producto** esta asociado con el producto real que se crea en esta propuesta, y que será instanciado en cada evaluación que se realice a una herramienta de desarrollo; el **Modelo del Proceso** contiene todos los pasos que se deben seguir para aplicar el producto y poder obtener unos resultados, y el **Modelo Humano** plantea como debe estar conformado el grupo de personas que deberán aplicar el proceso para generar unos resultados utilizando el producto. La Figura N° 28, ilustra el contenido del Meta-Modelo de Calidad para HDS, a través de un diagrama de clases.

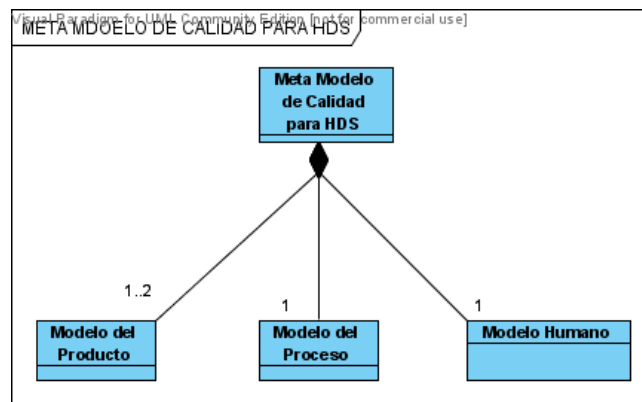


Figura N° 28. Meta-Modelo de Calidad para HDS
Fuente: Elaboración Propia.

A continuación se desarrollarán los Modelos del Producto, del Proceso y el Humano respectivamente.

6.1 MODELO DEL PRODUCTO

El Modelo del Producto esta formado por dos grandes modelos, en primer lugar el **Modelo Teórico**, el cual describe toda la arquitectura del MC-HDS-MDA, mostrando todos sus niveles de

abstracción, y en segundo lugar para darle mayor robustez al modelo teórico se propondrá un **Modelo Matemático** que valide las mediciones que se hagan a la hora de experimentar con el modelo teórico y ayude a agregarle objetividad a los resultados de la aplicación del Modelo de calidad para evaluar herramientas de desarrollo de software con soporte a MDA (MC-HDS-MDA). A continuación se describirá el Modelo Teórico.

6.1.1 PROPUESTA DEL MODELO TEÓRICO MC-HDS-MDA.

La propuesta del MC-HDS-MDA es una actualización del MC-CASE, la cual obedece al hecho comprobado de que no cubre los aspectos relativos a la evaluación de una herramienta CASE de hoy en día, tal como se evidenció en el Capítulo V, lógicamente si el objeto de evaluación evoluciona, el modelo de calidad usado para hacer la evaluación también debe evolucionar, y poder brindarle un apoyo más real y más actualizado al sector productivo encargado de desarrollar productos software.

Para garantizar la comprensión del Modelo Teórico inicialmente se plantearán un conjunto de definiciones que servirán de vocabulario técnico para construir el modelo de calidad, posteriormente se presentará los aspectos obligatorios de MDA que debe soportar el modelo, luego se describirá la cadena de valor del MC-HDS-MDA, se construirá la arquitectura distribuida por capas de abstracción, y posteriormente se hará la propuesta de métricas que finalmente conformarán al Modelo Teórico. A continuación se detallarán cada uno de estos elementos.

6.1.1.1 VOCABULARIO DEL MC-HDS-MDA.

En esta sección serán definidos los términos básicos que se usarán dentro del lenguaje del Modelo de Calidad para evaluar herramientas de desarrollo de software con soporte a MDA, importantes para garantizar la comprensión del modelo por parte del lector.

- **Capa:** se refiere a un nivel de abstracción dentro del modelo. El modelo se estructura de forma Top-Down, es decir de arriba hacia abajo, por lo que, la capa 1, será el nivel de abstracción más general, y la capa n, será el nivel de abstracción más concreto u específico.
- **Categoría, Característica, Tipo de Característica, Subcaracterística, Subcaracterística atómica y Métricas:** son tipos de capas, y están asociadas a un nivel de abstracción dentro del Modelo.
- **Métrica:** es un rasgo que posee la Herramienta de Desarrollo de Software, susceptible de ser valorado por un evaluador. Las métricas se dividen en Bolean, Escala de Liker y Tasa de Cobertura.

- **Métrica – Boolean:** es una métrica dicotómica, solo admite un si o un no, que equivale a un 1 si es un si, y un 0 si es un no.
- **Métrica – Escala de Liker:** esta métrica posee una escala que va del 0 al 4, dependiendo de lo que se este midiendo, es considerada una métrica subjetiva, porque depende de la percepción que tiene el evaluador acerca de un rasgo o una función de la HDS. En este tipo de métrica se encuentran los grados de conformidad, grados de satisfacción, grados de facilidad, entre otros; donde el valor de 4 esta asociado a la mejor apreciación que se tiene de un rasgo, y el 0 se asocia a la ausencia de ese rasgo.
- **Métrica – Tasa de Cobertura:** esta es considerada una métrica **crítica**, en todos los casos esta asociada con una restricción o necesidad que impone como prioritaria el cliente para quien se hace la evaluación. Esta métrica relaciona la proporción de cumplimiento que tiene la HDS con respecto a la necesidad planteada. En el mejor de los casos tendrá el valor de 1, lo cual significa que satisface totalmente el requisito del cliente, y en cualquier otro caso puede asumir un valor real entre 0 y 1.

6.1.1.2 ASPECTOS QUE DEBE SOPORTAR FORMALMENTE.

Uno de los aspectos que debe soportar el modelo propuesto MC-HDS-MDA es el enfoque MDA el cual establece todo un proceso de desarrollo a través de las transformaciones entre los modelos que se van generando durante la construcción de un producto software. Este enfoque queda resumido en la Figura N° 29, que muestra a través de una cadena de valor los procesos medulares que soporta MDA y los procesos de apoyo sobre los cuales se basa dicho enfoque.

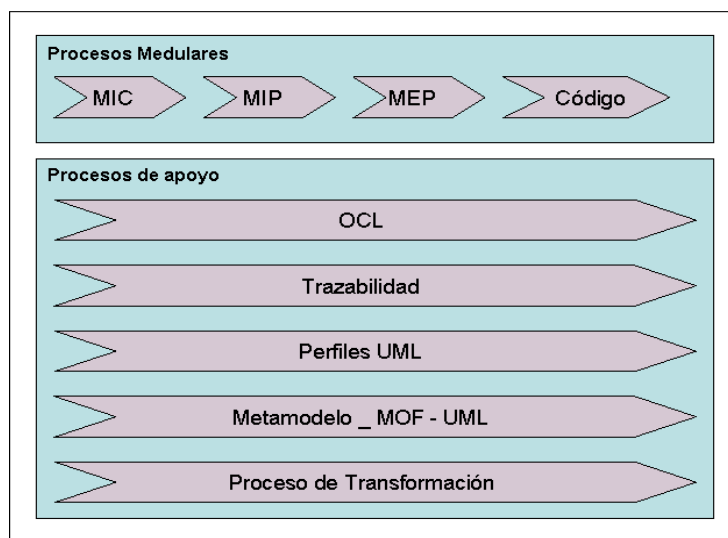


Figura N° 29. Cadena de Valor del Enfoque MDA
Fuente: Elaboración Propia.

Estos procesos son muy importantes para poder considerar a la herramienta de desarrollo una herramienta MDA, y debido a su importancia y a que también existen herramientas de desarrollo muy valoradas que formalmente no son MDA, es que en este modelo se crea toda una Subcaracterística para MDA, en tal sentido si una herramienta no es MDA no se recomienda que en la instanciación del modelo se le incluya, dado que arrojará valores que no favorecerán a la herramienta que se este evaluando.

Tal como lo muestra la figura N° 30, donde se observa la equivalencia del proceso de desarrollo tradicional y el proceso de desarrollo a través de las transformaciones de modelos, el modelo propuesto en esta investigación busca soportar ambos procesos.

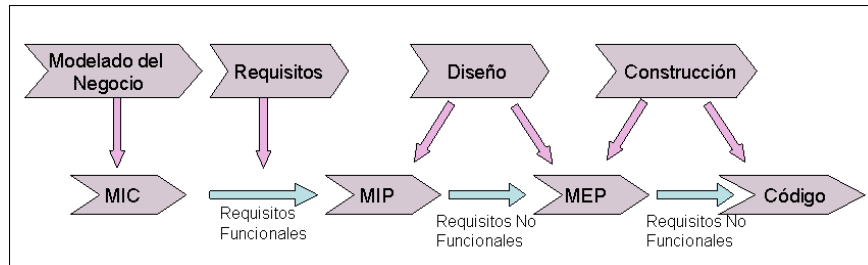


Figura N° 30. Relación entre un Proceso de Desarrollo Tradicional y el Enfoque MDA
Fuente: Elaboración Propia.

6.1.1.3 CADENA DE VALOR DEL MODELO DE CALIDAD (MC-HDS-MDA).

De acuerdo a la estructura arquitectónica que se le dará al MC-HDS-MDA a continuación se presenta un resumen del modelo a nivel de procesos, tomando como base la **cadena de valor**, en tal sentido el modelo tiene dos **procesos medulares** que son **Ciclo de Vida**, que determina la razón de ser de la herramienta de desarrollo de software, y **Cliente**, quien establece las necesidades de uso que le dará a la herramienta de desarrollo. Y posteriormente se encuentran los **procesos de apoyo**, como se puede observar en la Figura N° 31, se tiene a **Calidad**, es decir que tan buena es la herramienta como producto de software, y a **Comercialización**, que relaciona los elementos asociados con el proveedor de la HDS.

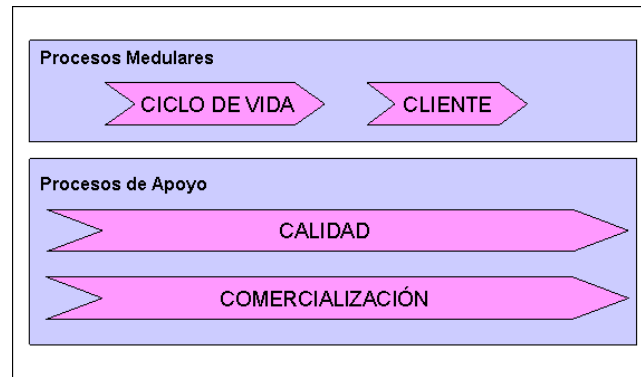


Figura N° 31. Cadena de Valor del MC-HDS-MDA
Fuente: Elaboración Propia.

6.1.1.4 ARQUITECTURA DEL MODELO DE CALIDAD (MC-HDS-MDA).

El Modelo de Calidad para evaluar herramientas de desarrollo de software con soporte a MDA estará basado en el MC-CASE, debido a que éste a su vez estaba fundamentado en lo que plantea la ISO 14102 que es un estándar internacional propuesto por la ISO para evaluar herramientas CASE, entonces el MC-HDS-MDA quiere tomar las ventajas que posee el MC-CASE. Otra particularidad importante del modelo propuesto es que se quiere que sea un modelo sistémico, es decir, perfectamente adaptable a las necesidades de la evaluación.

El MC-HDS-MDA esta compuesto por seis capas de abstracción, de arriba hacia abajo se tienen: capa 1 – **categorías**, capa 2 – **característica**, capa 3 – **tipo de característica**, capa 4 – **subcaracterística**, capa 5 – **subcaracterística atómica** y capa 6 – **métricas**.

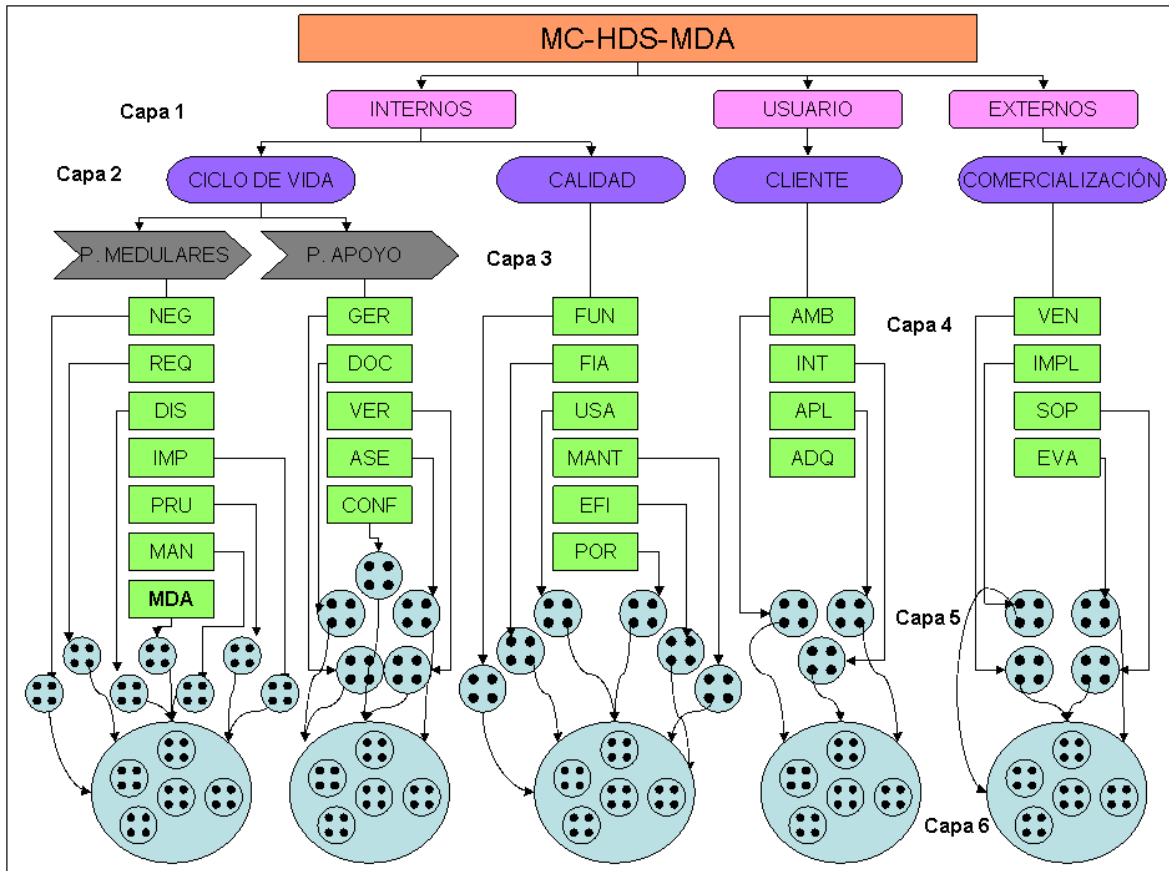


Figura N° 32. Arquitectura del MC-HDS-MDA
Fuente: Elaboración Propia.

A continuación se describirán las capas ilustradas en la Figura N° 32.

- Capa 1 – Categorías:** Esta es la capa de mayor nivel de abstracción y se refiere al tipo de aspectos que se están evaluando, esta capa se divide en tres categorías: Internos, Usuario y Externos. La categoría Internos se refiere a todos elementos involucrados en la evaluación que sean propios de la herramienta, en cuanto a su funcionalidad y a su calidad. La categoría Usuario, hace referencia a todos los aspectos relacionados con la funcionalidad de la herramienta esperada por la persona o la organización para quien se hace la evaluación de la herramienta de desarrollo. Y la categoría Externos esta relacionada con los elementos que no tienen que ver ni con la funcionalidad de la herramienta, ni con su calidad, ni con el cliente para el cual se hace la evaluación; esta categoría esta asociada directamente con información que depende de la calidad del desarrollador de la herramienta y del grado de aceptación que ha tenido la herramienta dentro del sector productivo de software.
- Capa 2 – Características:** Esta capa hace una subclasificación más específica de cada una de las categorías descritas anteriormente. En tal sentido se tiene a la categoría Internos, que se divide en primer lugar en la característica Ciclo de Vida, con todo lo relacionado al

proceso de desarrollo de software, donde tradicionalmente se tienen a cada una de las grandes etapas del proceso de desarrollo separadas, esta característica relaciona todo lo que tiene que ver con la funcionalidad para la cual ha sido creada la herramienta CASE. Y en segundo lugar la característica Calidad, la cual tiene asociado todos los aspectos relacionados con la medición de la calidad de un producto software, y está basada en el estándar internacional ISO 9126. Luego se tiene a la categoría Usuario, ésta está subdividida en la característica Cliente, que reúne todos los elementos que están íntimamente involucrados con las necesidades que tiene el cliente como usuario que desea adoptar la herramienta para soportar sus procesos de desarrollo. Y por último tiene la característica Comercialización, que depende directamente de la categoría Externos; esta característica engloba todos los aspectos asociados con el proceso de comercialización de la herramienta de desarrollo.

- **Capa 3 – Tipo de Característica:** Esta capa por ahora sólo la posee la característica Ciclo de Vida y se divide en dos, el tipo de característica Procesos Medulares y el Tipo de característica Procesos de Apoyo. Esta capa se crea tomando como base la estructuración que tienen los procesos de desarrollo de hoy en día por ejemplo: RUP, Métrica, Watch, entre otros, que separan los procesos medulares de los procesos de apoyo, por lo tanto en este trabajo de investigación se quiere destacar la inclinación que tiene la herramienta de desarrollo de soportar procesos medulares o procesos de apoyo.
- **Capa 4 – Subcaracterísticas:** Esta es la cuarta capa del MC-HDS-MDA, y nace de una especificación más concreta de los aspectos que involucra la capa 3 como es el caso de la característica ciclo de vida; o de la capa 2 – características para los casos de Calidad, Cliente y Comercialización. Por ejemplo, en el caso de la Característica de Calidad se hace una subclasificación de esta y nacen las subcaracterísticas: Funcionalidad, Fiabilidad, Portabilidad, Usabilidad, Mantenibilidad y Eficiencia.
- **Capa 5 – Subcaracterísticas Atómicas:** Es la quinta capa del modelo y corresponde a un nivel más bajo de abstracción en cuanto al manejo de las subcaracterísticas; centra la atención en aspectos muy particulares de cada subcaracterística, por ejemplo, en el caso de la Subcaracterística Indicadores de Soporte, señalada en el modelo arquitectónico como **SOP**, posee las siguientes subcaracterísticas atómicas: perfil del vendedor, perfil del producto y disponibilidad de entrenamiento. Es importante recordar, que éste modelo esta basado en el estándar internacional para evaluar herramientas CASE 14102 y éstas subcaracterísticas son propuestas por este estándar. El modelo MC-HDS-MDA propone un conjunto de subcaracterísticas atómicas provenientes de las subcaracterísticas NEG

(Modelado de Negocio), REQ (Ingeniería de Requisitos), MDA (Arquitectura Dirigida por Modelos), ADQ (Proceso de Adquisición) y VEN (Proceso de Venta).

- **Capa 6 – Métricas:** Es la sexta y última capa del MC-HDS-MDA, ésta capa hace referencia a un conjunto de métricas que se proponen por cada una de las subcaracterísticas atómicas del modelo. Estas métricas son los aspectos más concretos del MC-HDS-MDA, y los verdaderamente medibles. Cada una de las métricas tiene unos valores de referencia por medio de los cuales el evaluador colocará una respuesta numérica como resultado de la medición de la métrica.

Una vez descritas las capas arquitectónicas del MC-HDS-MDA, se detallarán cada uno de los elementos que lo conforman, tal como lo muestran las Tablas N° 78, 79 y 80 en las cuales se presenta el contenido relacionado con las características, las características tipo, las subcaracterísticas, y las subcaracterísticas atómicas de cada categoría del Modelo Propuesto.

CATEGORÍA INTERNOS		
CARACTERÍSTICA CICLO DE VIDA: Son todas las características asociadas con un proceso de desarrollo de software.		
Característica Tipo	Subcaracterísticas	Subcaracterísticas atómicas
Procesos Medulares. Se refiere a los características que pertenecen solo a los procesos medulares del desarrollo de software	Modelado del Negocio (NEG): se refiere a un conjunto de atributos relacionados con el modelado del negocio actual donde se enmarcará la solución software que se pretende mantener o desarrollar.	Modelos del negocio Lenguajes de modelado Proceso de modelado
	Ingeniería de Requisitos (REQ): tiene asociado un conjunto de atributos relacionados con el proceso de ingeniería de requisitos.	Especifica requisitos funcionales Modela requisitos funcionales Gerencia el proceso de ingeniería de requisitos
	Proceso de Diseño (DIS): se refiere a un conjunto de atributos relacionados con el proceso de diseño de la aplicación software.	Desarrollo de Diagramas Análisis de diagramas Soporte de especificación de diseño Modelado de la estructura de los procesos de software. Simulación Prototipo Modelado de la interfaz humana
	Proceso de Implementación (IMP): se refiere a un conjunto de atributos relacionados con el proceso de construcción o programación de la aplicación software.	Generación de código Generación de esquemas de bases de datos Generación de pantallas Generación de reportes Compilación Edición dirigida por la sintaxis Depurador
	Proceso de Pruebas (PRU): se refiere a un conjunto de atributos relacionados con el proceso de probar una aplicación software.	Prueba técnica de correctitud Análisis de fallas Análisis de defectos Casos de prueba y espera de los datos de entrada Casos de prueba y generación de

		<ul style="list-style-type: none"> resultados esperados Prueba de trazabilidad Instrumentación de código fuente Captura y reutilización de los datos de entrada Manejo de pruebas Análisis en tiempo de corrida Análisis de fiabilidad Análisis de alcance de pruebas Gerencia del proceso de pruebas
	Proceso de Mantenimiento (MAN): se refiere a un conjunto de atributos relacionados con el proceso de mantenimiento de una aplicación software.	<ul style="list-style-type: none"> Entendimiento del problema Localización Análisis de impacto Ingeniería de reverso de datos Ingeniería de reverso de procesos-procedimientos Reestructuración del código fuente Traducción del código fuente
	Proceso MDA (MDA): se refiere a un conjunto de atributos relacionados con el proceso de desarrollo de una aplicación software bajo el enfoque MDA o Arquitectura dirigida por Modelos.	<ul style="list-style-type: none"> Soporte a MIC Soporte a MIP Soporte a MEP Soporte a Código Proceso de transformación Soporte a OCL Soporte a perfiles Soporte a metamodelo – MOF y UML Soporte a MIC – MIP – MEP - Código
Procesos de Apoyo: Son todos los procesos dentro del ciclo de vida que sirven de apoyo a los procesos medulares o que están presentes durante todo el desarrollo de los procesos medulares.	Proceso de Gerencia (GER): Esta relacionado con una serie de atributos que se basan en la existencia de un conjunto de funciones y sus propiedades específicas para soportar las actividades del proceso de gerencia.	<ul style="list-style-type: none"> Estimando costos y planificación Planificación Seguimiento del proyecto Análisis y reportes del estatus del proyecto
	Proceso de Documentación (DOC): Esta subcaracterística esta asociada con un conjunto de atributos que sirven para caracterizar a una herramienta de desarrollo de software para el apoyo del proceso de documentación dentro del desarrollo de una aplicación.	<ul style="list-style-type: none"> Edición de texto Edición gráfica Edición basada en formas Publicidad Soporte de hipertexto Manejador de variantes Extracción automática de datos y generación de documentos
	Proceso de Verificación (VER): Estos atributos permiten determinar si la herramienta soporta el proceso de verificación.	<ul style="list-style-type: none"> Análisis de especificación de trazabilidad Análisis de especificación Análisis de código fuente
	Proceso de Aseguramiento de la Calidad (ASE): Estos atributos permiten determinar si la herramienta soporta el proceso de aseguramiento de la calidad.	<ul style="list-style-type: none"> Gerencia de la calidad de los datos Gerencia de riesgos
	Proceso de Gestión de Configuración (CONF): Estos atributos permiten verificar si la herramienta soporta el proceso de gerencia de la configuración.	<ul style="list-style-type: none"> Control de acceso Búsqueda de modificaciones Definición y gerencia de versiones múltiples Estatus de configuración Generación de versiones Capacidad de archivo
	CARACTERÍSTICA CALIDAD: Son todas las características asociadas con la medición de la calidad de un producto software basado en el estándar internacional ISO 9126.	
Subcaracterísticas		Subcaracterísticas atómicas
Funcionalidad (FUN): Se refiere a un conjunto de atributos que miden si la herramienta de desarrollo de software hace lo que especifican sus proveedores.		<ul style="list-style-type: none"> Seguridad Exactitud Restricciones legales

	Restricciones técnicas
Fiabilidad (FIA): Se refiere a un conjunto de atributos que miden la capacidad de la herramienta de desarrollo de software para mantener su nivel de ejecución bajo condiciones específicas por un periodo de tiempo determinado.	Integridad de los datos Backup automático Manejo de errores Tolerancia de fallas Recuperabilidad
Usabilidad (USA): Conjunto de atributos que miden la capacidad de la herramienta de ser fácil de usar.	Amigabilidad del usuario Guía del usuario Homogeneidad Adaptabilidad Claridad de control Manejo de errores Concisión Fácil de aprender Calidad de documentación de la herramienta Fácil de instalar
Mantenibilidad (MANT): Se refiere a un conjunto de atributos que se basan en medir si la herramienta es fácilmente modificable para que cumpla con ciertos requisitos del usuario.	Soporte del vendedor Habilidad de la herramienta para hacer cambios en la metodología Actualización Expansibilidad
Eficiencia (EFI): Esta relacionada con un conjunto de atributos que se basan en la relación entre el nivel de ejecución de la herramienta y la cantidad de los recursos usados bajo ciertas condiciones específicas.	Tiempo de respuesta aceptable Requerimientos de almacenamiento de datos Capacidad de memoria aceptable Velocidad de procesamiento aceptable
Portabilidad (POR): Conjunto de atributos que se basan en medir la capacidad que tiene la herramienta para ser transportada de un ambiente a otro.	Portabilidad a diferentes plataformas de hardware Compatibilidad con diferentes sistemas operativos Habilidad para mover datos entre versiones de la herramienta Portabilidad con los sistemas Windows Portabilidad con sistemas Linux

Tabla N° 78. Arquitectura de la Categoría Internos del MC-HDS-MDA
Fuente: Elaboración Propia.

Como puede observarse en esta tabla, se modificó el nombre de las subcaracterísticas atómicas *Normativas regulatorias* y *Normativas técnicas* por, **Restricciones legales** y **Restricciones técnicas** respectivamente, por considerar los nombres propuestos más significativos. Por otro lado, al agregar la subcaracterística **Ingeniería de Requisitos**, se elimina automáticamente la Subcaracterística atómica *Soporte a las especificaciones de requisitos*, de la Subcaracterística Diseño.

También se ha modificado el nombre de la Subcaracterística atómica *Modelado de especificaciones de la estructura de los procesos* por **Modelado de la estructura de los procesos de software**, por considerarlo más acorde con los aspectos que miden las métricas asociadas a ella. Igualmente se modifica el nombre de la métrica *Casos de prueba y espera del resultado de entrada*, por el nombre **Casos de prueba y espera de los datos de entrada**, por considerarla más significativa para el evaluador.

De igual forma se modifica el nombre de la métrica *Captura de la entrada y repetición* por **Captura y reutilización de los datos de entrada**, por considerarla más significativa.

En cuanto a la Subcaracterística Portabilidad se agrega la subcaracterística atómica Portabilidad con sistemas Linux, toda vez que este sistema operativo ha demostrado robustez y cada vez más la comunidad de sus usuarios crece.

A continuación se muestra la Tabla N° 79, que muestra la Característica, las subcaracterísticas y las subcaracterísticas atómicas de la categoría Usuario.

CATEGORÍA USUARIO	
CARACTERÍSTICA CLIENTE: Son todas las características asociadas con los requerimientos que pueden estar asociados al cliente o al usuario de la herramienta de desarrollo de software.	
Subcaracterísticas	Subcaracterísticas atómicas
Ambiente en el cual opera la herramienta CASE (AMB): Esta subcaracterística reúne un conjunto de atributos relacionados con la correspondencia que debe existir entre la herramienta de desarrollo de software y su ambiente operacional.	Características de hardware requeridas por la herramienta Ambiente de software requerido por la herramienta Repositorio de software Ambiente físico de la herramienta
Integrabilidad de la herramienta CASE (INT): Conjunto de atributos que miden la habilidad de la herramienta de desarrollo de software para integrarse e interoperar con otros productos de software en su ambiente operacional.	Compatibilidad con los elementos del ambiente Integración de data Integración de control Integración de presentación Acceso a metadata
Aspectos de la aplicación de la herramienta CASE (APL): Conjunto de atributos que se basan en la relación entre la herramienta de desarrollo de software y los proyectos a los cuales es aplicada, incluyendo el ambiente de sus productos y las características de esos productos.	Ambiente de hardware y software de los productos de la herramienta Conformidad de estándares-productos de la herramienta Dominio de la aplicación Tamaño de la aplicación soportada Lenguajes soportados Bases de datos soportadas Metodologías soportadas Globalización
Proceso de Adquisición (ADQ): Conjunto de atributos relacionados con la capacidad que tiene el cliente para adquirir la herramienta de desarrollo.	Costos de implantación de la herramienta Documentación de la herramienta

Tabla N° 79. Arquitectura de la Categoría Usuario del MC-HDS-MDA
Fuente: Elaboración Propia.

La Subcaracterística atómica *Internacionalización* perteneciente a la Subcaracterística Aspectos de la Aplicación de la Herramienta CASE, se modificó por **Globalización**, por considerar este nombre más significativo.

También se ha agregado una nueva subcaracterística denominada **Proceso de Adquisición**, la cual depende exclusivamente de la capacidad económica que tiene el cliente de poder adquirir la herramienta de desarrollo de software, en consecuencia tiene asociada la subcaracterística atómica **Costos de implantación de la herramienta**. También se incorpora la subcaracterística atómica **Documentación de la herramienta** debido a que debe venir incluido algún tipo de documentación cuando un cliente u organización adquiere una HDS.

Ahora se muestra la Tabla N° 80, que detalla la Característica, las subcaracterísticas y las subcaracterísticas atómicas de la categoría Externos.

CATEGORÍA EXTERNOS	
CARACTERÍSTICA COMERCIALIZACIÓN: Son todas las características asociadas con el proveedor y el conjunto de desarrolladores de la herramienta de desarrollo de software	
Subcaracterísticas	Subcaracterísticas atómicas
Proceso de Venta (VEN): Se refiere al conjunto de atributos que permiten hacer una revisión de las condiciones necesarias durante la venta de la herramienta de desarrollo.	Políticas de licencias Restricciones de exportación
Implementación (IMP): Conjunto de atributos que permiten evaluar la entrega, instalación y operación de la herramienta de desarrollo de software.	Efectividad del costo Restricciones de desarrollo-entregas Actividades requeridas por los usuarios de la organización Necesidades de infraestructura
Indicadores de Soporte (SOP): Conjunto de atributos que miden la habilidad del vendedor para proporcionar soporte técnico acerca de la herramienta que comercializan.	Perfil del vendedor Perfil del producto Disponibilidad de entrenamiento
Evaluación o Certificación (EVA): Esta subcaracterística se refiere a un conjunto de atributos que permiten medir la certificación del desarrollador de la herramienta y de la certificación de la herramienta de desarrollo de software.	Evaluación o certificación del desarrollador Certificación del producto

Tabla N° 80. Arquitectura de la Categoría Externos del MC-HDS-MDA
Fuente: Elaboración Propia.

Al observar la Tabla N° 80, se puede apreciar que se eliminó la subcaracterística Proceso de Adquisición, y se sustituyó por **Proceso de Venta**, siendo ésta la más adecuada, tomando como perspectiva la del vendedor. Esta subcaracterística tiene asociada las subcaracterísticas atómicas: **Políticas de licencias**, y **Restricciones de exportación**.

Tomando como base el Análisis Semántico a nivel de Métricas que se hizo en el capítulo V, se tomarán las métricas que no obtuvieron ninguna observación, y de aquellas que si, se le hará la modificación que corresponda.

Tal como se indicó al inicio del capítulo, se procederá a la definición de las métricas del Modelo de calidad o capa 6.

6.1.1.5 PROPUESTA DE MÉTRICAS.

Las métricas del Modelo propuesto se describirán en las Tablas N° 81 a la 220. Las métricas propuestas se mostrarán clasificadas por Característica y por Subcaracterísticas.

CICLO DE VIDA. PROCESOS MEDULARES

Subcaracterística: Modelado del Negocio (NEG)

Subcaracterística Atómica: Modelos del Negocio (NEG 1)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Modela objetivos	¿La HDS permite modelar los objetivos de la organización?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción del modelado de objetivos	¿Qué tan satisfecho esta el diseñador con el modelado de objetivos realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0= No aplica	Escala de liker
Modela procesos de negocio	¿La HDS permite modelar los procesos de negocio de la organización?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción del modelado de procesos	¿Qué tan satisfecho esta el diseñador con el modelado de procesos realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0= No aplica	Escala de liker
Modela objetos de negocio	¿La HDS permite modelar los objetos de negocio de la organización?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción del modelado de objetos de negocio	¿Qué tan satisfecho esta el diseñador con el modelado de objetos de negocio realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0= No aplica	Escala de liker
Modela actores	¿La HDS permite modelar los actores de la organización?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción del modelado de actores	¿Qué tan satisfecho esta el diseñador con el modelado de actores realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0= No aplica	Escala de liker
Modela reglas del negocio	¿La HDS permite modelar las reglas del negocio de la organización?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción del modelado de reglas del negocio	¿Qué tan satisfecho esta el diseñador con el modelado de reglas del negocio realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0= No aplica	Escala de liker
Identifica conflictos o problemas	¿La HDS identifica posibles conflictos o problemas en cuanto a flujos de información, cuellos de botella o información no procesada?	0	1	No = 0 Si = 1	Bolean
Total de Métricas = 11					

Tabla N° 81. Propuesta de métricas de la Subcaracterística Atómica Modelos del Negocio del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Lenguajes de Modelado del Negocio (NEG 2)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Lenguaje estandarizado	¿La HDS utiliza un lenguaje estandarizado para modelar el negocio?	0	1	No = 0 Si = 1	Bolean
Tipo de lenguaje	¿Qué lenguaje de modelado utiliza la HDS: UML, BPMN, otros?	0	1	0 = Lenguaje de modelado no conocido 1 = cualquier lenguaje(s) conocidos	Bolean
Adaptaciones al lenguaje de modelado	¿La HDS permite hacer adaptaciones al lenguaje de modelado del negocio?	0	1	No = 0 Si = 1	Bolean

Total de Métricas = 3

Tabla N° 82. Propuesta de métricas de la Subcaracterística Atómica Lenguajes de Modelado de Negocio del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Proceso de Modelado (NEG 3)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Soporta procesos de modelado del negocio	¿La HDS soporta algún proceso de modelado de negocio?	0	1	No = 0 Si = 1	Bolean
Proceso de modelado soportado	¿Qué tipo de procesos de modelado de negocio utiliza la HDS: Metamodelo, UML Business, otros?	0	1	0 = proceso de modelado no conocido 1 = cualquier proceso(s) conocido	Bolean

Total de Métricas = 2

Tabla N° 83. Propuesta de métricas de la Subcaracterística Atómica Proceso de Modelado del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística: Ingeniería de Requisitos (REQ)

Subcaracterística Atómica: Especificación de Requisitos Funcionales (REQ 1)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Especifica requisitos Funcionales	¿La HDS permite especificar requisitos funcionales?	0	1	No = 0 Si = 1	Bolean
Utiliza plantilla	¿La HDS utiliza un plantilla para especificar los requisitos funcionales?	0	1	No = 0 Si = 1	Bolean
Adapta la plantilla	¿La HDS permite hacer adaptaciones a la plantilla para especificar los requisitos funcionales?	0	1	No = 0 Si = 1	Bolean
Clasifica requisitos	¿La HDS permite hacer algún tipo de clasificación de requisitos dentro de los re. Funcionales?	0	1	No = 0 Si = 1	Bolean

Total de Métricas = 4

Tabla N° 84. Propuesta de métricas de la Subcaracterística Atómica Especificación de Requisitos Funcionales del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Modelado de Requisitos (REQ 2)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Modela requisitos Funcionales	¿La HDS permite modelar requisitos funcionales?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción del modelado de requisitos	¿Qué tan satisfecho esta el diseñador con el modelado de requisitos realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Verifica consistencia	¿La HDS verifica la consistencia entre la especificación y el modelado de los requisitos funcionales?	0	1	No = 0 Si = 1	Bolean
Utiliza Diagramas de casos de uso	¿La HDS utiliza el diagrama de casos de uso para modelar los requisitos funcionales?	0	1	No = 0 Si = 1	Bolean
Construcción de	¿La HDS permite construir un prototipo	0	1	No = 0	Bolean

prototipo	para especificar requisitos funcionales?			Si = 1	
Generación automática del modelado de requisitos	¿La HDS genera de manera automática el modelado de requisitos a partir de la especificación de los requisitos funcionales?	0	1	No = 0 Si = 1	Boolean
Grado de satisfacción del modelado de requisitos automático	¿Qué tan satisfecho está el diseñador con el modelado de requisitos realizado por la HDS a partir de la especificación?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 7					

Tabla N° 85. Propuesta de métricas de la Subcaracterística Atómica Modelado de Requisitos del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Especificación de Requisitos No Funcionales (REQ 3)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Especifica requisitos No Funcionales	¿La HDS permite especificar requisitos no funcionales?	0	1	No = 0 Si = 1	Boolean
Clasifica requisitos no funcionales	¿La HDS permite hacer algún tipo de clasificación de requisitos dentro de los re. Funcionales?	0	1	No = 0 Si = 1	Boolean
Total de Métricas = 2					

Tabla N° 86. Propuesta de métricas de la Subcaracterística Atómica Especificación de Requisitos No Funcionales del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Gerencia del Proceso de ingeniería de requisitos (REQ 4)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Reuso de requisitos	¿La HDS permite el reuso de los requisitos?	0	1	No = 0 Si = 1	Boolean
Gestiona el proceso de ingeniería de requisitos	¿La HDS permite gestionar el proceso de ingeniería de requisitos?	0	1	No = 0 Si = 1	Boolean
Resuelva conflictos entre requisitos	¿La HDS chequea y resuelve algunos conflictos presentes en la especificación de los requisitos?	0	1	No = 0 Si = 1	Boolean
Total de Métricas = 3					

Tabla N° 87. Propuesta de métricas de la Subcaracterística Atómica Gerencia del proceso de Ingeniería de Requisitos del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística: Diseño (DIS)

Subcaracterística Atómica: Desarrollo de Diagramas (DIS 1)				
Nombre	Valor Mín	Valor Max	Formulación	Tipo de Métrica

Realiza diagramas	0	1	0 = No 1 = Si	Bolean
Relaciona requisitos con diseño Definición operacional: ¿Utiliza elementos de la especificación de los requisitos para generar elementos de diseño?	0	1	0 = No 1 = Si	Bolean
Utiliza UML para el modelado	0	1	0 = No 1 = Si	Bolean
Utiliza una versión de UML actualizada	0	1	0 = No 1 = Si	Bolean
Facilidad de diagramación	0	1	0 = No 1 = Si	Bolean
Calidad de la edición gráfica	0	1	0 = No 1 = Si	Bolean
Grado de facilidad de edición	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Porcentaje de diagramas estáticos que maneja	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Porcentaje de diagramas dinámicos que maneja	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Grado de satisfacción del desarrollo de diagramas	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Grado de completitud de la caja de herramientas para realizar los diagramas	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Permite importar objetos entre diagramas	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 12				

Tabla N° 88. Propuesta de métricas de la Subcaracterística Atómica Desarrollo de Diagramas del MC-HDS-MDA

Fuente: Elaboración Propia.

Como puede observarse en la Tabla N° 88, la métrica *Cantidad de diagramas* se ha transformado en las siguientes métricas **Porcentaje de diagramas estáticos que maneja** y **Porcentaje de diagramas dinámicos que maneja**. También se incorpora tres nuevas métricas; **Relaciona requisitos con diseño**, la cual pretende medir si la herramienta permite generar aspectos del diseño a partir de las especificaciones de los requisitos y **Utiliza UML para el modelado** y **Utiliza una versión de UML actualizada**, y de esta forma determinar explícitamente si la herramienta soporta el modelado con el lenguaje de modelado unificado.

Subcaracterística Atómica: Análisis de Diagramas (DIS 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite variaciones de visión de documentos	0	1	0 = No 1 = Si	Bolean
Verifica consistencia de los diagramas	0	1	0 = No 1 = Si	Bolean
Facilidad para detectar inconsistencia	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Herramienta de análisis	0	1	0 = No 1 = Si	Bolean
Verifica reglas semánticas entre los diagramas	0	1	0 = No 1 = Si	Bolean
Permite el almacenamiento de la información de diseño	0	1	0 = No 1 = Si	Bolean
Verifica reglas de diagramación	0	1	0 = No 1 = Si	Bolean

Permite la impresión de los diagramas	0	1	0 = No	1 = Si	Bolean
Emite reportes sobre los diagramas	0	1	0 = No	1 = Si	Bolean
Total de Métricas = 9					

Tabla N° 89. Propuesta de métricas de la Subcaracterística Atómica Análisis de Diagramas del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Soporte de Especificación de diseño (DIS 3)					
Nombre	Valor Min	Valor Max	Formulación		Tipo de Métrica
Grado de ajuste a las metodologías en la etapa de diseño	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica		Escala de liker
Facilidad para detectar inconsistencia.	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica		Escala de liker
Posee un lenguaje formal para representar el diseño de datos	0	1	0 = No	1 = Si	Bolean
Emite reportes de los datos del diseño	0	1	0 = No	1 = Si	Bolean
Clasifica los tipos de datos del diseño	0	1	0 = No	1 = Si	Bolean
Porcentaje de metodologías soportadas.	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%		Escala de liker
Grado de satisfacción de las especificaciones del diseño	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica		Escala de liker
Total de Métricas = 7					

Tabla N° 90. Propuesta de métricas de la Subcaracterística Atómica Soporte de Especificación de diseño del MC-HDS-MDA

Fuente: Elaboración Propia.

La métrica *Grado de consistencia de la diagramación respecto a la metodología adoptada por la organización*, es eliminada de esta subcaracterística atómica y pasa a formar parte de las métricas de la categoría de Usuario, por depender de la metodología que utiliza la organización como cliente. Se incorpora la métrica **Porcentaje de metodologías soportadas**, es virtud de que las especificaciones de diseño durante el desarrollo de un producto software depende de la metodología de desarrollo que se este instanciando.

Subcaracterística Atómica: Modelado de la estructura de los procesos de software (DIS 4)					
Nombre	Valor Min	Valor Max	Formulación		Tipo de Métrica
Permite modelar procesos de software	0	1	0 = No	1 = Si	Bolean
Permite modelar la estructura de datos asociada a los procesos de software	0	1	0 = No	1 = Si	Bolean
Permite modelar la estructura dinámica asociada a los procesos de software	0	1	0 = No	1 = Si	Bolean
Grado de satisfacción de la edición de modelos de especificación de procesos	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica		Escala de liker
Total de Métricas = 4					

Tabla N° 91. Propuesta de métricas de la Subcaracterística Atómica Modelado de las especificación de la estructura de los procesos del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Simulación (DIS 5)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite hacer simulaciones de la utilidad operacional.	0	1	0 = No 1 = Si	Bolean
Permite hacer simulaciones de la interfaz con el operador	0	1	0 = No 1 = Si	Bolean
Permite hacer simulaciones del tiempo de respuesta	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción con respecto a las simulaciones hechas por la herramienta.	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 4				

Tabla N° 92. Propuesta de métricas de la Subcaracterística Atómica Simulación del MC-HDS-MDA

Fuente: Elaboración Propia.

En la Tabla N° 92, se puede apreciar que se ha eliminado la métrica *Permite hacer simulaciones del desempeño*, por considerarla exactamente igual a la métrica **Permite hacer simulaciones del tiempo de respuesta**.

Subcaracterística Atómica: Prototipo (DIS 6)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite la realización de prototipo.	0	1	0 = No 1 = Si	Bolean
Nivel de generación de prototipo	0	4	4 = Va más allá 3 = Completo 2 = Med. Completo 1 = Incompleto 0 = No está definido	Escala de liker
Grado de satisfacción de la generación de prototipo	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 93. Propuesta de métricas de la Subcaracterística Atómica Prototipo del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Modelado de la Interfaz Humana (DIS 7)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite modelar las interfaces. (menús, ventanas, reportes, etc.)	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción con respecto al modelado de interfaces.	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Facilidad para modelar interfaces	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logra realizar 0 = No aplica	Escala de liker
Riqueza de la paleta	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Total de Métricas = 4				

Tabla N° 94. Propuesta de métricas de la Subcaracterística Atómica Modelado de la Interfaz Humana del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística: Implementación (IMP)

Subcaracterística Atómica: Generación de Código (IMP 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Generación de código	0	1	0 = No 1 = Si	Bolean
Grado de completitud del código generado	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Calidad del código generado	0	4	4 = Excelente 3 = Bueno 2 = Deficiente 1 = Malo 0 = No está definido	Escala de liker
Porcentaje de lenguajes en que genera código	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Ejecutables para distintas plataformas	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 5				

Tabla N° 95. Propuesta de métricas de la Subcaracterística Atómica Generación de Código del MC-HDS-MDA

Fuente: Elaboración Propia.

La métrica *Cantidad de lenguajes en que genera código* era de tipo Entera, es decir, solo había que indicar si soportaba 3, 6 o 10 lenguajes de programación. Sin embargo luego de analizar cuidadosamente los valores arrojados para el modelo de calidad, se notó que ese valor no es significativo, mientras que decir, si genera código para el 100% de los lenguajes de programación que existan para el momento de la evaluación, si resulta relevante para el modelo propuesto. Tomando en cuenta esta observación el nombre de la métrica cambia a **Porcentaje de lenguajes en que genera código**.

Subcaracterística Atómica: Generación de Esquemas de Bases de Datos (IMP 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Porcentaje de manejadores de bases de datos que maneja.	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Generación de los esquemas de bases de datos a partir de la especificación de los requisitos	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción de los esquemas de bases de datos generados por la herramienta.	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Porcentaje de manejadores de bases de datos en que genera esquemas	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Total de Métricas = 4				

Tabla N° 96. Propuesta de métricas de la Subcaracterística Atómica Generación de Esquemas de Bases de Datos del MC-HDS-MDA

Fuente: Elaboración Propia.

La métrica *Número de manejadores de bases de datos en que genera esquemas* era de tipo Entera, es decir, solo había que indicar si soportaba 3, 6 o 10 manejadores de bases de datos. Sin embargo luego de analizar cuidadosamente los valores arrojados para el modelo de calidad, se notó que ese

valor no es significativo, mientras que decir, si genera esquemas para el 100% de los manejadores de bases de datos que existan para el momento de la evaluación, si resulta relevante para el modelo propuesto. Tomando en cuenta esta observación el nombre de la métrica cambia a **Porcentaje de manejadores de bases de datos en que genera esquemas**. Por otro lado, se incorpora la métrica **Porcentaje de manejadores de bases de datos que maneja**, para el caso de una HDS que soporte no solo la generación de esquemas sino que manipule o se integre con manejadores de bases de datos.

Subcaracterística Atómica: Generación de Pantallas (IMP 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Genera pantallas a partir de alguna especificación de requisitos	0	1	0 = No 1 = Si	Boolean
Grado de satisfacción de la generación de pantallas	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Facilidad con que se generan las pantallas	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 97. Propuesta de métricas de la Subcaracterística Atómica Generación de Pantallas del MC-HDS-MDA

Fuente: Elaboración Propia.

Se eliminan las métricas *Generador de pantallas* y *Bondad de diseño visual* por estar contenidas y mejor formuladas en el resto de las métricas de la Subcaracterística atómica **Generación de pantallas**.

Subcaracterística Atómica: Generación de Reportes (IMP 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite generar reportes	0	1	0 = No 1 = Si	Boolean
Permite definir reportes con formatos del usuario	0	1	0 = No 1 = Si	Boolean
Calidad de los reportes que genera	0	4	4 = Excelente 3 = Bueno 2 = Deficiente 1 = Malo 0 = No está definido	Escala de liker
Grado de completitud de los reportes	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Total de Métricas = 4				

Tabla N° 98. Propuesta de métricas de la Subcaracterística Atómica Generación de Reportes del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Compilación (IMP 5)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite compilar código fuente	0	1	0 = No 1 = Si	Boolean
Porcentaje de lenguajes en que compila código fuente	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Grado de satisfacción del proceso de compilación realizado por la	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho	Escala de liker

herramienta			0 = No aplica	
Emite reportes acerca de la compilación	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 4				

Tabla N° 99. Propuesta de métricas de la Subcaracterística Atómica Compilación del MC-HDS-MDA
Fuente: Elaboración Propia.

La métrica *Cantidad de lenguajes en que compila código fuente* era de tipo Entera, es decir, solo había que indicar si soportaba 3, 6 o 10 lenguajes de programación. Sin embargo luego de analizar cuidadosamente los valores arrojados para el modelo de calidad, se notó que ese valor no es significativo, mientras que decir, si genera código para el 100% de los lenguajes de programación que existan para el momento de la evaluación, si resulta relevante para el modelo propuesto. Tomando en cuenta esta observación el nombre de la métrica cambia a **Porcentaje de lenguajes en que compila código fuente**.

Subcaracterística Atómica: Edición dirigida por la sintaxis (IMP 6)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Provee soporte de sintaxis para la edición del código fuente. (en varios lenguajes de programación).	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción del soporte de sintaxis de la herramienta	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Porcentaje de lenguajes en que soporta la sintaxis del código fuente	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Total de Métricas = 3				

Tabla N° 100. Propuesta de métricas de la Subcaracterística Atómica Edición Dirigida por la Sintaxis del MC-HDS-MDA
Fuente: Elaboración Propia.

La métrica *Cantidad de lenguajes en que soporta la sintaxis del código fuente* era de tipo Entera, es decir, solo había que indicar si soportaba 3, 6 o 10 lenguajes de programación. Sin embargo luego de analizar cuidadosamente los valores arrojados para el modelo de calidad, se notó que ese valor no es significativo, mientras que decir, si genera código para el 100% de los lenguajes de programación que existan para el momento de la evaluación, si resulta relevante para el modelo propuesto. Tomando en cuenta esta observación el nombre de la métrica cambia a **Porcentaje de lenguajes en que soporta la sintaxis del código fuente**.

Subcaracterística Atómica: Depurador (IMP 7)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
La herramienta identifica y aísla los errores detectados por el compilador	0	1	0 = No 1 = Si	Bolean

Permite definir policías en el código a depurar	0	1	0 = No 1 = Si	Bolean
Permite seleccionar variables para observar los valores que toma	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción de la identificación de los errores	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 4				

Tabla N° 101. Propuesta de métricas de la Subcaracterística Atómica Depurador del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística: Pruebas (PRU)

Subcaracterística Atómica: Prueba de Técnicas de correctitud (PRU 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite constatar si se cumplen las restricciones (precondiciones y postcondiciones del software)	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 1				

Tabla N° 102. Propuesta de métricas de la Subcaracterística Atómica Prueba de técnicas de correctitud del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Análisis de Fallas (PRU 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Frecuencia de fallas por tiempo que presenta el software	0	4	4 = Siempre 3 = Casi siempre 2 = Algunas veces. 1 = Pocas veces 0 = Nunca	Escala de liker
Rapidez de recuperación después de haberse presentado la falla	0	4	4 = Rápido 3 = Med. rápido 2 = Lento 1 = Muy lento 0 = No aplica	Escala de liker
Lleva un registro de las fallas presentadas	0	1	0 = No 1 = Si	Bolean
Lleva un registro de la causa que genera la falla	0	1	0 = No 1 = Si	Bolean
Emite reportes de las fallas presentadas	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 5				

Tabla N° 103. Propuesta de métricas de la Subcaracterística Atómica Análisis de Fallas del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Análisis de Defectos (PRU 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Reporta el número de defectos encontrados	0	1	0 = No 1 = Si	Bolean
Lleva un registro de los defectos encontrados	0	1	0 = No 1 = Si	Bolean
Lleva un registro de las consecuencias de las fallas	0	1	0 = No 1 = Si	Bolean
Lleva un registro de las consecuencias de los defectos	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 4				

Tabla N° 104. Propuesta de métricas de la Subcaracterística Atómica Análisis de Defectos del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Casos de prueba y espera de los datos de entrada (PRU 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Facilidad para ingresar casos de prueba	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Genera datos de entrada de acuerdo con los casos de prueba	0	1	0 = No 1 = Si	Boolean
Facilidad para comparar con los resultados esperados	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Permite llevar un registro de los casos de prueba y sus resultados	0	1	0 = No 1 = Si	Boolean
Grado de satisfacción de la comparación con los resultados esperados	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 5				

Tabla N° 105. Propuesta de métricas de la Subcaracterística Atómica Casos de prueba y espera de los resultados de entrada del MC-HDS-MDA

Fuente: Elaboración Propia.

Como puede observarse en la Tabla N° 105, se ha agregado una nueva métrica **Genera datos de entrada de acuerdo con los casos de prueba**, por considerarla importante como ayuda al encargado de realizar las pruebas de software.

Subcaracterística Atómica: Casos de prueba y generación de resultados esperados (PRU 5)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Genera automáticamente casos de prueba	0	1	0 = No 1 = Si	Boolean
Genera automáticamente los resultados esperados	0	1	0 = No 1 = Si	Boolean
Grado de satisfacción de la generación de los casos de prueba	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Grado de satisfacción de los resultados esperados	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 4				

Tabla N° 106. Propuesta de métricas de la Subcaracterística Atómica Casos de prueba y generación de resultados esperados del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Prueba de trazabilidad (PRU 6)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Verifica consistencia entre las pruebas y los datos de prueba	0	1	0 = No 1 = Si	Boolean
Grado de satisfacción de la verificación de consistencia entre las pruebas y los datos de prueba	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 2				

Tabla N° 107. Propuesta de métricas de la Subcaracterística Atómica Prueba de trazabilidad del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Instrumentación del Código fuente (PRU 7)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Ejecuta el código fuente	0	1	0 = No 1 = Si	Bolean
Grado de completitud de la ejecución del código fuente	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Grado de satisfacción de la ejecución del código fuente	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 108. Propuesta de métricas de la Subcaracterística Atómica Instrumentación del código fuente del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Captura los datos de entrada y los reutiliza (PRU 8)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Guarda datos que hayan sido editados para casos de prueba	0	1	0 = No 1 = Si	Bolean
Permite reutilizar los datos de prueba	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 2				

Tabla N° 109. Propuesta de métricas de la Subcaracterística Atómica Captura los datos de entrada y los reutiliza del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Manejo de pruebas (PRU 9)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite repetir casos de prueba previamente ejecutados	0	1	0 = No 1 = Si	Bolean
Permite realizar búsqueda de pruebas	0	1	0 = No 1 = Si	Bolean
Permite hacer planificación de pruebas	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 3				

Tabla N° 110. Propuesta de métricas de la Subcaracterística Atómica Manejo de pruebas del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Análisis en tiempo de corrida (PRU 10)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Genera reportes del desempeño del programa durante la ejecución del mismo	0	1	0 = No 1 = Si	Bolean
Permite ingresar indicadores de desempeño de ejecución del programa	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción de los reportes de desempeño generados	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 111. Propuesta de métricas de la Subcaracterística Atómica Análisis en tiempo de corrida del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Análisis de fiabilidad (PRU 11)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite definir indicadores de confiabilidad	0	1	0 = No 1 = Si	Bolean
Genera reportes de los indicadores de confiabilidad	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 2				

Tabla N° 112. Propuesta de métricas de la Subcaracterística Atómica Análisis de fiabilidad del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Análisis de alcance de pruebas (PRU 12)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Genera reportes del alcance de las pruebas	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción del alcance de las pruebas	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 2				

Tabla N° 113. Propuesta de métricas de la Subcaracterística Atómica Análisis de alcance de pruebas del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Gerencia del proceso de pruebas (PRU 13)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite formular un plan para la ejecución de las pruebas	0	1	0 = No 1 = Si	Bolean
Definición y modificación de prioridades en la ejecución de las pruebas	0	1	0 = No 1 = Si	Bolean
Definición y modificación de actividades de las pruebas	0	1	0 = No 1 = Si	Bolean
Definición y modificación de tiempos en los cuales se realizaran las pruebas	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción de la gerencia del proceso de prueba	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 5				

Tabla N° 114. Propuesta de métricas de la Subcaracterística Atómica Gerencia del proceso de pruebas del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Prueba de regresión (PRU 14)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Compara los resultados de diferentes casos de prueba	0	1	0 = No 1 = Si	Bolean
Genera reportes de las pruebas de regresión	0	1	0 = No 1 = Si	Bolean
Grado de completitud de los reportes de las pruebas de regresión	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Grado de satisfacción con los reportes de las pruebas de regresión	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker

Total de Métricas = 4

Tabla N° 115. Propuesta de métricas de la Subcaracterística Atómica Prueba de regresión del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Chequeo automático de resultados (PRU 15)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Chequeo o comparación de los resultados de las pruebas con los resultados esperados	0	1	0 = No 1 = Si	Bolean
Generación de reportes del chequeo de los resultados de las pruebas	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción del chequeo de los resultados de las pruebas	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Grado de completitud de los reportes del chequeo de los resultados de las pruebas	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Total de Métricas = 4				

Tabla N° 116. Propuesta de métricas de la Subcaracterística Atómica Chequeo automático de resultados del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Análisis de pruebas estadísticas (PRU 16)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite definir pruebas estadísticas de los resultados	0	1	0 = No 1 = Si	Bolean
Genera reportes estadísticos de los resultados de las pruebas	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción de las pruebas estadísticas de los resultados	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 117. Propuesta de métricas de la Subcaracterística Atómica Análisis de pruebas estadísticas del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Simulación en ambiente de operaciones (PRU 17)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Genera simulaciones en el ambiente real de operación	0	1	0 = No 1 = Si	Bolean
Permite definir las pautas de la simulación	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción de las simulaciones en ambiente real	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 118. Propuesta de métricas de la Subcaracterística Atómica Simulación en ambiente de operaciones del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Pruebas de integración (PRU 18)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite registrar las actividades de	0	1	0 = No 1 = Si	Bolean

integración de software				
Registra los tiempos de las actividades de integración	0	1	0 = No 1 = Si	Bolean
Registra la secuencia de las actividades de integración	0	1	0 = No 1 = Si	Bolean
Emite reportes de las pruebas de integración	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción de los reportes de las pruebas de integración.	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Grado de completitud de los reportes de las pruebas de integración	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Total de Métricas = 6				

Tabla N° 119. Propuesta de métricas de la Subcaracterística Atómica Pruebas de integración del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística: Mantenimiento (MAN)

Subcaracterística Atómica: Entendimiento del problema (MAN 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Emite reportes de fallas	0	1	0 = No 1 = Si	Bolean
Emite reportes de defectos	0	1	0 = No 1 = Si	Bolean
Restringe las operaciones cuando se presenta una falla	0	1	0 = No 1 = Si	Bolean
Tiene la capacidad de determinar si los problemas han sido resueltos	0	1	0 = No 1 = Si	Bolean
Tiene la capacidad de aprender de los defectos y fallas encontradas	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 5				

Tabla N° 120. Propuesta de métricas de la Subcaracterística Atómica Entendimiento del problema del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Localización (MAN 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Identifica el problema que requiera modificación.	0	1	0 = No 1 = Si	Bolean
Identifica la ubicación del problema	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción de la localización de errores	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 121. Propuesta de métricas de la Subcaracterística Atómica Localización del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Análisis de impacto (MAN 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Identifica las consecuencias de hacer cambios	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción de la	0	4	4 = Muy Satisfecho 3 = Satisfecho	Escala de

identificación de las consecuencias de hacer cambios			2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	liker
Difunde los cambios	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción con los mecanismos de difusión utilizados por la herramienta	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 4				

Tabla N° 122. Propuesta de métricas de la Subcaracterística Atómica Análisis de impacto del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Ingeniería de reverso de datos (MAN 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite hacer ingeniería de reverso de datos	0	1	0 = No 1 = Si	Bolean
Compleitud del proceso de ingeniería de reverso de datos	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Calidad del producto de la Ingeniería de reverso de datos	0	4	4 = Excelente 3 = Bueno 2 = Deficiente 1 = Malo 0 = No está definido	Escala de liker
Porcentaje de lenguajes soportados por la ingeniería de reverso de datos	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Total de Métricas = 4				

Tabla N° 123. Propuesta de métricas de la Subcaracterística Atómica Ingeniería de reverso de datos del MC-HDS-MDA
Fuente: Elaboración Propia.

La métrica *Cantidad de lenguajes soportados por la ingeniería de reverso de datos* era de tipo Entera, es decir, solo había que indicar si soportaba 3, 6 o 10 lenguajes de modelado de datos. Sin embargo luego de analizar cuidadosamente los valores arrojados para el modelo de calidad, se notó que ese valor no es significativo, mientras que decir, si genera código para el 100% de los lenguajes que existan para el momento de la evaluación, si resulta relevante para el modelo propuesto. Tomando en cuenta esta observación el nombre de la métrica cambia a **Porcentaje de lenguajes soportados por la ingeniería de reverso de datos**.

Subcaracterística Atómica: Ingeniería de reverso de procesos (MAN 5)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite hacer ingeniería de reverso de procesos	0	1	0 = No 1 = Si	Bolean
Compleitud del proceso de ingeniería de reverso de procesos	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Calidad del producto de la Ingeniería de reverso de procesos	0	4	4 = Excelente 3 = Bueno 2 = Deficiente 1 = Malo 0 = No está definido	Escala de liker
Cantidad de lenguajes soportados por la ingeniería de reverso de procesos	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Total de Métricas = 4				

Tabla N° 124. Propuesta de métricas de la Subcaracterística Atómica Ingeniería de reversa de procesos del MC-HDS-MDA
Fuente: Elaboración Propia.

La métrica *Cantidad de lenguajes soportados por la ingeniería de reverso de procesos*, al igual que las métricas que anteriormente arrojaban un valor entero positivo, también cambia a **Porcentaje de lenguajes soportados por la ingeniería de reverso de procesos**.

Subcaracterística Atómica: Reestructuración del código fuente (MAN 6)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Modifica el formato y estructura del código fuente en uno o más lenguajes. (reducir el tamaño del código, reducir el tiempo de ejecución).	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción de la modificación y reestructuración del código fuente	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Porcentaje de lenguajes en que reestructura el código fuente	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Total de Métricas = 3				

Tabla N° 125. Propuesta de métricas de la Subcaracterística Atómica Reestructuración del código fuente del MC-HDS-MDA
Fuente: Elaboración Propia.

La métrica *Número de lenguajes en que reestructura el código fuente*, también cambia a **Porcentaje de lenguajes en que reestructura el código fuente**, y se valora a través de una escala de liker.

Subcaracterística Atómica: Traducción del código fuente (MAN 7)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite traducir código fuente de un lenguaje a otro	0	1	0 = No 1 = Si	Bolean
Porcentaje de lenguajes en que traduce el código fuente	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Grado de satisfacción de la traducción del código fuente	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 126. Propuesta de métricas de la Subcaracterística Atómica Traducción del Código fuente del MC-HDS-MDA
Fuente: Elaboración Propia.

La métrica *Número de lenguajes en que traduce el código fuente*, también cambia a **Porcentaje de lenguajes en que traduce el código fuente**, y se valora a través de una escala de liker.

Subcaracterística: Arquitectura Dirigida por Modelos (MDA)

Subcaracterística Atómica: Soporte a MIC (MDA 1)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Soporta el MIC	¿La HDS soporta la construcción del Modelo independiente de la computación?	0	1	No = 0 Si = 1	Boolean
Grado de satisfacción del soporte a MIC	¿Qué tan satisfecho esta el diseñador con la construcción del MIC realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Transforma MIC a MIP	¿La HDS soporta la transformación de un MIC a un MIP?	0	1	No = 0 Si = 1	Boolean
Grado de satisfacción de la transformación de MIC a MIP	¿Qué tan satisfecho esta el diseñador con la transformación de MIC a MIP realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Transforma Texto a MIC	¿La HDS soporta la transformación de Texto a MIC?	0	1	No = 0 Si = 1	Boolean
Grado de satisfacción de la transformación de Texto a MIC	¿Qué tan satisfecho esta el diseñador con la transformación de Texto a MIC realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Transforma MIC a Texto	¿La HDS soporta la transformación de MIC a Texto?	0	1	No = 0 Si = 1	Boolean
Grado de satisfacción de la transformación de MIC a Texto	¿Qué tan satisfecho esta el diseñador con la transformación de MIC a Texto realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Permite modificaciones al MIC	¿La HDS permite que el diseñador realice modificaciones al MIC?	0	1	No = 0 Si = 1	Boolean
Transforma MIC a otros MIC	¿La HDS soporta la transformación de MIC a otros MIC?	0	1	No = 0 Si = 1	Boolean
Grado de satisfacción de la transformación de MIC a otros MIC	¿Qué tan satisfecho esta el diseñador con la transformación de MIC a otros MIC realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 11					

Tabla N° 127. Propuesta de métricas de la Subcaracterística Atómica Soporte a MIC del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Soporte a MIP (MDA 2)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Soporta el MIP	¿La HDS soporta la construcción del Modelo independiente de la plataforma?	0	1	No = 0 Si = 1	Boolean
Grado de satisfacción del soporte a MIP	¿Qué tan satisfecho esta el diseñador con la construcción del MIP realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Transforma MIP a MIC	¿La HDS soporta la transformación de un MIP a un MIC?	0	1	No = 0 Si = 1	Boolean
Grado de satisfacción de la transformación de MIP a MIC	¿Qué tan satisfecho esta el diseñador con la transformación de MIP a MIC realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Permite modificaciones al MIP	¿La HDS permite que el diseñador realice modificaciones al MIP?	0	1	No = 0 Si = 1	Boolean
Transforma MIP a	¿La HDS soporta la transformación de un	0	1	No = 0	Boolean

MEP	MIP a un MEP?			Si = 1	
Grado de satisfacción de la transformación de MIP a MEP	¿Qué tan satisfecho esta el diseñador con la transformación de MIP a MEP realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Permite generar varios MEP a partir de un MIP	¿La HDS permite generar varios MEP a partir de un MIP?	0	1	No = 0 Si = 1	Bolean
Transforma MIP a otros MIP	¿La HDS soporta la transformación de un MIP a otros MIP?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción de la transformación de MIP a otros MIP	¿Qué tan satisfecho esta el diseñador con la transformación de MIP a otros MIP realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 10					

Tabla N° 128. Propuesta de métricas de la Subcaracterística Atómica Soporte a MIP del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Soporte a MEP (MDA 3)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Soporta el MEP	¿La HDS soporta la construcción del Modelo Específico de la plataforma?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción del soporte a MEP	¿Qué tan satisfecho esta el diseñador con la construcción del MEP realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Transforma MEP a Código	¿La HDS soporta la transformación de un MEP a Código?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción de la transformación de MEP a Código	¿Qué tan satisfecho esta el diseñador con la transformación de MEP a Código realizada por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Permite modificaciones al MEP	¿La HDS permite que el diseñador realice modificaciones al MEP?	0	1	No = 0 Si = 1	Bolean
Transforma MEP a MIP	¿La HDS soporta la transformación de un MEP a MIP?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción de la transformación de MEP a MIP	¿Qué tan satisfecho esta el diseñador con la transformación de MEP a MIP realizada por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Transforma MEP a otros MEP	¿La HDS soporta la transformación de un MEP a otros MEP?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción de la transformación de MEP a otros MEP	¿Qué tan satisfecho esta el diseñador con la transformación de MEP a otros MEP realizada por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 9					

Tabla N° 129. Propuesta de métricas de la Subcaracterística Atómica Soporte a MEP del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Soporte a Código (MDA 4)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Soporta el Código	¿La HDS soporta la construcción del código?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción del	¿Qué tan satisfecho esta el diseñador con la construcción del Código realizado por la	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho	Escala de liker

soporte a Código	HDS?			0 = No aplica	
Transforma Código a MEP	¿La HDS soporta la transformación de Código a MEP?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción de la transformación de Código a MEP	¿Qué tan satisfecho esta el diseñador con la transformación de Código a MEP realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Permite modificaciones al Código	¿La HDS permite que el diseñador realice modificaciones al MIC?	0	1	No = 0 Si = 1	Bolean
Total de Métricas = 5					

Tabla N° 130. Propuesta de métricas de la Subcaracterística Atómica Soporte a código del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Proceso de Transformación (MDA 5)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Transformación basada en marcas	¿La HDS realiza las transformaciones basadas en marcas?	0	1	No = 0 Si = 1	Bolean
Transformación basada en meta-modelos	¿La HDS realiza las transformaciones basadas en meta-modelos?	0	1	No = 0 Si = 1	Bolean
Transformación basada en modelos	¿La HDS realiza las transformaciones basadas en modelos?	0	1	No = 0 Si = 1	Bolean
Transformación basada en patrones	¿La HDS realiza las transformaciones basadas en patrones?	0	1	No = 0 Si = 1	Bolean
Transformación basada en perfiles UML	¿La HDS realiza las transformaciones basadas en perfiles UML?	0	1	No = 0 Si = 1	Bolean
Registro de transformaciones	¿La HDS registra las transformaciones realizadas?	0	1	No = 0 Si = 1	Bolean
Trazabilidad de transformaciones	¿La HDS verifica la trazabilidad de las transformaciones?	0	1	No = 0 Si = 1	Bolean
Total de Métricas = 7					

Tabla N° 131. Propuesta de métricas de la Subcaracterística Atómica Proceso de transformación del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Soporte a OCL (MDA 6)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Soporte para XMI-XML	¿La HDS soporta XMI-XML?	0	1	No = 0 Si = 1	Bolean
Soporte Query-View-Transformation	¿La HDS soporta QVT?	0	1	No = 0 Si = 1	Bolean
Soporte OCL	¿La HDS soporta OCL?	0	1	No = 0 Si = 1	Bolean
Total de Métricas = 3					

Tabla N° 132. Propuesta de métricas de la Subcaracterística Atómica Soporte a OCL del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Soporte a Perfiles (MDA 7)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Soporte a Perfiles	¿La HDS soporta los Perfiles UML?	0	1	No = 0	Bolean

UML				Si = 1	
Total de Métricas = 1					

Tabla N° 133. Propuesta de métricas de la Subcaracterística Atómica Soporte a Perfiles del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Soporte a Metamodelo – MOF y UML (MDA 8)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Soporte a UML	¿La HDS soporta el UML?	0	1	No = 0 Si = 1	Bolean
Soporte a MOF	¿La HDS soporta el MOF?	0	1	No = 0 Si = 1	Bolean
Total de Métricas = 2					

Tabla N° 134. Propuesta de métricas de la Subcaracterística Atómica Soporte a Metamodelo – MOF y UML del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Soporte a MIC – MIP – MEP - Código (MDA 9)					
Nombre	Definición operacional	Valor Mín	Valor Máx	Formulación	Tipo de Métrica
Transforma MIC a MIP a MEP a Código	¿La HDS soporta la transformación de MIC a MIP a MEP a Código?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción de la Transformación de MIC a MIP a MEP a Código	¿Qué tan satisfecho esta el diseñador con la transformación de MIC a MIP a MEP a Código realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Transforma Código a MEP a MIP a MIC	¿La HDS soporta la transformación de Código a MEP a MIP a MIC?	0	1	No = 0 Si = 1	Bolean
Grado de satisfacción de la transformación de Código a MEP a MIP a MIC	¿Qué tan satisfecho esta el diseñador con la transformación de Código a MEP a MIP a MIC realizado por la HDS?	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 4					

Tabla N° 135. Propuesta de métricas de la Subcaracterística Atómica Soporte a MIC – MIP – MEP – Código del MC-HDS-MDA
Fuente: Elaboración Propia.

CICLO DE VIDA. PROCESOS DE APOYO

Subcaracterística: Gerencia (GER)

Subcaracterística Atómica: Estimación de costos y de planificación (GER 1)				
Nombre	Valor Mín	Valor Max	Formulación	Tipo de Métrica
Estimación de costos del proyecto	0	1	0 = No 1 = Si	Bolean
Planificación del proyecto	0	1	0 = No 1 = Si	Bolean
Porcentaje de métodos que posee para la estimación de costos y planificación	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker

Permite ingresar datos referentes a la estimación de costos y planificación	0	1	0 = No 1 = Si	Bolean
Permite modificar los datos referentes a la estimación de costos y la planificación	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 5				

Tabla N° 136. Propuesta de métricas de la Subcaracterística Atómica Estimación de costos y de planificación del MC-HDS-MDA

Fuente: Elaboración Propia.

La métrica *Cantidad de métodos que posee para la estimación de costos y planificación*, también cambia a **Porcentaje de métodos que posee para la estimación de costos y planificación**, y se valora a través de una escala de liker.

Subcaracterística Atómica: Planificación (GER 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Grado de satisfacción de la planificación hecha por la herramienta	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Registro de entregables por tareas	0	1	0 = No 1 = Si	Bolean
Registro de tareas o actividades	0	1	0 = No 1 = Si	Bolean
Registro de tiempos por tareas	0	1	0 = No 1 = Si	Bolean
Registro de recursos asignados por tareas	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 5				

Tabla N° 137. Propuesta de métricas de la Subcaracterística Atómica Planificación del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Seguimiento del proyecto (GER 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite hacer seguimiento al proyecto	0	1	0 = No 1 = Si	Bolean
Permite registrar los recursos consumidos	0	1	0 = No 1 = Si	Bolean
Genera documentos para el seguimiento del proyecto	0	1	0 = No 1 = Si	Bolean
Indica el numero de líneas de código desarrolladas	0	1	0 = No 1 = Si	Bolean
Indica el numero de casos de prueba ejecutados	0	1	0 = No 1 = Si	Bolean
Detecta defectos	0	1	0 = No 1 = Si	Bolean
Permite determinar el estado del proyecto	0	1	0 = No 1 = Si	Bolean
Corrige defectos	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción del seguimiento del proyecto	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 9				

Tabla N° 138. Propuesta de métricas de la Subcaracterística Atómica Seguimiento del proyecto del MC-HDS-MDA

Fuente: Elaboración Propia.

En esta subcaracterística atómica se han eliminado las métricas *Numero de documentos generados para el seguimiento del proyecto, líneas de código desarrolladas, numero de casos de prueba concluidos, número de defectos detectados, número de defectos corregidos*, debido a que no aportan información relevante para seleccionar o no a una herramienta de desarrollo de software.

Subcaracterística Atómica: Análisis y reportes del estado del proyecto (GER 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Emite reportes del seguimiento del proyecto, en formato definido por el usuario	0	1	0 = No 1 = Si	Bolean
Emite reportes de la estimación de costos del proyecto	0	1	0 = No 1 = Si	Bolean
Emite reportes de los costos reales del proyecto	0	1	0 = No 1 = Si	Bolean
Emite reportes de la planificación del proyecto	0	1	0 = No 1 = Si	Bolean
Emite reportes de la gerencia del proyecto	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción de la generación de reportes del seguimiento del proyecto.	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 6				

Tabla N° 139. Propuesta de métricas de la Subcaracterística Atómica Análisis y reportes del estado del proyecto del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Gerencia del proceso (GER 5)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Definición de entradas para la gerencia del proceso	0	1	0 = No 1 = Si	Bolean
Definición de recursos	0	1	0 = No 1 = Si	Bolean
Definición de fechas límites	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 3				

Tabla N° 140. Propuesta de métricas de la Subcaracterística Atómica Gerencia del proceso del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística: Documentación (DOC)

Subcaracterística Atómica: Edición de texto (DOC 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Facilidad de documentación	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Facilidad para editar texto	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Grado de satisfacción de la edición de texto	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Discrimina instrucciones de variables	0	1	0 = No 1 = Si	Bolean
Permite la edición automática de	0	1	0 = No 1 = Si	Bolean

texto				
Total de Métricas = 5				

Tabla N° 141. Propuesta de métricas de la Subcaracterística Atómica Edición de texto del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Edición gráfica (DOC 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Grado de satisfacción de la edición gráfica	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Calidad de la edición gráfica	0	4	4 = Excelente 3 = Bueno 2 = Deficiente 1 = Malo 0 = No está definido	Escala de liker
Grado de facilidad de edición gráfica	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Grado de completitud de la caja de herramientas para editar datos gráficamente	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Total de Métricas = 4				

Tabla N° 142. Propuesta de métricas de la Subcaracterística Atómica Edición gráfica del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Edición basada en formas (DOC 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite la edición a través de formas	0	1	0 = No 1 = Si	Boolean
Permite la edición a través de formas definidas por el usuario	0	1	0 = No 1 = Si	Boolean
Grado de satisfacción de la edición basada en formas	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 143. Propuesta de métricas de la Subcaracterística Atómica Edición basada en formas del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Publicidad (DOC 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Cuenta con mecanismos tales como e-mail, groupware, para difundir la documentación	0	1	0 = No 1 = Si	Boolean
Grado de satisfacción de la difusión de la documentación	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 2				

Tabla N° 144. Propuesta de métricas de la Subcaracterística Atómica Publicidad del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Soporte de hipertexto (DOC 5)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
La herramienta soporta la edición de hipertexto	0	1	0 = No 1 = Si	Boolean
La herramienta soporta todas las funciones de hipertexto	0	1	0 = No 1 = Si	Boolean

Grado de satisfacción del soporte de hipertexto.	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 145. Propuesta de métricas de la Subcaracterística Atómica Soporte de hipertexto del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Manejo de variantes (DOC 6)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Facilidad de cambio en producto terminado	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Permite identificar las diferencias entre las versiones de un documento	0	1	0 = No 1 = Si	Boolean
Permite llevar un registro para cada versión	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 3				

Tabla N° 146. Propuesta de métricas de la Subcaracterística Atómica Manejo de variantes del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Extracción automática de datos y generación de documentos (DOC 7)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite la recuperación de las especificaciones de datos textuales y gráficos.	0	1	0 = No 1 = Si	Boolean
Genera documentos de las especificaciones de los datos	0	1	0 = No 1 = Si	Boolean
Grado de satisfacción de la generación de documentos	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Permite personalizar los documentos	0	1	0 = No 1 = Si	Boolean
Grado de completitud de la recuperación de los datos	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Total de Métricas = 5				

Tabla N° 147. Propuesta de métricas de la Subcaracterística Atómica Extracción automática de datos y generación de documentos del MC-HDS-MDA

Fuente: Elaboración Propia.

Las métricas *Número de especificaciones de datos textuales que recupera* y *Número de especificaciones de datos gráficos que recupera*, quedan eliminadas, en vista de que no aportan nada al modelo de calidad.

Subcaracterística: Verificación (VER)

Subcaracterística Atómica: Análisis de especificación de trazabilidad (VER 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica

Facilidad para detectar inconsistencia	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Verifica consistencia entre los modelos	0	1	0 = No 1 = Si	Boolean
Grado de satisfacción de la verificación de consistencia realizada	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Permite trazar los cambios de un modelo a otro	0	1	0 = No 1 = Si	Boolean
Facilidad para trazar los cambios de un modelo a otro	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Total de Métricas = 5				

Tabla N° 148. Propuesta de métricas de la Subcaracterística Atómica Análisis de especificación de trazabilidad del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Análisis de especificación (VER 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Emite reportes de los modelos de análisis y sus elementos	0	1	0 = No 1 = Si	Boolean
Emite reportes de los modelos de diseño y sus elementos	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 2				

Tabla N° 149. Propuesta de métricas de la Subcaracterística Atómica Análisis de especificación del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Análisis de código fuente (VER 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite la definición de métricas de análisis de código fuente	0	1	0 = No 1 = Si	Boolean
Emite reportes sobre las métricas de análisis de código fuente	0	1	0 = No 1 = Si	Boolean
Grado de satisfacción de los reportes sobre métricas para analizar el código fuente	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 150. Propuesta de métricas de la Subcaracterística Atómica Análisis del código fuente del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística: Aseguramiento de la Calidad (ASE)

Subcaracterística Atómica: Gerencia de la calidad de los datos (ASE 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Proporciona planes de auditoria y revisiones de resultados o productos	0	1	0 = No 1 = Si	Boolean
Proporciona revisiones de resultados o productos	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 2				

Tabla N° 151. Propuesta de métricas de la Subcaracterística Atómica Gerencia de la calidad de los datos del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Gerencia de riesgos (ASE 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite definir y modificar los riesgos del proyecto	0	1	0 = No 1 = Si	Bolean
Permite elaborar listas de chequeo de riesgos o hacer control de riesgos	0	1	0 = No 1 = Si	Bolean
Permite clasificar los riesgos	0	1	0 = No 1 = Si	Bolean
Emite reportes de los riesgos del proyecto	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción de la gerencia de los riesgos	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 5				

Tabla N° 152. Propuesta de métricas de la Subcaracterística Atómica Gerencia de riesgos del MC-HDS-MDA
Fuente: Elaboración Propia.

La métrica *Permite elaborar listas de chequeo de riesgos* se le ha agregado lo siguiente: “o hacer control de riesgos”, quedando definida **Permite elaborar listas de chequeo de riesgos o hacer control de riesgos**, de esta forma es más claro para el evaluador saber que se trata del control o el monitoreo de los riesgos.

Subcaracterística: Gerencia de la Configuración (CONF)

Subcaracterística Atómica: Control de acceso (CONF 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite establecer restricciones para acceder a los elementos de datos	0	1	0 = No 1 = Si	Bolean
Permite llevar registros de las modificaciones realizadas a los elementos de datos	0	1	0 = No 1 = Si	Bolean
Permite diferenciar las autoridades	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción del control sobre el acceso que posee la herramienta	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 4				

Tabla N° 153. Propuesta de métricas de la Subcaracterística Atómica Control de acceso del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Búsqueda de modificaciones (CONF 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite llevar registros de las modificaciones realizadas al sistema en general	0	1	0 = No 1 = Si	Bolean
Permite realizar búsquedas de las modificaciones de acuerdo a	0	1	0 = No 1 = Si	Bolean

diferentes criterios				
Grado de satisfacción de la búsqueda de modificaciones	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 154. Propuesta de métricas de la Subcaracterística Atómica Búsqueda de modificaciones del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Definición y gerencia de versiones múltiples (CONF 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Facilidad de integración de versiones	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Permite el registro de versiones	0	1	0 = No 1 = Si	Bolean
Facilidad de búsqueda de las versiones	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 155. Propuesta de métricas de la Subcaracterística Atómica Definición y gerencia de versiones múltiples del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Estatus de configuración (CONF 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Genera reportes donde se resuma la información de las configuraciones gerenciadas en una sesión de trabajo	0	1	0 = No 1 = Si	Bolean
Grado de completitud de los reportes de configuración generados	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Grado de satisfacción de los reportes de configuración generados	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 156. Propuesta de métricas de la Subcaracterística Atómica Estatus de configuración del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Generación de versiones (CONF 5)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite la creación de versiones de un producto a partir de otro	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción de la generación de versiones	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Registra las versiones creadas	0	1	0 = No 1 = Si	Bolean
Facilidad de búsqueda de versiones	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Total de Métricas = 4				

Tabla N° 157. Propuesta de métricas de la Subcaracterística Atómica Generación de versiones del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Capacidad de archivo (CONF 6)				
---	--	--	--	--

Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Graba automáticamente los productos generados	0	1	0 = No 1 = Si	Boolean
Facilidad de búsqueda de los productos generados	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Total de Métricas = 2				

Tabla N° 158. Propuesta de métricas de la Subcaracterística Atómica Capacidad de arquito del MC-HDS-MDA

Fuente: Elaboración Propia.

CALIDAD

Subcaracterística: Funcionalidad (FUN)

Subcaracterística Atómica: Seguridad (FUN 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Mecanismo de manejo de niveles de seguridad	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 1				

Tabla N° 159. Propuesta de métricas de la Subcaracterística Atómica Seguridad del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Exactitud (FUN 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Son claros los mensajes emitidos por la HDS	0	1	0 = No 1 = Si	Boolean
Localización rápida de opciones	0	1	0 = No 1 = Si	Boolean
Es versátil la navegación con la HDS.	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 3				

Tabla N° 160. Propuesta de métricas de la Subcaracterística Atómica Exactitud del MC-HDS-MDA

Fuente: Elaboración Propia.

Tal como se muestra en la Tabla N° 160, se cambia el nombre de la métrica *Claridad de los mensajes*, por **Son claros los mensajes emitidos por la HDS**, y la métrica *Versatilidad de la navegación* se cambia por, **Es versátil la navegación con la HDS**, por considerarse más claras sintácticamente para el modelo de calidad.

Subcaracterística Atómica: Restricciones legales (FUN 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Genera productos acordes con el dominio y con las restricciones legales del país	0	1	0 = No 1 = Si	Boolean
Permite configurar la herramienta para incorporar restricciones legales	0	1	0 = No 1 = Si	Boolean
Grado de completitud en que los	0	4	4 = Va más allá 3 = Completa	Escala de

productos responden a las restricciones legales del país			2 = Med. Completa 1 = Incompleta 0 = No está definida	liker
Total de Métricas = 3				

Tabla N° 161. Propuesta de métricas de la Subcaracterística Atómica Restricciones legales del MC-HDS-MDA

Fuente: Elaboración Propia.

Se incorpora la métrica **Permite configurar la herramienta para incorporar restricciones legales**, para poder determinar si la herramienta se puede configurar para incorporar restricciones legales, para los productos que se deban generar.

Subcaracterística Atómica: Restricciones técnicas (FUN 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Genera productos acordes con el dominio y con las restricciones técnicas del país	0	1	0 = No 1 = Si	Bolean
Permite configurar la herramienta para incorporar restricciones técnicas	0	1	0 = No 1 = Si	Bolean
Grado de completitud en que los productos responden a las restricciones técnicas del país	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Total de Métricas = 3				

Tabla N° 162. Propuesta de métricas de la Subcaracterística Atómica Restricciones técnicas del MC-HDS-MDA

Fuente: Elaboración Propia.

Se incorporan las métricas **Permite configurar la herramienta para incorporar restricciones técnicas** y **Grado de completitud en que los productos responden a las restricciones técnicas del país**; para poder determinar si la herramienta se puede configurar para incorporar restricciones técnicas, para los productos que se deban generar y que tan completos son los productos generados con las restricciones técnicas incorporadas.

Subcaracterística: Fiabilidad (FIA)

Subcaracterística Atómica: Integridad de los datos (FIA 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Permite solo accesos autorizados	0	1	0 = No 1 = Si	Bolean
Permite jerarquizar los accesos	0	1	0 = No 1 = Si	Bolean
Tiene mecanismos de "backup" o replicación de los datos	0	1	0 = No 1 = Si	Bolean
Permite recuperar los datos almacenados	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 4				

Tabla N° 163. Propuesta de métricas de la Subcaracterística Atómica Integridad de los datos del MC-HDS-MDA

Fuente: Elaboración Propia.

La métrica *Permite accesos no autorizados*, se cambia por, **Permite solo accesos autorizados**, en vista de que el valor 1, en lugar de ser un rasgo “bueno” para la herramienta es todo lo contrario, por lo que es absolutamente necesario cambiarlo, ya que en todo el modelo, los valores mayores representan los rasgos mejores, y los valores mínimos representan los rasgos calificados como no deseables, para una herramienta de desarrollo de software.

Subcaracterística Atómica: Backup automático (FIA 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Realiza copias de seguridad automáticamente.	0	1	0 = No 1 = Si	Boolean
Grado de satisfacción de las copias de seguridad realizadas por la herramienta	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 2				

Tabla N° 164. Propuesta de métricas de la Subcaracterística Atómica Backup automático del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Manejo de errores (FIA 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Capacidad de detectar comportamientos anormales	0	1	0 = No 1 = Si	Boolean
Guarda los trabajos en el momento de alguna interrupción	0	1	0 = No 1 = Si	Boolean
Los mensajes de error que genera están acordes a los errores presentados	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 3				

Tabla N° 165. Propuesta de métricas de la Subcaracterística Atómica Manejo de errores del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Tolerancia de fallas (FIA 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Conservación del nivel de desempeño al presentarse fallas (hardware, software, red, etc.)	0	1	0 = No 1 = Si	Boolean
Frecuencia con la que se presentan las fallas	0	4	4 = Nunca 3 = Pocas veces 2 = Algunas veces. 1 = Casi siempre 0 = Siempre	Escala de liker
Grado de satisfacción de la tolerancia a fallas de la h. CASE	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 166. Propuesta de métricas de la Subcaracterística Atómica Tolerancia de fallas del MC-HDS-MDA
Fuente: Elaboración Propia.

Se ha eliminado la métrica *Número de fallas presentadas durante una jornada de trabajo*, por no aportar nada al modelo de calidad propuesto.

Subcaracterística Atómica: Recuperabilidad (FIA 5)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Recupera su nivel de desempeño después de presentada una falla	0	1	0 = No 1 = Si	Bolean
Recupera los datos afectados durante la falla	0	1	0 = No 1 = Si	Bolean
Tiempo que requiere para recuperar el nivel de desempeño	0	4	4 = Inmediatamente 3 = Poco tiempo 2 = Mucho tiempo. 1 = No se recupera 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 167. Propuesta de métricas de la Subcaracterística Atómica Recuperabilidad del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística: Usabilidad (USA)

Subcaracterística Atómica: Amigabilidad del usuario (USA 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
La herramienta es intuitiva	0	1	0 = No 1 = Si	Bolean
Facilidad de uso	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Total de Métricas = 2				

Tabla N° 168. Propuesta de métricas de la Subcaracterística Atómica Amigabilidad del usuario del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Guía del usuario (USA 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Presencia de metáforas	0	1	0 = No 1 = Si	Bolean
Rapidez del sistema de ayuda	0	4	4 = Rápido 3 = Med. rápido 2 = Lento 1 = Muy lento 0 = No aplica	Escala de liker
Total de Métricas = 2				

Tabla N° 169. Propuesta de métricas de la Subcaracterística Atómica Guía del usuario del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Homogeneidad (USA 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Facilidad de detectar inconsistencia	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Total de Métricas = 1				

Tabla N° 170. Propuesta de métricas de la Subcaracterística Atómica Homogeneidad del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Adaptabilidad (USA 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Capacidad de parametrización	0	1	0 = No 1 = Si	Bolean
La interfaz de la herramienta se adapta a diferentes hábitos y modos culturales	0	1	0 = No 1 = Si	Bolean
Permite la parametrización o personalización	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 3				

Tabla N° 171. Propuesta de métricas de la Subcaracterística Atómica Adaptabilidad del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Claridad de control (USA 5)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Es versátil la navegación con la HDS.	0	1	0 = No 1 = Si	Boolean
Localización rápida de opciones	0	4	5 = Muy Rápido 3 = Rápido 2 = Med. rápido 1 = Lento 0 = Muy lento	Escala de liker
Sensibilidad de contexto	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 3				

Tabla N° 172. Propuesta de métricas de la Subcaracterística Atómica Claridad de control del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Manejo de errores (USA 6)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Son claros los mensajes emitidos por la HDS.	0	1	0 = No 1 = Si	Boolean
Permite la corrección de los errores evitando datos incorrectos y cambios en los procesos	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 2				

Tabla N° 173. Propuesta de métricas de la Subcaracterística Atómica Manejo de errores del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Concisión (USA 7)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Presencia de metáforas.	0	1	0 = No 1 = Si	Boolean
Sensibilidad de contexto	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 2				

Tabla N° 174. Propuesta de métricas de la Subcaracterística Atómica Concisión del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Fácil de aprender (USA 8)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Requiere de entrenamiento para aprender a usar de manera óptima la HDS.	0	1	0 = Si 1 = No	Boolean
La herramienta es intuitiva	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 2				

Tabla N° 175. Propuesta de métricas de la Subcaracterística Atómica Fácil de aprender del MC-HDS-MDA
Fuente: Elaboración Propia.

La métrica *Horas de entrenamiento requeridas para lograr la experticia necesaria para el uso completo de la herramienta, en una persona que desconozca la misma*, se ha cambiado por **Requiere de entrenamiento para aprender a usar de manera óptima la HDS**, por considerarse que la segunda es más significativa para el modelo de calidad. Esta métrica tiene la particularidad de ser de tipo boolean donde el Si equivale a 0 y el No equivale a un 1.

Subcaracterística Atómica: Calidad de la documentación de la herramienta (USA 9)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Grado de completitud del material de apoyo según la funcionalidad de la herramienta.	0	4	4 = Va más allá 3 = Completa 2 = Med. Completa 1 = Incompleta 0 = No está definida	Escala de liker
Calidad de la clasificación de los temas.	0	4	4 = Excelente 3 = Bueno 2 = Deficiente 1 = Malo 0 = No está definido	Escala de liker
Profundidad en el tratamiento de los temas.	0	1	0 = No 1 = Si	Boolean
Consistencia del material.	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 4				

Tabla N° 176. Propuesta de métricas de la Subcaracterística Atómica Calidad de la documentación de la herramienta del MC-HDS-MDA
Fuente: Elaboración Propia.

Se decide eliminar las métricas *Variedad del material de apoyo en cuanto a los usuarios a los que esta dirigido*, y *Variedad del material de apoyo de acuerdo a su presentación*, ya que ayudan a determinar el valor de otra métrica que se encuentra en la categoría Usuario.

Subcaracterística Atómica: Fácil de instalar (USA 10)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Facilidad para instalar la herramienta	0	4	4 = Fácil 3 = Difícil 2 = Muy difícil 1 = No se logro realizar 0 = No aplica	Escala de liker
Se requiere al proveedor para la instalación	0	1	0 = Si 1 = No	Boolean
Se requiere de material de apoyo para la instalación	0	1	0 = Si 1 = No	Boolean
Total de Métricas = 3				

Tabla N° 177. Propuesta de métricas de la Subcaracterística Atómica Fácil de instalar del MC-HDS-MDA
Fuente: Elaboración Propia.

Tal como se muestra en la Tabla N° 177, se eliminan las métricas *Tiempo real de instalación* y *Tasa del tiempo promedio de instalación*, por no aportar nada al modelo de calidad. Las métricas **Se requiere al proveedor para la instalación** y **Se requiere de material de apoyo para la instalación**, tienen la particularidad de ser de tipo boolean donde el Si equivale a 0 y el No equivale a un 1, por ser lo “ideal” que no se requiera de ningún tipo de ayuda.

Subcaracterística: Mantenibilidad (MANT)

Subcaracterística Atómica: Soporte del vendedor (MANT 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Soporte técnico permanente.	0	1	0 = No 1 = Si	Boolean
Soportes adicionales (web, mail, etc.).	0	1	0 = No 1 = Si	Boolean

Grado de satisfacción del tiempo promedio de respuesta ante una solicitud de respuesta	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 3				

Tabla N° 178. Propuesta de métricas de la Subcaracterística Atómica Mantenibilidad del MC-HDS-MDA
Fuente: Elaboración Propia.

La métrica *Horas semanales de soporte técnico*, se elimina por no aportar nada al modelo de calidad propuesto.

Subcaracterística Atómica: Habilidad de la herramienta para hacer cambios en la metodología (MANT 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
El vendedor puede modificar la herramienta para mantener el soporte a las metodologías	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 1				

Tabla N° 179. Propuesta de métricas de la Subcaracterística Atómica Habilidad de la herramienta para hacer cambios en la metodología del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Actualización (MANT 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Grado de mejoramiento de la herramientas	0	4	4 = Mas de 1 ver. Por año 3 = Una por año 2 = Entre 1 y 2 años. 1 = Entre 2 y 5 años 0 = Mas de 5 años	Escala de liker
Total de Métricas = 1				

Tabla N° 180. Propuesta de métricas de la Subcaracterística Atómica Actualización del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Expansibilidad (MANT 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
El desarrollador de la H. CASE genera "upgrade" de la herramienta	0	1	0 = No 1 = Si	Bolean
Grado de facilidad para obtener los "upgrade" de la herramienta	0	4	4 = Fácil 3 = Dificil 2 = Muy dificil 1 = No se logro realizar 0 = No aplica	Escala de liker
Frecuencia con la cual el proveedor lanza al mercado los "upgrade"	0	4	4 = Siempre 3 = Casi siempre 2 = Algunas veces. 1 = Pocas veces 0 = Nunca	Escala de liker
Total de Métricas = 3				

Tabla N° 181. Propuesta de métricas de la Subcaracterística Atómica Expansibilidad del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística: Eficiencia (EFI)

Subcaracterística Atómica: Tiempo de respuesta aceptable (EFI 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica

Grado de satisfacción del tiempo de respuesta.	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 1				

Tabla N° 182. Propuesta de métricas de la Subcaracterística Atómica Tiempo de respuesta aceptable del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Requerimientos de almacenamiento (EFI 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Requerimiento de almacenamiento de la HDS con respecto a lo necesario por una HDS promedio	0	4	4 = Menos del 25% 3 = Aprox. 25% 2 = Aprox. 50% 1 = Aprox. 75% 0 = Mas del 100%	Escala de liker
Total de Métricas = 1				

Tabla N° 183. Propuesta de métricas de la Subcaracterística Atómica Requerimientos de almacenamiento de datos del MC-HDS-MDA

Fuente: Elaboración Propia.

Se decide eliminar la métrica *Cantidad de memoria secundaria necesaria para almacenar a la herramienta y a las bases de datos requeridas y generadas por ella*, en vista de que el valor que arroje es totalmente irrelevante para que se encuentre dentro del modelo de calidad. Claro esta, es un valor importante para calcular la siguiente métrica, pero ella por si sola no se debe incluir. También se elimina la métrica *La plataforma de hardware de la organización cuenta con la cantidad de memoria secundaria necesaria*, por depender de la organización como cliente y no de la herramienta como tal, y en consecuencia es incorporada como métrica de la categoría usuario. Por otro lado, se crea una nueva métrica **Requerimiento de almacenamiento de la HDS con respecto a lo necesario por una HDS promedio**, en vista de que esta respuesta si es significativa para el modelo y para la Subcaracterística atómica que se esta midiendo.

Subcaracterística Atómica: Capacidad de memoria aceptable (EFI 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Requerimiento de memoria de la HDS con respecto a lo necesario por una HDS promedio	0	4	4 = Menos del 25% 3 = Aprox. 25% 2 = Aprox. 50% 1 = Aprox. 75% 0 = Mas del 100%	Escala de liker
Total de Métricas = 1				

Tabla N° 184. Propuesta de métricas de la Subcaracterística Atómica capacidad de memoria aceptable del MC-HDS-MDA

Fuente: Elaboración Propia.

Se elimina la métrica *Cantidad de memoria RAM mínima requerida para cargar y operar la herramienta satisfactoriamente*, ya que por si sola no aporta nada al modelo, ahora si se relaciona con la disponibilidad de memoria RAM que posee la organización donde será instalada la HDS, si es importante. También se elimina la métrica *Grado de desempeño de la herramienta con respecto a la cantidad de memoria RAM disponible en el ambiente de trabajo de la organización*, por

dependen exclusivamente del cliente. Y al igual que en el caso anterior se crea una nueva métrica, **Requerimiento de memoria de la HDS con respecto a lo necesario por una HDS promedio**, toda vez que este valor si es significativo a la hora de compararla con otra herramienta de desarrollo de software.

Subcaracterística Atómica: Velocidad de procesamiento aceptable (EFI 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Requerimiento de procesador de la HDS con respecto a lo necesario por una HDS promedio	0	4	4 = Menos del 25% 3 = Aprox. 25% 2 = Aprox. 50% 1 = Aprox. 75% 0 = Mas del 100%	Escala de liker
Aceptabilidad del desempeño de la herramienta durante la ejecución del proyecto	0	4	4 = Excelente 3 = Bueno 2 = Deficiente 1 = Malo 0 = Muy malo	Escala de liker
Grado de rapidez con que se acceden las opciones dentro de la herramienta en ejecución	0	4	4 = Muy Rápido 3 = Rápido 2 = Med. rápido 1 = Lento 0 = Muy lento	Escala de liker
Total de Métricas = 3				

Tabla N° 185. Propuesta de métricas de la Subcaracterística Atómica Velocidad de procesamiento aceptable del MC-HDS-MDA

Fuente: Elaboración Propia.

Se decide eliminar la métrica *Tipo y velocidad del procesador requerido para operar con la herramienta de manera satisfactoria*, e incorporar a la métrica **La organización cuenta con el tipo y velocidad de procesador necesario para que la HDS se desempeñe satisfactoriamente**, en vista de que la primera no aporta nada al modelo de calidad propuesto. Posteriormente se elimina esta métrica, *La organización cuenta con el tipo y velocidad del procesador necesario para que la HDS se desempeñe satisfactoriamente*, por depender exclusivamente del cliente. Para darle mayor robustez a los valores que arroje esta subcaracterística atómica se crea la métrica **Requerimiento de procesador de la HDS con respecto a lo necesario por una HDS promedio**, que depende necesariamente de la herramienta.

Subcaracterística: Portabilidad (POR)

Subcaracterística Atómica: Portabilidad a diferentes plataformas de hardware (POR 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Portabilidad a otras plataformas de hardware	0	1	0 = No 1 = Si	Boolean
Los productos de la h. CASE se pueden ejecutar para distintas plataformas de hardware	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 2				

Tabla N° 186. Propuesta de métricas de la Subcaracterística Atómica Portabilidad a diferentes plataformas de hardware del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Compatibilidad con diferentes sistemas operativos (POR 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
La herramienta puede ejecutarse sobre varias versiones de un mismo sistema operativo	0	1	0 = No 1 = Si	Boolean
Se ejecuta sobre varios sistemas operativos	0	1	0 = No 1 = Si	Boolean
Porcentaje de sistemas operativos sobre los cuales opera la herramienta	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Total de Métricas = 3				

Tabla N° 187. Propuesta de métricas de la Subcaracterística Atómica Compatibilidad con diferentes sistemas operativos del MC-HDS-MDA

Fuente: Elaboración Propia.

La métrica *Cantidad de sistemas operativos sobre los cuales opera la herramienta*, también cambia a **Porcentaje de sistemas operativos sobre los cuales opera la herramienta**, y se valora a través de una escala de liker.

Subcaracterística Atómica: Habilidad para mover datos entre versiones de la herramienta (POR 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
La herramienta puede usar datos provenientes de otra versión de la herramienta	0	1	0 = No 1 = Si	Boolean
Permite guardar como otra versión de la h. CASE	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 2				

Tabla N° 188. Propuesta de métricas de la Subcaracterística Atómica Habilidad para mover datos entre versiones de la herramienta del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Portabilidad con sistemas Windows (POR 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
La herramienta se ejecuta bajo plataformas Windows	0	1	0 = No 1 = Si	Boolean
Se ejecuta en la versión más actualizada de Windows	0	1	0 = No 1 = Si	Boolean
Porcentaje de versiones de Windows que soportan a la h. CASE	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Total de Métricas = 3				

Tabla N° 189. Propuesta de métricas de la Subcaracterística Atómica Portabilidad con sistemas Windows del MC-HDS-MDA

Fuente: Elaboración Propia.

La métrica *Versiones de Windows que soportan a la h. CASE*, también cambia a **Porcentaje de Versiones de Windows que soportan a la h. CASE**, y se valora a través de una escala de liker.

Por otro lado se anexa una nueva métrica **Se ejecuta en la versión más actualizada de Windows**, y así determinar en esta subcaracterística atómica la compatibilidad de la HDS con las versiones más actualizadas de este sistema operativo.

Subcaracterística Atómica: Portabilidad con sistemas Linux (POR 5)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
La herramienta se ejecuta bajo plataformas Linux	0	1	0 = No 1 = Si	Boolean
Se ejecuta en la versión más actualizada de Linux	0	1	0 = No 1 = Si	Boolean
Porcentaje de versiones de Linux que soportan a la h. CASE	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Total de Métricas = 3				

Tabla N° 190. Propuesta de métricas de la Subcaracterística Atómica Portabilidad con sistemas Linux del MC-HDS-MDA

Fuente: Elaboración Propia.

La métrica *Versiones de Linux que soportan a la h. CASE*, también cambia a **Porcentaje de Versiones de Linux que soportan a la h. CASE**, y se valora a través de una escala de liker. Además se anexa una nueva métrica **Se ejecuta en la versión más actualizada de Linux**, y así determinar en esta subcaracterística atómica la compatibilidad de la HDS con las versiones más actualizadas de este sistema operativo.

CLIENTE

Subcaracterística: Ambiente en el cual opera la herramienta CASE (AMB)

Subcaracterística Atómica: Características de hardware requeridas por la herramienta (AMB 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
La plataforma de hardware de la organización cuenta con la cantidad de memoria secundaria necesaria	0	1	0 = No 1 = Si	Boolean
Grado de desempeño de la herramienta con respecto a la cantidad de memoria RAM disponible en el ambiente de trabajo de la organización	0	4	4 = Excelente 3 = Bueno 2 = Deficiente 1 = Malo 0 = Muy malo	Escala de liker
La organización cuenta con el tipo y velocidad del procesador necesario para que la HDS se desempeñe satisfactoriamente	0	1	0 = No 1 = Si	Boolean
La organización o el usuario posee la plataforma de hardware necesaria para soportar a la h. CASE	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 4				

Tabla N° 191. Propuesta de métricas de la Subcaracterística Atómica Características de hardware requeridas por la herramienta del MC-HDS-MDA

Fuente: Elaboración Propia.

Se incorporan las métricas **La plataforma de hardware de la organización cuenta con la cantidad de memoria secundaria necesaria**, **Grado de desempeño de la herramienta con respecto a la cantidad de memoria RAM disponible en el ambiente de trabajo de la organización** y **La organización cuenta con el tipo y velocidad del procesador necesario para que la HDS se desempeñe satisfactoriamente**; por depender de las características del hardware del cliente.

Subcaracterística Atómica: Ambiente de software requerido por la herramienta (AMB 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
La organización cuenta con el software requerido para ejecutar la HDS	0	1	0 = No 1 = Si El software requerido sea igual al software de la organización	Boolean
El software requerido es compatible con el software del cliente.	0	1	0 = No 1 = Si El software requerido sea compatible con el software de la organización	Boolean
Tasa de cobertura de los sistemas operativos requeridos por la organización	0	1	<u>Nro de S.O. de la org. Que son requeridas por la HDS</u> Nro de S.O. requeridos por la HDS	Tasa de Cobertura
Tasa de la capacidad de la plataforma de la organización para soportar la herramienta	0	1	<u>Nro de plataformas de la org. Que son requeridas por la HDS</u> Nro de plataformas requeridos por la HDS	Tasa de Cobertura
Total de Métricas = 4				

Tabla N° 192. Propuesta de métricas de la Subcaracterística Atómica Ambiente de software requerido por la herramienta del MC-HDS-MDA

Fuente: Elaboración Propia.

Se eliminan las métricas *Cantidad de plataformas de software que soportan a la herramienta* y *Cantidad de plataformas de software que posee la organización*, por no aportar nada al modelo de calidad, al igual que en otras métricas, estos valores sirven para calcular la Tasa de la capacidad de la plataforma de la organización para soportar la herramienta, y esta sí es una métrica interesante para el modelo de calidad propuesto. Por otro lado, se agregan dos métricas que son **La organización cuenta con el software requerido para ejecutar la HDS** y **el Software requerido es compatible con el software del cliente**, en vista de que si aportan información valiosa para saber si la organización posee la plataforma de software necesaria y suficiente para poner en ejecución la HDS.

Subcaracterística Atómica: Repositorio de software (AMB 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
La HDS cuenta con un repositorio común en caso de ser requerido por el cliente	0	1	0 = No 1 = Si	Boolean

Total de Métricas = 1

Tabla N° 193. Propuesta de métricas de la Subcaracterística Atómica Repositorio de software del MC-HDS-MDA

Fuente: Elaboración Propia.

La métrica *repositorio común* cambia a **La HDS cuenta con un repositorio común en caso de ser requerido por el cliente**, para que se pueda ver claramente como una necesidad del cliente.

Subcaracterística Atómica: Ambiente físico de la herramienta (AMB 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Requiere actualizar un repositorio en múltiples sitios	0	1	0 = No 1 = Si	Boolean
Permite distribuir geográficamente los nodos de una red	0	1	0 = No 1 = Si	Boolean
Porcentaje de usuarios concurrentes que admite la HDS con respecto a los usuarios que requiere la organización	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Grado de satisfacción en el manejo de las concurrencias	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Tasa de cobertura de la concurrencia requerida	0	1	<small>Nro de usuarios concurrentes que soporta la HDS</small> <small>Nro de usuarios concurrentes requeridos</small>	Tasa de Cobertura
Total de Métricas = 5				

Tabla N° 194. Propuesta de métricas de la Subcaracterística Atómica Ambiente físico de la herramienta del MC-HDS-MDA

Fuente: Elaboración Propia.

La métrica *Número de usuarios concurrentes que admite la HDS con respecto a los usuarios que requiere al organización*, también cambia a **Porcentaje de usuarios concurrentes que admite la HDS con respecto a los usuarios que requiere al organización**, y se valora a través de una escala de liker.

Subcaracterística: Integrabilidad de la herramienta CASE (INT)

Subcaracterística Atómica: Compatibilidad con los elementos del ambiente (INT 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Grado de portabilidad.	0	4	4 = Excelente 3 = Buena 2 = Deficiente 1 = Mala 0 = No esta definida	Escala de liker
Portabilidad a otras plataformas.	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 2				

Tabla N° 195. Propuesta de métricas de la Subcaracterística Atómica Compatibilidad con los elementos del ambiente del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Integración de data (INT 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica

Calidad de la integración.	0	4	4 = Excelente 3 = Buena 2 = Deficiente 1 = Mala 0 = No esta definida	Escala de liker
Es una herramienta integrada.	0	1	0 = No 1 = Si	Boolean
Integración de data.	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 3				

Tabla N° 196. Propuesta de métricas de la Subcaracterística Atómica Integración de data del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Integración de control (INT 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Integración de actividades.	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 1				

Tabla N° 197. Propuesta de métricas de la Subcaracterística Atómica Integración de control del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Integración de presentación (INT 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Integración de interface.	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 1				

Tabla N° 198. Propuesta de métricas de la Subcaracterística Atómica Integración de presentación del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Acceso a metadata (INT 5)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Repositorio común.	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 1				

Tabla N° 199. Propuesta de métricas de la Subcaracterística Atómica Acceso a metadata del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística: Aspectos de la aplicación de la herramienta CASE (APL)

Subcaracterística Atómica: Ambiente de hardware y software de los productos de la herramienta (APL 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Los productos de la herramienta requieren de un ambiente de hardware y software especial	0	1	0 = No 1 = Si	Boolean
La organización posee la plataforma de hardware que requieren los productos de la h. CASE	0	1	0 = No 1 = Si	Boolean
La organización posee la plataforma de software que requieren los productos de la h. CASE	0	1	0 = No 1 = Si	Boolean
Grado de satisfacción del desempeño de los productos de la herramienta en el ambiente de software	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Grado de satisfacción del	0	4	4 = Muy Satisfecho 3 = Satisfecho	Escala de

desempeño de los productos de la herramienta en el ambiente de hardware			2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	liker
Total de Métricas = 5				

Tabla N° 200. Propuesta de métricas de la Subcaracterística Atómica Ambiente de hardware y software de los productos de la herramienta del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Conformidad de estándares – productos de la herramienta (APL 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
La herramienta genera productos que están acordes con los estándares establecidos por la organización	0	1	0 = No 1 = Si	Boolean
La herramienta permite definir los estándares que deben llevar los productos de la organización	0	1	0 = No 1 = Si	Boolean
Porcentaje de etapas de desarrollo soportadas	0	4	4 = 100% 3 = 75% 2 = 50% 1 = 25% 0 = 0%	Escala de liker
Tasa de cobertura de las etapas requeridas	0	1	$\frac{\text{Nro de etapas requeridas y soportadas por la HDS}}{\text{Nro de etapas requeridas}}$	Tasa de Cobertura
Total de Métricas = 4				

Tabla N° 201. Propuesta de métricas de la Subcaracterística Atómica Conformidad de estándares – productos de la herramienta del MC-HDS-MDA
Fuente: Elaboración Propia.

La métrica *Total de etapas de desarrollo soportadas*, también cambia a **Porcentaje de etapas de desarrollo soportadas**, y se valora a través de una escala de liker.

Subcaracterística Atómica: Dominio de la aplicación (APL 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Soporta el tipo de aplicación que el usuario desee desarrollar	0	1	0 = No 1 = Si	Boolean
Grado de satisfacción del soporte de la herramienta con respecto a la aplicación requerida por la organización	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Total de Métricas = 2				

Tabla N° 202. Propuesta de métricas de la Subcaracterística Atómica Dominio de la aplicación del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Tamaño de la aplicación soportada (APL 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Soporta el tamaño de la aplicación que el usuario desea desarrollar	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 1				

Tabla N° 203. Propuesta de métricas de la Subcaracterística Atómica Tamaño de la aplicación soportada del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Lenguajes soportados (APL 5)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Tasa de cobertura del código generado	0	1	$\frac{\text{Nro de lenguajes requeridos y soportados por la HDS}}{\text{Nro de lenguajes requeridos}}$	Tasa de Cobertura
Tasa de cobertura de los lenguajes en que compila código fuente	0	1	$\frac{\text{Nro de lenguajes requeridos y soportados por la HDS}}{\text{Nro de lenguajes requeridos}}$	Tasa de Cobertura
Tasa de cobertura de la reestructuración de código fuente	0	1	$\frac{\text{Nro de lenguajes requeridos y soportados por la HDS}}{\text{Nro de lenguajes requeridos}}$	Tasa de Cobertura
Tasa de cobertura de traducción del código fuente	0	1	$\frac{\text{Nro de lenguajes requeridos y soportados por la HDS}}{\text{Nro de lenguajes requeridos}}$	Tasa de Cobertura
Tasa de cobertura de los lenguajes de ingeniería de reverso	0	1	$\frac{\text{Nro de lenguajes requeridos y soportados por la HDS}}{\text{Nro de lenguajes requeridos}}$	Tasa de Cobertura
Total de Métricas = 5				

Tabla N° 204. Propuesta de métricas de la Subcaracterística Atómica Lenguajes soportados del MC-HDS-MDA

Fuente: Elaboración Propia.

Las métricas *Cantidad de lenguajes en que genera código* y *Cantidad de lenguajes soportados por la ingeniería de reverso*, se eliminan por no depender de las necesidades del cliente. Se anexan las métricas **Tasa de cobertura de los lenguajes en que compila código fuente**, **Tasa de cobertura de la reestructuración de código fuente** y **Tasa de cobertura de traducción del código fuente**, porque sus valores dependen del cliente para quien se este realizando la evaluación.

Subcaracterística Atómica: Bases de datos soportadas (APL 6)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Tasa de la manipulación de los manejadores de bases de datos requeridos por la organización.	0	1	$\frac{\text{Nro de manejadores requeridos y soportados por la HDS}}{\text{Nro de manejadores requeridos}}$	Tasa de Cobertura
Tasa de cobertura de la generación de esquemas de Bases de Datos	0	1	$\frac{\text{Nro de manejadores requeridos y soportados por la HDS}}{\text{Nro de manejadores requeridos}}$	Tasa de Cobertura
Total de Métricas = 2				

Tabla N° 205. Propuesta de métricas de la Subcaracterística Atómica Bases de datos soportadas del MC-HDS-MDA

Fuente: Elaboración Propia.

Se elimina la métrica *Total de manejadores de bases de datos que maneja*, por arrojar un valor que no depende del cliente. Por otro lado se incorpora la métrica **Tasa de cobertura de la generación de esquemas de Bases de Datos**, por depender exclusivamente de una necesidad del cliente.

Subcaracterística Atómica: Metodologías soportadas (APL 7)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Tasa de cobertura de las metodologías requeridas.	0	1	$\frac{\text{Nro de metodologías requeridos y soportados por la HDS}}{\text{Nro de metodologías requeridos}}$	Tasa de Cobertura
Grado de consistencia de la diagramación respecto a la metodología adoptada por la organización.	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker

Total de Métricas = 2

Tabla N° 206. Propuesta de métricas de la Subcaracterística Atómica Metodologías soportadas del MC-HDS-MDA

Fuente: Elaboración Propia.

Se incorpora la métrica **Grado de consistencia de la diagramación respecto a la metodología adoptada por la organización** a la subcaracterística atómica Metodologías soportadas, por depender de las necesidades del cliente. La métrica *Total de metodologías soportadas*, se elimina por no estar relacionada directamente con los requerimientos del cliente.

Subcaracterística Atómica: Globalización (APL 8)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Varia la filosofía de trabajo de acuerdo con la cultura o país	0	1	0 = No 1 = Si	Boolean
Grado de satisfacción de la variación de filosofía de trabajo	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Cambia el idioma de acuerdo a la cultura o país	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 3				

Tabla N° 207. Propuesta de métricas de la Subcaracterística Atómica Internacionalización del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística: Proceso de Adquisición (ADQ)

Subcaracterística Atómica: Costos de implantación de la herramienta (ADQ 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Tasa de cobertura del costo de adquisición de la herramienta.	0	1	<u>Costo de adquisición de la HDS</u> Costo máximo que puede pagar el cliente	Tasa de Cobertura
Tasa de cobertura de costo de hardware	0	1	<u>Costo de hardware necesario para ejecutar de la HDS</u> Costo máximo que puede pagar el cliente	Tasa de Cobertura
Tasa de cobertura de costo de software	0	1	<u>Costo del software necesario para ejecutar de la HDS</u> Costo máximo que puede pagar el cliente	Tasa de Cobertura
Tasa de cobertura de los costos de instalación	0	1	<u>Costo de instalación de la HDS</u> Costo máximo que puede pagar el cliente	Tasa de Cobertura
Tasa de cobertura de los costos de mantenimiento inicial	0	1	<u>Costo de mantenimiento 190nicial de la HDS</u> Costo máximo que puede pagar el cliente	Tasa de Cobertura
Total de Métricas = 5				

Tabla N° 208. Propuesta de métricas de la Subcaracterística Atómica Costos de implantación de la herramienta del MC-HDS-MDA

Fuente: Elaboración Propia.

Se propone eliminar las métricas: *Costos de adquisición, costos de entrenamiento, costos de hardware, costos de software, costos de instalación y costos de mantenimiento inicial*, por no aportar información relevante para el modelo de calidad.

Subcaracterística Atómica: Documentación de la herramienta (ADQ 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Tasa de cobertura del material de apoyo en cuanto a su presentación.	0	1	Presentación del material de apoyo de la HDS Presentación del material de apoyo requerido por el cliente	Tasa de Cobertura
Tasa de cobertura del material de apoyo en cuanto a los usuarios a los que esta dirigido.	0	1	Material d apoyo de la HDS Material de apoyo en cuanto a usuarios requerido por el cliente	Tasa de Cobertura
Total de Métricas = 2				

Tabla N° 209. Propuesta de métricas de la Subcaracterística Atómica Documentación de la herramienta del MC-HDS-MDA

Fuente: Elaboración Propia.

En esta Subcaracterística atómica se incorporan las métricas **Tasa de cobertura del material de apoyo en cuanto a su presentación** y **Tasa de cobertura del material de apoyo en cuanto a los usuarios a los que esta dirigido**, de las cuales sus valores dependen del cliente. Y por otro lado son métricas asociadas al material de apoyo que viene incorporado con la HDS a la hora de su compra.

COMERCIALIZACIÓN

Subcaracterística: Proceso de Venta (VEN)

Subcaracterística Atómica: Políticas de licencia (VEN 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
El proveedor establece políticas de licencias con respecto a la herramienta.	0	1	0 = No 1 = Si	Bolean
El proveedor otorga el derecho a copias	0	1	0 = No 1 = Si	Bolean
El proveedor restringe algún uso de la herramienta.	0	1	0 = No 1 = Si	Bolean
La herramienta tiene garantía	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 4				

Tabla N° 210. Propuesta de métricas de la Subcaracterística Atómica Políticas de licencia del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Restricciones de exportación (VEN 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
El proveedor restringe la exportación de la herramienta	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 1				

Tabla N° 211. Propuesta de métricas de la Subcaracterística Atómica Restricciones de exportación del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística: Implementación (IMPL)

Subcaracterística Atómica: Efectividad del costo (IMPL 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Es la herramienta rentable para la organización. (Análisis costo – beneficio)	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 1				

Tabla N° 212. Propuesta de métricas de la Subcaracterística Atómica Efectividad del costo del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Restricciones de desarrollo - entregas (IMPL 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
La herramienta es intuitiva.	0	1	0 = No 1 = Si	Bolean
Tiene tutorial	0	1	0 = No 1 = Si	Bolean
Grado de satisfacción del servicio del tutorial	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
La herramienta permite ir más allá del producto requerido por la organización	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 4				

Tabla N° 213. Propuesta de métricas de la Subcaracterística Atómica Restricciones de desarrollo – entregas del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Actividades requeridas por los usuarios de la organización (IMPL 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
La herramienta soporta actividades especiales con respecto al ambiente del usuario (desde la perspectiva del proveedor)	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 1				

Tabla N° 214. Propuesta de métricas de la Subcaracterística Atómica Actividades requeridas por los usuarios de la organización del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística Atómica: Necesidades de infraestructura (IMPL 4)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Requiere de condiciones de infraestructura especiales para su operación. (Espacio del piso, condiciones eléctricas, condiciones de temperatura, etc.)	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 1				

Tabla N° 215. Propuesta de métricas de la Subcaracterística Atómica Necesidades de infraestructura del MC-HDS-MDA

Fuente: Elaboración Propia.

Subcaracterística: Indicadores de Soporte (SOP)

Subcaracterística Atómica: Perfil del vendedor (SOP 1)

Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Es considerada la empresa desarrolladora estable económicamente	0	1	0 = No 1 = Si	Boolean
Tiempo de la empresa desarrolladora en el mercado con esa herramienta.	0	4	4 = Más de 10 años 3 = Menos de 10 y más de 5 2 = aprox. 5 años 1 = Menos de 5 y más de 1 0 = Menos de un año	Escala de liker
Participación en el mercado con esta herramienta.	0	4	4 = 100% del merc. 3 = Aprox. 75% del merc. 2 = 50% del merc. 1 = Aprox. 25% del merc. 0 = Menos del 25% del merc.	Escala de liker
Total de Métricas = 3				

Tabla N° 216. Propuesta de métricas de la Subcaracterística Atómica Perfil del vendedor del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Perfil del producto (SOP 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
Publicaciones expertas positivas.	0	1	0 = No 1 = Si	Boolean
Grado de satisfacción promedio de otras empresas con el uso de la HDS.	0	4	4 = Muy Satisfecho 3 = Satisfecho 2 = Poco Satisf. 1 = Insatisfecho 0 = No aplica	Escala de liker
Experiencia positiva con otras empresas	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 3				

Tabla N° 217. Propuesta de métricas de la Subcaracterística Atómica Perfil del Producto del MC-HDS-MDA
Fuente: Elaboración Propia.

Subcaracterística Atómica: Disponibilidad de entrenamiento (SOP 3)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
El proveedor brinda entrenamiento a sus clientes	0	1	0 = No 1 = Si	Boolean
El proveedor dispone de cursos básicos al año	0	1	0 = No 1 = Si	Boolean
El proveedor dispone de cursos avanzados al año	0	1	0 = No 1 = Si	Boolean
El proveedor brinda soporte técnico post-entrenamiento	0	1	0 = No 1 = Si	Boolean
El proveedor brinda soporte técnico las 24 horas	0	1	0 = No 1 = Si	Boolean
Total de Métricas = 5				

Tabla N° 218. Propuesta de métricas de la Subcaracterística Atómica Disponibilidad de entrenamiento del MC-HDS-MDA
Fuente: Elaboración Propia.

Se propone eliminar las métricas *Cantidad de cursos básicos al año*, *Cantidad de cursos avanzados al año*, *Cantidad de tipos de cursos*, *Cobertura de cursos básicos al año*, y *Cobertura de cursos avanzados al año*, e incorporar **El proveedor brinda entrenamiento a sus clientes**, **El proveedor dispone de cursos básicos al año**, **El proveedor dispone de cursos avanzados al año**, por considerarlas más significativas para el modelo. También se agregan las métricas **El proveedor brinda soporte técnico post-entrenamiento** y **El proveedor brinda soporte técnico las 24 horas**, como indicadores importantes de los servicios post-venta que otorgue el proveedor.

Subcaracterística: Evaluación o Certificación (EVA)

Subcaracterística Atómica: Evaluación o Certificación del desarrollador (EVA 1)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
El desarrollador posee certificaciones.	0	1	0 = No 1 = Si	Bolean
La organización que evalúa o certifica es reconocida	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 2				

Tabla N° 219. Propuesta de métricas de la Subcaracterística Atómica Evaluación o certificación del desarrollador del MC-HDS-MDA

Fuente: Elaboración Propia.

Tal como se puede apreciar en la Tabla N° 218, la métrica *Posee certificaciones* se le ha agregado **El desarrollador posee certificaciones**, por considerarse más clara para el evaluador.

Subcaracterística Atómica: Certificación del producto (EVA 2)				
Nombre	Valor Min	Valor Max	Formulación	Tipo de Métrica
La herramienta posee certificaciones	0	1	0 = No 1 = Si	Bolean
La organización que certificó es reconocida	0	1	0 = No 1 = Si	Bolean
Total de Métricas = 2				

Tabla N° 220. Propuesta de métricas de la Subcaracterística Atómica Certificación del producto del MC-HDS-MDA

Fuente: Elaboración Propia.

En resumen el Modelo Teórico ha quedado conformado por 487 métricas que corresponden a la capa 6 de la arquitectura del MC-HDS-MDA, discriminadas de la siguiente manera:

- característica: Ciclo de Vida – procesos medulares: 242 métricas.
- característica: Ciclo de Vida – procesos de apoyo: 89 métricas.
- característica: Calidad: 76 métricas.
- característica: Cliente: 53 métricas
- característica: Comercialización: 27 métricas

Por otro lado, es muy importante acotar que durante el desarrollo de este capítulo, se hizo un análisis semántico mucho más profundo al MC-CASE, lo que originó algunos cambios que se reflejaron en la propuesta de métricas descrita anteriormente.

Una vez construido el Modelo Teórico, se le debe dar el soporte numérico, para esto a continuación se describirá todos los elementos que permiten elaborar el Modelo Matemático.

6.1.2 MODELO MATEMÁTICO DEL MC-HDS-MDA

Este modelo tiene como objetivo principal establecer matemáticamente la forma de arrojar los resultados una vez que se aplique el modelo teórico a una o más herramientas de desarrollo de software. Para ilustrar mejor su construcción, esta sección estará dividida en tres subsecciones que son: **Consideraciones Generales**, la cual pretende establecer algunas premisas que se utilizaron para conformar el modelo; luego estará **Calculo de la Calidad de la HDS**, donde se hace una abstracción de la arquitectura del MC-HDS-MDA para generar las relaciones que permiten ir calculando la Calidad de la Herramienta de Desarrollo de Software y por último se encuentra la **Determinación del Índice de Confiabilidad del Evaluador**, donde se califica de acuerdo a unas características a los evaluadores encargados de valorar a las HDS. A continuación se detallaran cada una de estas subsecciones.

6.1.2.1 CONSIDERACIONES GENERALES.

- El Modelo se instancia de arriba hacia abajo, y se ejecuta de abajo hacia arriba, es decir, los resultados deben ir organizándose de abajo hacia arriba.
- El Cliente es quien establece los requerimientos y las prioridades, es decir, lo que necesita, y lo que prefiere y lo que esta dispuesto a omitir, por ejemplo, el cliente pudiera establecer, que para él es necesario que se cumplan sus requisitos, pero en cuanto a calidad, debe ser la mejor, sin importar los aspectos relacionados con el proveedor, ni los aspectos relacionados con el ciclo de vida.
- Esta consideración conduce a que el modelo matemático tiene que permitir el establecimiento de prioridades, lo que a su vez caracteriza a este Meta-Modelo como un modelo de calidad **dirigido al cliente**. Estas prioridades en el Modelo Matemático serán denominadas **Pesos**.
- El conjunto de prioridades se pueden establecer a partir de la capa 5, de abajo hacia arriba, es decir, la capa 6 no admite el establecimiento de prioridades.
- La escala de prioridades o pesos podrá ser 1, 2, 3, 4 o 5, por defecto el sistema asumirá el valor de 1, es decir, si el cliente no establece prioridades, el encargado de instanciar al modelo colocará el valor de 1 en los pesos.

- Si se quisiera omitir alguna Subcaracterística atómica, subcaracterística, característica tipo, característica o categoría, se le colocará el valor de 0 en el peso.
- Cada métrica tiene un valor mínimo y máximo que puede admitir.
- A la hora de realizar los cálculos de la Calidad, todos los valores de las métricas deben ser normalizados a la escala del 0 al 1. Para las métricas Bolean y Tasa de Cobertura no hay problema porque su escala va del 0 al 1; pero en el caso de la métrica tipo Escala de Liker si hay que hacer una conversión, para esto se utilizará la siguiente relación:
 - 0 equivale a 0
 - 1 equivale a 0,25
 - 2 equivale a 0,5
 - 3 equivale a 0,75
 - 4 equivale a 1

6.1.2.2 CALCULO DE LA CALIDAD DE LA HDS.

Para ilustrar la abstracción que se hizo de la arquitectura del MC-HDS-MDA, y poder generar cuantitativamente la relación existente entre las diferentes métricas, subcaracterísticas atómicas, subcaracterísticas y características, con sus respectivos **pesos** a partir de las Subcaracterísticas atómicas hacia arriba, se muestra a continuación la Figura N° 33.

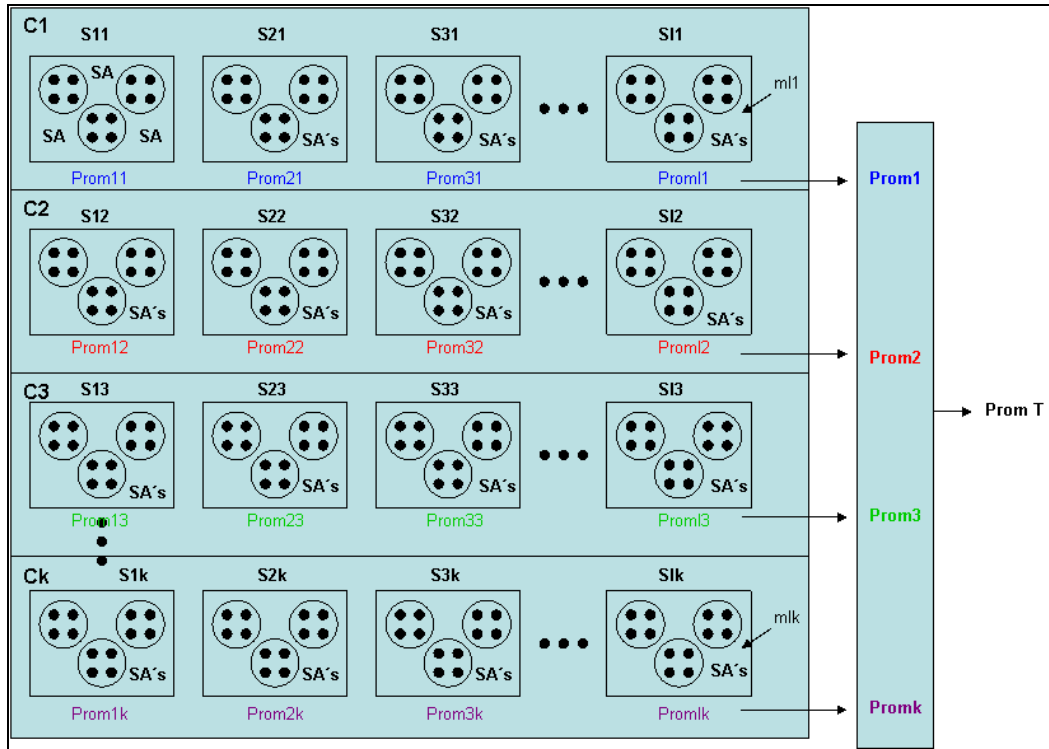


Figura N° 33. Abstracción a través de matrices del MC-HDS-MDA
Fuente: Elaboración Propia.

De acuerdo con la Figura N° 33, el Modelo queda dividido en cuatro grandes características, representadas en este caso por C, que va desde 1 hasta k, cada una de las características o de las C_k, posee un conjunto de Subcaracterísticas representadas por S, donde el numero de S por C, no necesariamente es igual, por lo que en la figura se muestra como l subcaracterísticas por cada C. Dentro de cada S hay un conjunto j de Subcaracterísticas atómicas representadas por SA, y cada SA_j, posee un número i de métricas, donde el i varía para cada una de las SA. Ahora bien, como cada Subcaracterística atómica puede asumir un peso determinado entre (0 y 5) dependiendo de las prioridades del cliente, se tiene que:

La calidad (representada por Q) de la J-ésima subcaracterística atómica, de la l-ésima subcaracterística, de la k-ésima característica, viene representada a través de la siguiente relación.

$$Q_{jlk} = \left[\sum_{i=1}^{n_{jlk}} M_i \right] \times P_{jlk} ; (a)$$

Donde P_{jlk} es el peso de cada j-ésima subcaracterística atómica.

En consecuencia, la Calidad de la l-ésima subcaracterística de la k-ésima característica, viene representada por la siguiente relación,

$$Q_{l_k} = \frac{\sum_{j=1}^{m_{l_k}} Q_{j_{l_k}}}{\sum_{j=1}^{m_{l_k}} P_{j_{l_k}}} ; (b)$$

Esta relación se determina por un promedio ponderado, en virtud de que cada subcaracterística atómica que conforma a la subcaracterística tiene asociado un peso.

Ahora, subiendo al nivel de las características, la calidad de la k-ésima característica también se calcula a través de un promedio ponderado, dado que cada subcaracterística que la conforma tiene un peso asociado representada por la relación siguiente:

$$Q_k = \frac{\sum_{l=1}^L Q_{l_k} \times P_{l_k}}{\sum_{l=1}^L P_{l_k}} ; (c)$$

Para calcular la Calidad total de la HDS, se determina a través de un promedio ponderado de todas las características que conforman al MC-HDS-MDA.

$$Q = \frac{\sum_{k=1}^K Q_k \times P_k}{\sum_{k=1}^K P_k} ; (d)$$

Tomando en cuenta que el Meta-Modelo que se propone en esta investigación, lo constituyen el Modelo del Producto, el Modelo del Proceso y el Modelo Humano, y dentro del Modelo humano que será especificado en una sección más adelante, se encuentran los evaluadores, y son ellos quienes realizan la evaluación de la HDS; se hace necesario examinar las características que poseen los evaluadores, debido a que en toda evaluación humana, existe un grado de subjetividad incluido en las valoraciones que haga con respecto a algún rasgo de la HDS.

6.1.2.3 DETERMINACIÓN DEL ÍNDICE DE CONFIABILIDAD DEL EVALUADOR.

Para medir el **Índice de Confiabilidad del evaluador (ICE)**, que será definido como el grado de confianza que se tendrá con respecto a sus evaluaciones, y que será determinado a través del uso de

dos elementos: el primero, el tiempo de uso de la HDS, que será medido en meses; y el segundo, el número de proyectos realizados utilizando la HDS, en consecuencia se tiene lo siguiente:

En esta investigación se tomarán como elementos ideales a:

- Para el caso del tiempo de uso, 6 meses como mínimo.
- Para el caso de los proyectos realizados, se asumirá como ideal que el evaluador haya realizado 2 proyectos utilizando la HDS.

Por supuesto, lo ideal es tener evaluadores con estas características, pero se tiene que recordar que a la hora de experimentar y posteriormente cuando se ponga el MC-HDS-MDA en uso con el sector productivo, pudieran darse tres casos:

- Que se tengan uno o varios evaluadores con las condiciones ideales.
- Que se tengan uno o varios evaluadores con solo alguna de las condiciones cumplidas.
- Que se tengan uno o varios evaluadores en condiciones no ideales.

Para dar respuesta a todas estas situaciones posibles, el Modelo Matemático propone calcular el Índice de Confiabilidad a cada uno de los evaluadores y de esta forma poder ofrecer al Experto en el Modelo una mejor fundamentación a la hora de tomar decisiones con respecto a una u otra HDS.

A continuación se mostrará la Figura N° 34, donde se ha graficado las diferentes situaciones descritas anteriormente.

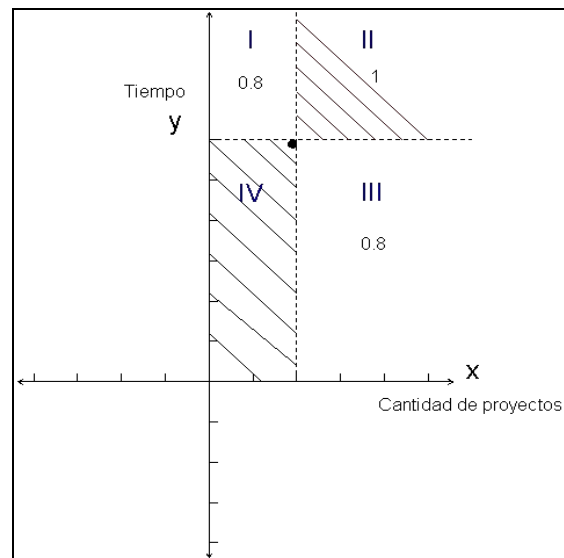


Figura N° 34. Ubicación de los evaluadores en un plano de dos dimensiones

Fuente: Elaboración Propia.

Observe que el punto ideal sería cuando el evaluador tiene 2 proyectos realizados y 6 meses de uso con la HDS, en este caso el Índice de Confiabilidad para ese evaluador es 1, pero si tuviera más de 2 proyectos realizados y más de 6 meses de uso con la HDS, el punto en la gráfica estaría ubicado en el II cuadrante, y en este caso el Índice de Confiabilidad también sería 1. Para los casos en que se tenga a x mayor a 2 o a y mayor a 6, el punto estaría en el primer y tercer cuadrante, donde se ha propuesto que el Índice de Confiabilidad sea de 0.8 en virtud de que, si el evaluador tiene más de dos proyectos en menos de 6 meses, quiere decir, que ha hecho mucho con la HDS en poco tiempo, y esta situación garantiza que hubo un trabajo intensivo con la herramienta en un corto tiempo. Por otro lado, si el evaluador ha usado la herramienta por más de 6 meses pero tiene menos de dos proyectos concluidos, es posible que la razón sea por el tamaño de los proyectos, pero ha usado por un tiempo prolongado a la HDS y eso garantiza un conocimiento importante.

Para esta investigación, el problema realmente se presenta cuando se tiene a uno o más evaluadores con menos de 2 proyectos y menos de 6 meses de uso de la HDS. Para estos casos se propone que el Índice de Confiabilidad sea calculado a través de la fórmula de la **Distancia Euclídeana**, la cual establece el cálculo de la distancia entre dos puntos en un plano de cualquier dimensión.

Para el caso de la distancia entre dos puntos, la fórmula quedaría de la siguiente manera:

$$d(x, y) = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$$

En consecuencia se tiene que:

- Para el caso de tener un evaluador con un Índice de Confiabilidad de 1, se confiará 100% en los resultados que arrojen sus evaluaciones.
- Si se tienen dos o tres evaluadores calificando a una misma HDS, la Q total será el promedio de las Q de cada evaluador, es decir:

$$Q_T = \frac{\sum_{e=1}^E Q_e}{E}$$

Donde, E será el número de evaluadores; e=1,2,3,...,E.

- Si se tienen más de cuatro evaluadores calificando a una misma HDS, se eliminarán los extremos inferior y superior, y se determinará un promedio con las Q intermedias. Esta estrategia se usa para tratar de eliminarle en mayor grado el sesgo que pueda tener la media calculada entre las Q.
- En los casos de tener evaluadores ubicados en el I y III cuadrante, se confiará un 80% en los resultados de sus evaluaciones, e igual que en los casos anteriores si se tienen dos o tres evaluadores se obtendrá un promedio entre las Q, y si se tienen de cuatro evaluadores en adelante, se eliminarán los extremos inferior y superior, y luego se obtiene el promedio con el resto de las Q.
- Si se tienen solo evaluadores ubicados en el IV cuadrante, se obtendrá el Índice de Confiabilidad de acuerdo con la fórmula de la Distancia Euclídeana, de tal manera de escoger al evaluador que tenga la distancia menor con respecto al punto ideal (2 proyectos realizados y 6 meses de uso). El índice arrojará un valor positivo y negativo, lo que significará que pudo haber sobreestimado o subestimado a la HDS en el porcentaje que arroje el cálculo de la Distancia Euclídeana.

Una vez construido este modelo se ha conformado el Modelo del Producto, es decir, el Modelo Teórico y el Modelo Matemático, el próximo paso es proceder a su puesta a prueba que no es más que experimentar con dicho modelo, en otras palabras hacer evaluaciones de herramientas de desarrollo instanciando al MC-HDS-MDA; y para poder hacer esto se deben seguir unos pasos, los cuales se detallarán en la siguiente sección, estos pasos constituyen el Modelo del Proceso.

6.2 MODELO DEL PROCESO

El Modelo del Proceso lo constituyen una serie de pasos que se deben seguir para que las personas que deben conformar el equipo de trabajo instancien el modelo del producto y así poder ofrecer una recomendación lo más objetiva posible al cliente. A continuación se describen cada uno de los pasos que conforman el modelo.

En primer lugar el Cliente establece sus necesidades y prioridades, es decir, los requerimientos indispensables que desee que cubra la HDS y las prioridades con respecto a las preferencias y omisiones que puede aceptar de la herramienta de desarrollo de software. El experto en el Modelo

recibe estas necesidades y prioridades y procede a instanciar al Modelo Teórico, esta instanciación se hace de arriba hacia abajo o (Top-Down).

Este modelo plantea las siguientes sugerencias para realizar la instanciación del MC-HDS-MDA:

- Para el caso de la característica **Cliente**, se deben instanciar todos los aspectos relacionados con los requerimientos del cliente, en este caso puede ser necesario crear métricas nuevas, este modelo por ser Dinámico con respecto al cliente, ofrece esta bondad.
- Con respecto a la Característica **Calidad**, se deben instanciar como mínimo las Subcaracterísticas: Funcionalidad, Portabilidad y Usabilidad, y seleccionar por lo menos el 50% de sus Subcaracterísticas atómicas, y el 50% de las métricas que correspondan.
- Para la característica **Comercialización**, se debe mínimo instanciar las subcaracterísticas: Proceso de Venta, e Indicadores de Soporte, y seleccionar por lo menos el 50% de sus Subcaracterísticas atómicas, y el 50% de las métricas que correspondan.
- Con respecto a la Característica **Ciclo de Vida**, dependerá del tipo de HDS evaluada, por ejemplo si es una HDS MDA, se debe instanciar la subcaracterística MDA y todas las de los procesos medulares, pero si fuera una herramienta solo de requisitos, se instanciaría la Subcaracterística Ingeniería de Requisitos. Cualquier subcaracterística se debe instanciar con el 80% mínimo de subcaracterísticas atómicas el y 80% de sus métricas.

Una vez que el Experto en el Modelo ha realizado la instanciación, el próximo paso es determinar el **Índice de Cumplimiento (IC)** con respecto a las necesidades básicas del cliente, este valor lo calcula utilizando la instanciación del modelo más los pesos o prioridades suministradas por el cliente. **El Índice de Cumplimiento** será definido como el valor máximo u óptimo que deben arrojar todas las métricas asociadas con las necesidades básicas del cliente, tomando en cuenta también los pesos que están representando la importancia de algún elemento con respecto a otro para ese cliente en específico.

Luego debe seleccionar a los evaluadores, para esto tomará en cuenta el **Índice de Confiabilidad del Evaluador** y paralelamente podrá generar los cuestionarios de acuerdo a la instanciación realizada. El experto debe enviar el cuestionario dirigido al proveedor de la HDS, lo más pronto posible, ya que este tendrá mínimo 10 días hábiles para responderlo. Una vez recibido este cuestionario y tener seleccionados a los evaluadores, se le entrega a los evaluadores el cuestionario respondido por el proveedor y el cuestionario relacionado con la Característica Cliente. El evaluador lo recibe y procede a llenarlo, para esta actividad y tomando en cuenta lo delicado de las

respuestas o valoraciones a las métricas que debe hacer el evaluador se sugiere, dar valores a no más de 20 métricas al día, de tal manera de garantizar la máxima objetividad de los resultados que de acuerdo a su percepción tenga de la HDS. Estos valores deben ser revisados por el evaluador y corregir detalles de ser necesario, posteriormente le debe suministrar esta tabulación de métricas al experto.

Para el modelo del proceso estamos llamando Q1 a la calidad relacionada con el ciclo de vida, Q2 a la calidad asociada con característica Calidad, Q3 a la calidad asociada con el cliente y a Q4 a la calidad referida a la comercialización.

Posteriormente el experto con esta información y haciendo uso del modelo matemático calcula la Q3, si esta Q3 es mayor o igual al **Índice de Cumplimiento (IC)** de los requisitos del cliente, el experto envía los cuestionarios de Ciclo de Vida, Calidad y Comercialización, de acuerdo con la instanciación realizada, el evaluador los llena y los revisa, para luego pasarlos al experto nuevamente, que es quien calcula la calidad (Q) de la HDS con respecto a ese evaluador, de tener más de 1 evaluador determina el promedio y luego genera un informe de resultados y recomendaciones, si tiene un solo evaluador, igualmente genera el informe. Finalmente debe entregar el informe al Cliente y realizar las sugerencias respectivas.

Para los casos de que Q3 sea inferior al IC, el experto genera un informe y lo somete a consideración con el cliente.

Para ilustrar estos pasos, observe la Figura N° 35, donde a través de un Diagrama de Actividades de UML, se muestra la dinámica que posee el Modelo del Proceso del MC-HDS-MDA. Se selecciona este Diagrama por ser altamente dinámico y ajustarse perfectamente a los requerimientos del modelo.

6.3 MODELO HUMANO

El Modelo Humano hace referencia al conjunto de personas que deben conformar el equipo de trabajo para someter a evaluación a una herramienta de desarrollo de software. Este equipo de personas ejecutará el modelo del proceso para lograr una correcta aplicación del MC-HDS-MDA y poder obtener los resultados que conducirán a indicarle a un cliente si le conviene o no adoptar una herramienta de desarrollo de software.

De acuerdo con la Figura N° 36, entre las personas que se encuentran dentro del equipo están:

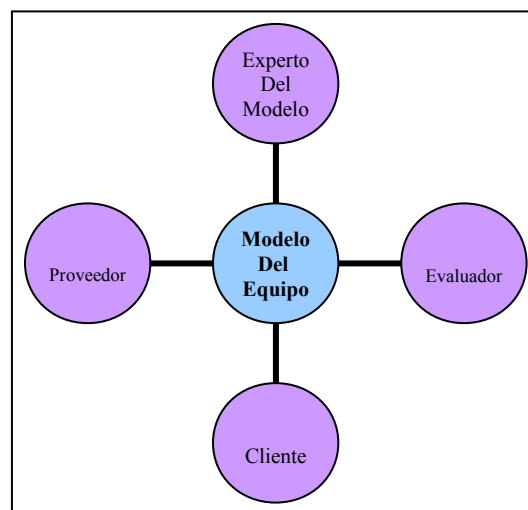


Figura N° 36. Modelo Humano del Meta-Modelo de Calidad para HDS
Fuente: Elaboración Propia.

- **Experto en el Modelo:** Es la persona o personas con el suficiente conocimiento acerca del modelo, su arquitectura, el modelo matemático y su aplicación, que permita hacer la instanciación del modelo de acuerdo a las necesidades planteadas por el cliente.
- **Evaluador:** Puede ser una o varias personas con experiencia en cuanto a desarrollo de aplicaciones de software y en cuanto al uso de la herramienta que será objeto de la evaluación. Se recomienda que sea más de una persona que ejecute la evaluación de las herramientas.
- **Cliente:** Es la persona u organización que desea adoptar una herramienta de desarrollo de software. Este cliente debe plantear sus necesidades y prioridades, es decir, cual será el uso que le pretende dar a la herramienta de desarrollo.

- **Proveedor:** Es la persona u organización que pone a disposición una o unas herramientas de desarrollo de software para que sean sometidas a evaluación.

A continuación se muestran las personas que conforman el Modelo Humano, a través de un Diagrama de Clases, el cual permite observar de una manera más clara la cantidad mínima y máxima de cada una de las personas y las relaciones que se establecen entre ellos.

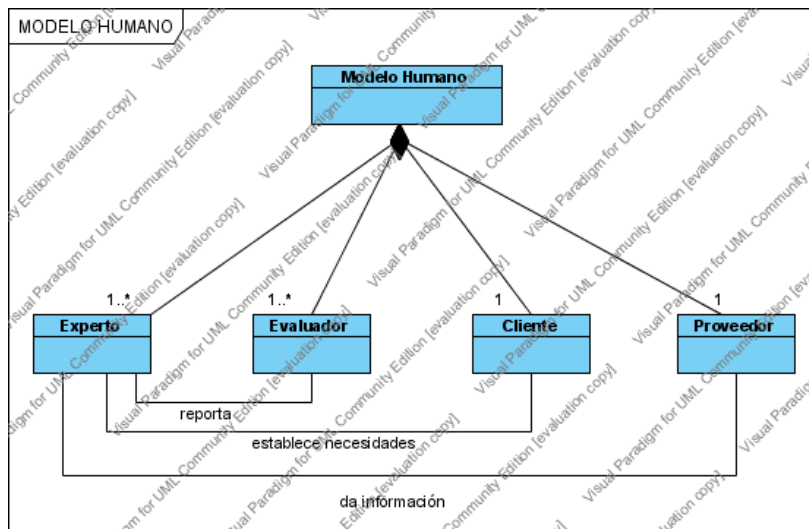


Figura N° 37. Modelo Humano del Meta-Modelo de Calidad para HDS a través de un Diagrama de Clases
Fuente: Elaboración Propia.

Con esta sección queda elaborada la primera versión de la propuesta de un Meta – Modelo de Calidad para evaluar herramientas de desarrollo de software con soporte a MDA, como se mencionó anteriormente lo siguiente es experimentar con este modelo, para poder hacerle los refinamientos necesarios y luego ofrecerlo al sector productivo como una estrategia ideal para someter a consideración HDS que deseen adquirir para la construcción de sus proyectos software. Cabe mencionar que cada vez más las empresas desarrolladoras de software están conscientes de que deben adquirir productos de calidad, y es responsabilidad de la academia en este caso nuestro Decanato de Ciencias y Tecnología brindar herramientas de apoyo que les permitan a las fábricas de software regionales y nacionales adoptar productos acordes a sus necesidades, más aun en estos tiempos donde la globalización pone a la mano un sin número de productos que a la hora de tomar una decisión no es fácil si no se tiene el suficiente conocimiento real de los aspectos que abarcan estas HDS. Con esta propuesta se quiere apoyar al sector productivo a tomar mejores decisiones con respecto a la adquisición de herramientas de desarrollo de software y de esta manera fortalecer los vínculos entre Academia y Fábricas de Software. A continuación se muestran las conclusiones y recomendaciones que se hacen de este trabajo de investigación.

CAPÍTULO VII

CONCLUSIONES Y RECOMENDACIONES

El objetivo de este capítulo es exponer las conclusiones más resaltantes que han sido el producto del desarrollo de esta Investigación. Estas conclusiones permitirán tener una visión más clara y amplia de los objetivos que persiguió este Trabajo. Para iniciar este capítulo se analizan las conclusiones relativas al Meta – Modelo de Calidad para evaluar herramientas de desarrollo de software con soporte a MDA y luego se discuten las recomendaciones que servirán de base para realizar futuras investigaciones en esta área del conocimiento como lo es la Evaluación y Selección de Herramientas de Desarrollo de Software.

6.1. META - MODELO DE CALIDAD PARA EVALUAR HERRAMIENTAS DE DESARROLLO DE SOFTWARE CON SOPORTE A MDA (MC-HDS-MDA).

- El MC-HDS-MDA es considerado un Meta-Modelo porque incluye: el Modelo del Producto, el Modelo de Proceso y el Modelo Humano.

Con respecto al Modelo Teórico.

- El MC-HDS-MDA permite ser más concreto que el MC-CASE en cuanto al tipo de herramienta que se está evaluando, y esto se logra a través de la división entre los procesos medulares y los procesos de apoyo de la característica Ciclo de Vida.
- El Modelo Teórico perteneciente al Modelo del Producto, le agrega a los procesos medulares tres subcaracterísticas atómicas muy importantes dentro del mundo de los procesos de desarrollo como son: las subcaracterísticas; modelado del negocio, ingeniería de requisitos y MDA. Y para cada una de estas subcaracterísticas se proponen las subcaracterísticas atómicas y las métricas correspondientes.
- Las métricas consideradas dentro del grupo de métricas críticas es decir, las que están asociadas íntimamente con las necesidades del usuario, pueden aumentar, dado que de acuerdo a los requerimientos del cliente se pueden adicionar nuevas métricas, por lo que la categoría usuario se considera altamente **dinámica**, lo que conduce a caracterizar el Modelo Teórico como un **Modelo Dinámico**.

- Se decide separar MDA como una subcaracterística aparte en función de que si la herramienta no fuera MDA, penalizaría considerablemente la puntuación si la herramienta fuera solo de diseño o solo de programación. Entonces para no desfavorecer a este tipo de herramientas de desarrollo de software, se crea una nueva Subcaracterística para evaluar todos los aspectos que implique que la herramienta soporte el enfoque MDA.
- Todas las métricas tipo tasa de cobertura que en el MC-CASE estaban distribuidas por todo el modelo, en el MC-HDS-MDA estarán concentradas en la categoría Usuario.
- Todas las métricas del Usuario deben instanciarse al momento de ejecutar una evaluación.
- Durante la construcción del Modelo Teórico debido al nivel de detalle necesitado para lograr su conformación, se hizo necesario hacer nuevamente un Análisis Semántico del MC-CASE, con respecto a las elementos que se mantendrían para el MC-HDS-MDA.

Con respecto al Modelo Matemático.

- El Modelo del Producto propone un modelo matemático que soporta al modelo teórico dándole objetividad a los resultados del MC-HDS-MDA.
- El Modelo Matemático toma en cuenta las prioridades establecidas por el cliente.
- El Modelo Matemático considera las características del evaluador y permite determinar el Índice de Confiabilidad del Evaluador.

Con respecto al Modelo del Proceso y al Modelo Humano.

- La instanciación del MC-HDS-MDA que se realiza durante el Modelo del Proceso es Top-Down.
- El Meta-Modelo propone un Modelo de Proceso, el cual permite de una forma sistemática aplicar el MC-HDS-MDA a una o un conjunto de herramientas de desarrollo de software.
- El Meta-Modelo es un modelo de Calidad **dirigido al cliente**, en virtud de que el cliente no solo establece sus necesidades para adquirir una HDS sino también que establece las prioridades que considera relevantes para sus intereses organizacionales.
- El Meta-Modelo propone un Modelo Humano, donde establece cuales son las personas que deben conformar el equipo que aplicará el MC-HDS-MDA.

6.2. RECOMENDACIONES

Dentro de las recomendaciones que se sugieren para futuras investigaciones están:

- Experimentar con el MC-HDS-MDA, de tal manera de ponerlo a prueba para lograr el refinamiento necesario y obtener la versión final del Meta-Modelo.
- Divulgar todas las investigaciones que en materia de Calidad de herramientas de desarrollo de software se concreten para lograr uno de los objetivos fundamentales de la academia: servir de apoyo a los sectores productivos involucrados en la adopción de nuevas tecnologías, particularmente las HDS.
- Crear un pool de herramientas de desarrollo de software tanto libres como propietarias y así contar con el elemento principal para realizar las evaluaciones con el MC-HDS-MDA.
- Generar una base de datos con la información resultante de las evaluaciones hechas a las herramientas CASE con las que cuenta el Decanato de Ciencias y Tecnología.
- Generar proyectos de investigación-extensión en el Decanato de Ciencias y Tecnología que permitan asesorar al sector productivo de la región, en la evaluación de HDS.