UNIVERSIDAD CATÓLICA ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

AMBIENTE DE EVALUACIÓN DE ARQUITECTURAS DE SOFTWARE

Apéndices y Anexos

TRABAJO ESPECIAL DE GRADO

presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

como parte de los requisitos para optar por el título de

INGENIERO EN INFORMÁTICA

REALIZADO POR Marín Carvajal, Andreína

PROFESOR GUÍA Grimán, Anna Cecilia

FECHA Caracas, Julio del 2003

APÉNDICE A – Comparación de Métodos

Estos fueron los métodos de evaluación estudiados según las características mostradas

Tabla A.1. Características de Bosch. Fuente: Elaboración Propia.

Metodo	Estructura	Documentación	Arquitectura	Calidad	Técnicas de Evaluación	Otros
Método de Bosch	El hecho de trabajar para una arquitectura en específico, ayuda en la evaluación de un sistema ya desarrollado. Debería ser más explícito en la proposición de roles al equipo evaluador y en la especificación de sus entregables, dado que no existe un patrón formal de ellos.	Su documentación es muy completa y se pueden conseguir muchas referencias de su aplicación.	Es un método de evaluación que busca la transformación de una arquitectura inicial en una ideal. Solo permite trabajar con una arquitectura candidata, la que se va a transformar.	Es importante, la formalización de la calidad bajo un modelo de calidad específico o un estándar. Ello se traduciría en la incorporación de un refinamiento y la propuesta de unas métricas de evaluación directamente relacionadas. En general, no es recomendable utilizarlo para evaluar muchos o todos los atributos de calidad, puesto que no resultaría costo efectivo realizar numerosas iteraciones.	Apropiado para la evaluación de atributos de calidad de desarrollo, es decir los atributos de calidad no observables en tiempo de ejecución (evaluación basada en escenarios), para evaluar atributos de calidad observables en tiempo de ejecución (evaluación basada en prototipos y simulación) y para valuar atributos de calidad operacionales. Se puede evaluar el comportamiento de la arquitectura en varios escenarios o circunstancias, evaluación basada en escenarios, evaluación basada en simulación y prototipos). Puede proporcionar valor agregado mediante la evaluación basada en la experiencia.	Es importante que se tengan presente el análisis de los riesgos y el impacto de la transformación de la arquitectura. Si bien el resultado puede mejorar los requerimientos de calidad, también se pueden obtener resultados desfavorables en otro sentido. Su uso está orientado a una persona experta y del área de la arquitectura de software, de allí que pueda resultar difícil para personas que no dominan el área.

Tabla A.2. Características de ABDM. Fuente: Elaboración Propia.

Metodo	Estructura	Documentación	Arquitectura	Calidad	Técnicas de Evaluación	Otros
Método ABD	Descompone el sistema en aplicación de infraestructura, subsistemas conceptuales, componentes conceptuales y componentes concretos para el manejo y dominio del problema.. Refina los casos de uso verificando la calidad.	Su documentación es completa y se pueden conseguir muchas referencias de su aplicación.	Provee una estructura para la arquitectura conceptual del sistema, que representa una base para lograr la calidad. Solo permite trabajar con una arquitectura ya que se va modelando a lo largo del método.	A pesar que usan los requerimientos funcionales combinados con los elementos de calidad, no se apoya bajo ningún modelo de calidad capaz de calificar el atributo.	Basado en el análisis de escenarios. La efectividad de la técnica depende en gran parte de lo representativo de los escenarios (evaluación basada en escenarios). Hay que ser cuidadosos al definir los escenarios, aunque este método toma como entrada una lista de requerimientos funcionales, de calidad y de negocios, y reglas, que lo facilita.	Usa herramientas llamadas "templates" que se van refinando conforme avanza el diseño de la arquitectura. A pesar de dejar plasmado observaciones importantes en cuanto a la aplicación de la calidad, no tiene especificación alguna para hacerlo.

Tabla A.3. Características de ATAM. Fuente: Elaboración Propia.

Metodo	Estructura	Documentación	Arquitectura	Calidad	Técnicas de Evaluación	Otros
Método ATAM	Su orientación, es la métrica de evaluación y aunque no la propone, está enfocada siempre a un nivel arquitectónico. Es un método de evaluación que busca la selección de una arquitectura entre varias candidatas, por lo tanto es necesario proponer al menos dos arquitecturas	Su documentación es clara y consistente	Revela como la arquitectura satisface metas particulares de calidad y además provee la interacción de las metas de calidad entre sí. Permite trabajar con más de una arquitectura candidata inicial, incluso con más de dos.	En cuanto a la evaluación de características de calidad, debería tomar en conjunto la evaluación de las mismas, por ser ellas dependientes entre sí. No intenta predecir el comportamiento de los atributos de calidad, sino que intenta aprender dónde un atributo de interés es afectado por decisiones de diseño arquitecturales. Se debe formalizar la estandarización de los objetivos de calidad.	Basado en escenarios. Usa la herramienta "utility tree" para la identificación de los objetivos de calidad, aplicación de la técnica de escenarios y análisis de los resultados. Debido a la utilización de escenarios, depende un poco de cómo estos sean definidos, pero se trata de equilibrar este hecho con la asignación de prioridades.	Incluye como equipo evaluador a los stakeholders del sistema, es decir el equipo evaluador es más completo y diverso. Orienta en la toma de decisiones a la hora de seleccionar una arquitectura como ideal, para ello analiza los puntos de sensibilidad y los riesgos involucrados

Tabla A.4. Características de MEL. Fuente: Elaboración Propia.

Método	Estructura	Documentación	Arquitectura	Calidad	Técnicas de Evaluación	Otros
Método MEL	Énfasis en revisar todo tipo de requerimiento, funcional y no funcional. Ello hace que la búsqueda de una calidad se oriente también hacia la satisfacción del cliente, aunque la participación de él sea explícita en el proceso.	Método reciente, sin embargo se pueden encontrar algunas aplicaciones con el mismo.	Permite trabajar con más de una arquitectura candidata inicial, dado que el método siempre se refiere a dos arquitecturas.	Su principal ventaja es que usa un estándar de calidad reconocido internacionalmente, ISO En cuanto a las métricas de evaluación, se caracteriza por proponer unas métricas a la hora de evaluar de las características de calidad, métricas no propuestas por ninguno de los otros métodos. Esto contribuye a que exista una mayor consistencia en el método de evaluación.	Las herramientas que usa para la entrada, salida y comparación de la evaluación; son fáciles de usar, legibles y eficientes. Esto se traduce en ahorros de costos de todo tipo: investigación, asesoramiento, personal, tiempo, recursos de apoyo como software o hardware. En conclusión, es un método económico. Basado en el análisis de los casos de uso del sistema.	Es el único de los métodos que usa un lenguaje formal para la descripción de la arquitectura.

APÉNDICE B – Comparación de los ADLs

Después de estudiar los ADLs, se puntualizaron ciertas características para establecer comparaciones.

Tabla B.1. Comparación de los ADLs según sus características. Fuente: (Vargas, 2002)

ADL	Estilos Arquitectónicos	Entidades	Ambiente	Simulación	Complejidad	Semejanza con	Generación de Código	Otras
ACME	Si	Componentes, Conectores	Linux y SunOS	No	Baja	N/A	No	N/A
Aesop	Si (combinación)	Componentes, Conectores	Linux y SunOS	No	Baja	ACME Wright	Limitada	OO
Armani	Si	Componentes, Conectores		No	Baja	ACME Aesop	N/A	N/A
C2	Si	Componentes		No	N/A	N/A	N/A	Ambiente Gráfico
Darwin	N/A	Componentes		No	Alta	N/A	N/A	N/A
MetaH	N/A	N/A	Windows		Normal	N/A	N/A	Reuso de Modelos
RAPIDE	N/A	Componentes	Linux y SunOS	Si	Normal	N/A	Escasa	Ambiente Gráfico. Carencia de Componentes
UniCon	Si	Componentes, Conectores		No	N/A	N/A	Si	N/A
Wright	Si	ComponentesConectores	Linux y SunOS	No	Baja	ACME Aesop	No	N/A

APÉNDICE C - Casos de Estudio

Caso de Estudio 1. Sistema de Gestión del Conocimiento para Proyectos de Investigación. (Aplicación de A.T.A.M.)

El objetivo de esta investigación fue el desarrollo de la primera versión de un Sistema de Gestión del Conocimiento para Proyectos de Investigación. Para el desarrollo del prototipo se utilizaron dos metodologías: una propuesta metodológica sistémica para Gestión del Conocimiento, para el modelamiento y entendimiento del negocio; y dentro de ésta, R.U.P. como metodología de desarrollo. En la fase de Elaboración, específicamente en la selección de la arquitectura, se utilizó un Método para la Evaluación de Arquitecturas de Software (A.T.A.M), lo que requirió identificar atributos de calidad para este tipo de sistemas mediante la instanciación del Modelo de Calidad del Producto propuesto por Ortega (2000), pero tomando la arquitectura como producto a evaluar. De las dos arquitecturas propuestas, la seleccionada para el sistema se caracteriza por ser multicapa y estar orientada a objetos, con los patrones de diseño.

A continuación se presenta el modelo conceptual elaborado para este caso de estudio.

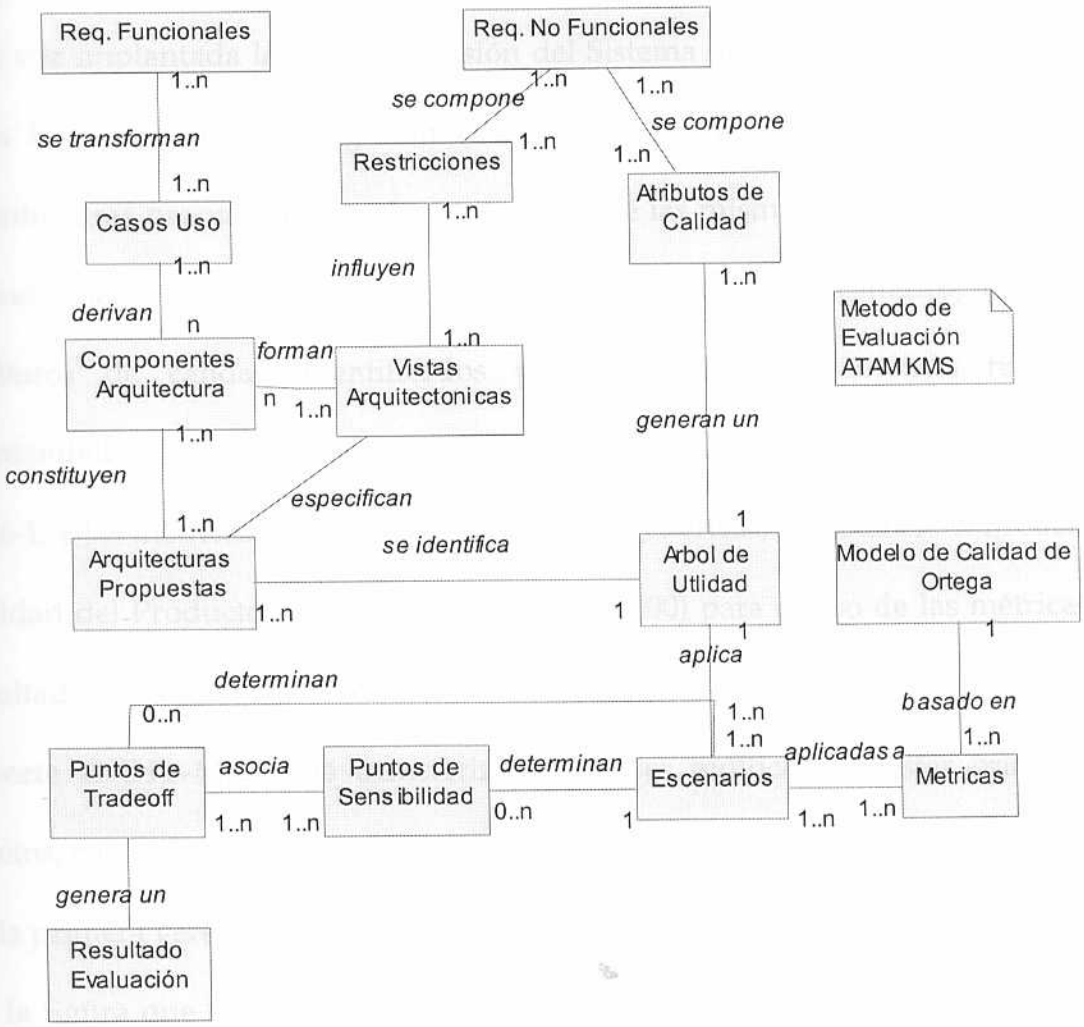

Figura C.1. Modelo Conceptual del KMS bajo el Método de Evaluación A.T.A.M. Fuente: Elaboración Propia.

Caso de Estudio 2. Sistema de Gestión del Conocimiento para Proyectos de Investigación. (Aplicación de M.E.L.)

Una vez implantada la primera versión del Sistema de Gestión del Conocimiento para Proyectos de Investigación (DID-KMS), se aplicó otra evaluación a las arquitecturas propuestas para la verificación de las mismas. Esta vez, se utilizó el método de evaluación de Losavio (M.E.L.), basado en escenarios, donde los atributos de calidad identificados para este tipo de sistemas fueron la mantenibilidad, la fiabilidad y la eficiencia. Dado que M.E.L. usa el Estándar ISO 9126-1, y las métricas que provee éste son muy genéricas, se instanció el Modelo de Calidad del Producto propuesto por Ortega (2000) para el uso de las métricas. El resultado de esta evaluación corroboró que la arquitectura que ofrece mayor soporte al DID-MKS está caracterizada por ser multicapa y estar orientada a objetos, con los patrones de diseño, siendo ésta, la misma arquitectura seleccionada en la primera versión de la evaluación (Caso de Estudio 1)

En la figura que se muestra a continuación, se encuentra representado el modelo conceptual referente al caso de estudio 2.

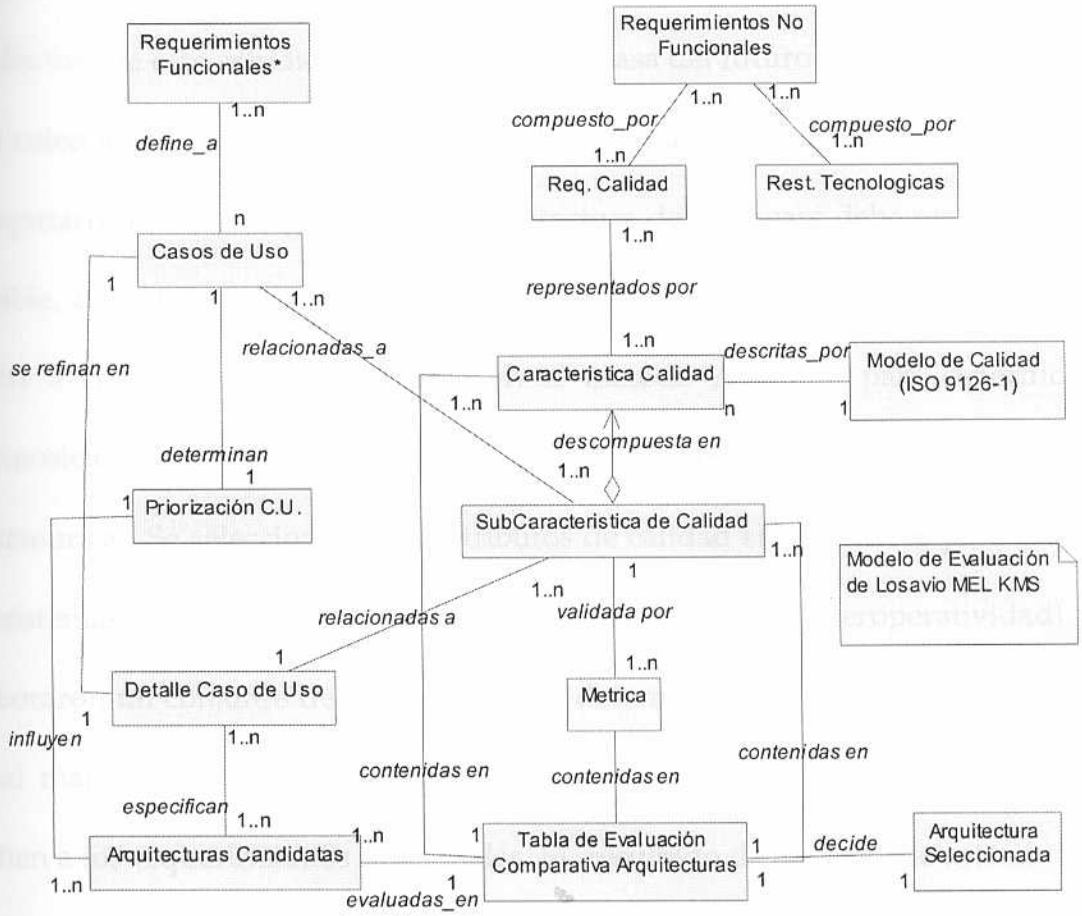

Figura C.2. Modelo Conceptual del KMS bajo el Método de Evaluación M.E.L.
Fuente: Elaboración Propia.

Caso de Estudio 3. Sistema para la Casa del Futuro . (Aplicación de A.B.D.M.)

El objetivo de este estudio fue el diseño de la casa del futuro donde se asume tener una colección de dispositivos dentro de la casa que sean controlados por una red computarizada. Para lograr esto, la arquitectura del software debe ser extensible y flexible, debe tener alta seguridad, alto rendimiento, y alta disponibilidad. Para llevar a cabo este diseño, se baso en el método A.D.B.M. para especificar la composición de la arquitectura considerando que esta red incluía comunicación inalámbrica. Se seleccionaron los atributos de calidad significativos para este tipo de sistemas como usabilidad, fiabilidad y funcionalidad (interoperatividad) y se elaboraron un conjunto de escenarios para determinar un estilo arquitectónico. De igual manera, se elaboraron un conjunto de grupos con responsabilidades que atañen a los requerimientos funcionales. El resultado de aplicar este método fue el diseño de una arquitectura que contempla aspectos de funcionamiento (interoperatividad), fiabilidad y usabilidad mediante la verificación de escenarios de calidad refinados y un conjunto de plantillas (templates) del software que incluyen una descripción detallada de los componentes que obran recíprocamente con servicios compartidos y también incluyen las responsabilidades de los grupos de componentes.

El modelo conceptual del caso de estudio la casa del futuro se encuentra en la figura a continuación.

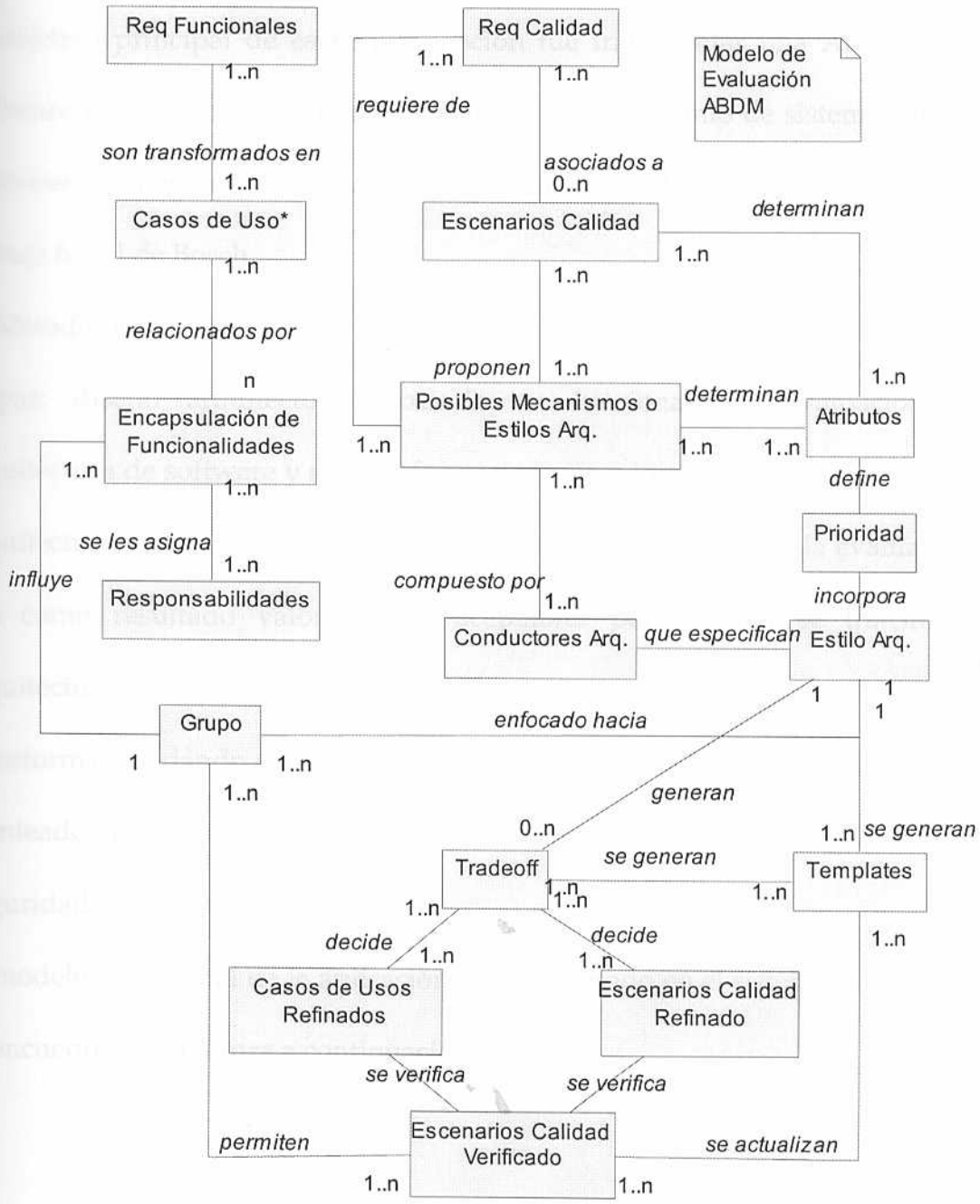

Figura C.3. Modelo Conceptual de la Casa del Futuro el Método de Evaluación A.B.D.M. Fuente: Elaboración Propia.

Caso de Estudio 4. Sistema de Emergencias, Call Center. (Aplicación de Bosch)

El objetivo principal de esta investigación fue transformar una Arquitectura de Software para fomentar atributos de calidad en el dominio de sistemas de (CRM) *Costumer Relationship Management.* El método utilizado para la realización de ese trabajo fue el de Bosch.

El Método para el diseño de Arquitecturas de Software Bosch, se desarrolla en tres etapas: diseño arquitectónico basado en funcionalidad, evaluación de la arquitectura de software y su transformación. Para lograr expresar formalmente la arquitectura fue necesario definir perfiles y escenarios; se aplicó la evaluación que dio como resultado valores poco aceptables por lo cual se transformó la arquitectura. Se iteró la evaluación con respecto a los requerimientos transformados, dando como resultado una arquitectura mejorada para el dominio planteado, propiciada por los atributos de calidad: Fiabilidad, Usabilidad, y Seguridad.

El modelo conceptual de la aplicación de este método en el sistema del Call Center se encuentra en la figura a continuación.

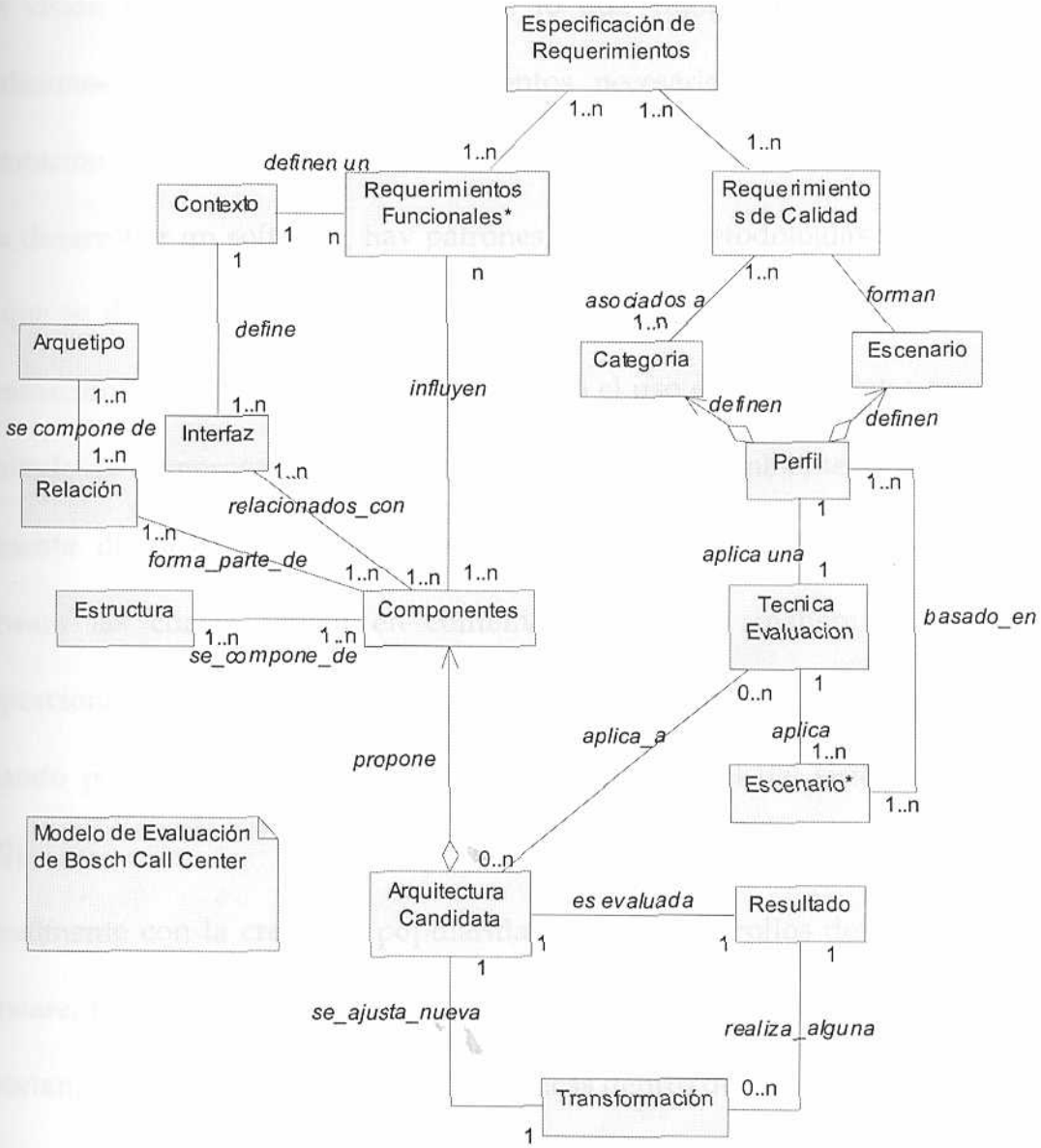

Figura C.4. Modelo Conceptual del Call Center bajo el Método de Evaluación de Bosch. Fuente: Elaboración Propia.

APENDICE D - Visión del Proyecto

Esta visión representa una sinapsis de lo que engloba el proyecto en sí, las condiciones iniciales y los requerimientos necesarios para llevar a cabo su elaboración.

Para desarrollar un software, hay patrones, normas, metodologías y estrategias en las que se deben apoyar los programadores para sustentar de manera segura la culminación exitosa de su producto, así como el uso de herramientas tecnológicas resultado de numerosas investigaciones que facilitan el ambiente de desarrollo. Un ambiente de desarrollo es con un conjunto de herramientas de hardware y software las cuales actúan en combinación de una manera integrada para proporcionar soporte a los procesos de desarrollo desde la especificación inicial pasando por las pruebas hasta la implementación total del sistema (Somerville, 2002).

Actualmente con la creciente popularidad de los desarrollos de los sistemas de software, ha surgido la necesidad y el interés por mejorar las herramientas que lo soportan, e incluso integrar a diversas técnicas dentro de cada una de ellas. Para la situación de la evaluación arquitectónica, esta necesidad es más apremiante aún, por esta razón se hace necesario el desarrollo de un ambiente que permita evaluar la arquitectura de software basándose en una especificación previa donde se incluyan los modelos de calidad, los métodos y las técnicas de evaluación,

los lenguajes de descripción de arquitecturas y su especificación.

Con la finalidad de apoyar el proceso de diseño de los sistemas, se intenta recurrir a herramientas que permitan integrar varios procesos en uno solo bajo el esquema de un ambiente de desarrollo de software.

Una lista detallada de los requerimientos a cumplir se encuentra en el Apéndice E - *Requerimientos Funcionales.*

Ante estos requerimientos, se propone el desarrollo de un ambiente de desarrollo permita evaluar bajo diferentes aspectos la arquitectura de software. Dado a que el ambiente está orientado para el uso de un solo actor, "el arquitecto" no se crearon perfiles de usuarios. El arquitecto debe ser capaz de manipular el ambiente, ingresar nuevos elementos y darle la funcionalidad respectiva.

El sistema permitirá realizar principalmente las siguientes actividades:

- **Administrar Modelos de Especificación de la Calidad**: en esta actividad los modelos de especificación de la calidad se pueden ingresar, modificar o eliminar dentro del ambiente. Cada vez que se ingrese, se le dará las características específicas y funcionalidades respectivas a cada elemento propio, de cada modelo.
- **Administrar Métodos de Evaluación de Arquitecturas:** al igual que en el caso anterior, el arquitecto puede incluir nuevos métodos, modificarlos o eliminarlos. Al incluir un método nuevo, debe darle la funcionalidad pertinente.

- **Administrar Técnicas de Evaluación de Arquitecturas:** en este caso, el arquitecto puede incluir, modificar o eliminar nuevas técnicas de evaluación. Siempre que se incluya un elemento nuevo al ambiente, se le debe dar su funcionalidad correspondiente.
- **Administrar Lenguajes Formales:** los ADL's deben poder interactuar con el ambiente, así como poder modificarlos y/o eliminarlos.
- **Generar Reporte:** una vez que se haya hecho la evaluación, se puede generar un reporte que soporte todos los procesos que se han realizado en el ambiente. Este reporte debe especificar si la arquitectura evaluada es idónea para el sistema en cuestión.
- **Especificar Arquitecturas:** le permite al arquitecto especificar la arquitectura para el sistema. Al especificar una arquitectura, se le dará características típicas funcionales y no funcionales de ésta.
- **Evaluar Arquitectura:** este módulo es el más importante ya que va a permitir al arquitecto evaluar integralmente, con esta herramienta, la arquitectura propuesta para el sistema. La evaluación estará basada en modelos de especificación de la calidad, métodos y técnicas de evaluación y lenguajes formales que serán parte del ambiente. El mismo resultado será arrojado por el reporte.

Además de las actividades que el sistema debe permitir realizar se encuentran

unos requerimientos complementarios que no pueden capturarse en el Modelo de Casos de Uso y se mencionan a continuación:

- **Mantenibilidad:** que acepte futuras actualizaciones para darle al ambiente otras funcionalidades y además debe poseer una buena documentación del diseño.
- **Interoperatividad:** permite que el ambiente trabaje con sistemas externos e independientes al mismo.
- **Fiabilidad:** el sistema debe ser confiable, con gran tolerancia a fallos. Se debe establecer medidas cuantitativas, tales como, tiempo entre fallas, tiempo de recuperación o defectos por cada cien líneas de código.

En el Apéndice F - *Requerimientos Complementarios* se muestran detallados cada uno.

Este ambiente de evaluación de arquitecturas involucra varias herramientas de desarrollo, como Java y SQL Server ya que éstas pueden operar bajo los ambientes Windows y Linux, y la herramienta de apoyo Rational Rose v.2002.

APENDICE E - Requerimientos Funcionales

Luego de una exhaustiva investigación se determinaron aspectos funcionales con los que el ambiente debía contar. A continuación, se presenta una lista de los requerimientos identificados para AMEVA.

Tabla E.1. Requerimientos Funcionales R.1. del Sistema. Fuente: Elaboración Propia.

Número:	R.1
Nombre:	Incluir Modelos de Especificación de la Calidad
Descripción:	Debe permitir al arquitecto incluir un nuevo modelo de especificación de la calidad al ambiente
Tipo:	Funcional
Versión:	1.0
Condición:	Debe

Tabla E.2. Requerimientos Funcionales R.2. del Sistema. Fuente: Elaboración Propia.

Número:	R.2
Nombre:	Modificar Modelos de Especificación de la Calidad
Descripción:	Debe permitir al arquitecto modificar un modelo de especificación de la calidad existente al ambiente
Tipo:	Funcional
Versión:	1.0
Condición:	Debe

Tabla E.3. Requerimientos Funcionales R.3. del Sistema. Fuente: Elaboración Propia.

Número:	R.3
Nombre:	Eliminar Modelos de Especificación de la Calidad
Descripción:	Debe permitir al arquitecto eliminar un modelo de especificación de la calidad al ambiente
Tipo:	Funcional
Versión:	1.0
Condición:	Debe

Tabla E.4. Requerimientos Funcionales R.5. del Sistema. Fuente: Elaboración Propia.

Número:	R.5
Nombre:	Modificar de Métodos de Evaluación
Descripción:	Debe permitir al arquitecto modificar un método de evaluación de arquitecturas de software existente al ambiente
Tipo:	Funcional
Versión:	1.0
Condición:	Debe

Tabla E.5. Requerimientos Funcionales R.4. del Sistema. Fuente: Elaboración Propia

Número:	R.4
Nombre:	Incluir de Métodos de Evaluación
Descripción:	Debe permitir al arquitecto incluir un nuevo método de evaluación de arquitecturas de software
Tipo:	Funcional
Versión:	1.0
Condición:	Debe

Tabla E.6. Requerimientos Funcionales R.6. del Sistema. Fuente: Elaboración Propia.

Número:	R.6
Nombre:	Eliminar Métodos de Evaluación
Descripción:	Debe permitir al arquitecto eliminar un método de evaluación de arquitecturas de software al ambiente
Tipo:	Funcional
Versión:	1.0
Condición:	Debe

Tabla E.7. Requerimientos Funcionales R.7. del Sistema. Fuente: Elaboración Propia.

Número:	R.7
Nombre:	Incluir Técnicas de Evaluación
Descripción:	Debe permitir al arquitecto incluir técnicas de evaluación de arquitecturas de software al ambiente
Tipo:	Funcional
Versión:	1.0
Condición:	Debe

Tabla E.8. Requerimientos Funcionales R.8. del Sistema. Fuente: Elaboración Propia.

Número:	R.8
Nombre:	Modificar Técnicas de Evaluación
Descripción:	Debe permitir al arquitecto modificar técnicas de evaluación de arquitecturas de software al ambiente
Tipo:	Funcional
Versión:	1.0
Condición:	Debe

Tabla E.9. Requerimientos Funcionales R.9. del Sistema. Fuente: Elaboración Propia.

Número:	R.9
Nombre:	Eliminar Técnicas de Evaluación
Descripción:	Debe permitir al arquitecto eliminar técnicas de evaluación de arquitecturas de software al ambiente
Tipo:	Funcional
Versión:	1.0
Condición:	Debe

Tabla E.10. Requerimientos Funcionales R.10. del Sistema. Fuente: Elaboración Propia.

Número:	R.10
Nombre:	Aplicación de ADL
Descripción:	Define el lenguaje de descripción de arquitecturas de software
Tipo:	Funcional
Versión:	1.0
Condición:	Debe

Tabla E.11. Requerimientos Funcionales R.11. del Sistema. Fuente: Elaboración Propia.

Número:	R.11
Nombre:	Especificación de Arquitecturas de Software
Descripción:	Debe permitir al arquitecto especificar gráficamente una arquitectura con sus componentes y funcionalidades
Tipo:	Funcional
Versión:	1.0
Condición:	Debe

Tabla E.12. Requerimientos Funcionales R.12. del Sistema. Fuente: Elaboración Propia.

Número:	R.12
Nombre:	Generación de Documentación
Descripción:	Muestra el resultado obtenido de la evaluación del diseño
Tipo:	Funcional
Versión:	1.0
Condición:	Debe

Tabla E.13. Requerimientos Funcionales R.13. del Sistema. Fuente: Elaboración Propia.

Número:	R.13
Nombre:	Evaluación de Arquitecturas de Software
Descripción:	Debe permitir al arquitecto evaluar la arquitectura propuesta mediante el uso de modelos de especificación de la calidad, métodos de evaluación, técnicas de evaluación y lenguajes de descripción de arquitecturas
Tipo:	Funcional
Versión:	1.0
Condición:	Debe

APENDICE F - Requerimientos Complementarios

Al igual que los requerimientos funcionales, también se determinaron ciertos aspectos complementarios para el ambiente. A continuación, se presenta una lista de los requerimientos complementarios que fueron identificados para AMEVA.

Tabla F.1. Requerimientos Complementarios RC.1 del Sistema. Fuente: Elaboración Propia

Número:	RC.1
Requerimiento:	Un cambio no debe tomar más de 16 Horas/Profesional
Descripción:	Debe poseer una buena documentación, además de un buen diseño. También es determinante la mantenibilidad para actualizaciones al momento de agregarle nuevas funcionalidades o simplemente eliminarlas, debe tener todas sus especificaciones de diseño.
Atributo:	Mantenibilidad
Condición:	Debe

Tabla F.2. Requerimientos Complementarios RC.2 del Sistema. Fuente: Elaboración Propia

Número:	RC.2
Requerimiento:	Aplicación de un ADL
Descripción:	Debe permitir acoplar otro sistema al ambiente y trabajar conjuntamente, en este caso, la posibilidad de interactuar con un ADL comercial.
Atributo:	Interoperatividad
Condición:	Debe

Tabla F.3. Requerimientos Complementarios RC.3 del Sistema. Fuente: Elaboración Propia

Número:	RC.3
Requerimiento:	Disponibilidad de 150 Horas/mes
Descripción:	El sistema debe ser confiable, no se debe permitir que tenga fallos contínuos, porque esto provocaría dudas por parte del usuario, por lo tanto se debe garantizar un bajo promedio de fallas.
Atributo:	Fiabilidad
Condición:	Debe

Tabla F.4. Requerimientos Complementarios RC.4 del Sistema. Fuente: Elaboración Propia

Número:	RC.4
Requerimiento:	Tiempo de recuperación de 10 segundos.
Descripción:	Debe establecer medidas cuantitativas, tales como, tiempo entre fallas o defectos por cada cien líneas de código.
Atributo:	Fiabilidad
Condición:	Debe

APÉNDICE G - Plan de Proyecto

En el plan de proyecto se definieron las actividades propias para cada etapa, y la duración de las mismas. A continuación se presenta el plan de proyecto definido para la elaboración de AMEVA.

Fase de Concepción

Tabla G.1. Plan de Proyecto de la Fase de Comienzo. Fuente: Elaboración Propia

Actividades	Semanas	Iteraciones
Conocer la Cultura Organizacional de LISI	2	1
Estudiar Actividades y Procesos de LISI	1	1
Elaborar Plan de Proyecto	1	1
Estudiar y Construir el Modelo de Negocio	1	1
Establecer los requerimientos, características y restricciones del Ambiente	1	1
Realizar lista inicial de riesgos y evaluarlos	1	1
Elaborar del Modelo de Casos de Uso Inicial con sus Actores	2	1
Elaborar el Glosario	2	1
Establecer estándares de interfaz y requerimientos complementarios	2	1

Fase de Elaboración

Tabla G.2. Plan de Proyecto de la Fase de Elaboración. Fuente: Elaboración Propia

Actividades	Semanas	Iteraciones
Actualización del plan de iteración	1	1
Generar una lista revisada de riesgos	2	1
Ajustar el modelo de casos de uso	2	1,2
Describir la arquitectura del software	4	1,2
Definir modelos conceptuales para diseño general y detallado	4	1,2
Validar el modelo conceptual con casos de estudio	3	1

Plan de Actividades de la Fase de Construcción y Fase de Transición

El alcance de este trabajo especial de grado, llega hasta su diseño. Como actividad adicional, se presentan los estudios de casos para validar que el ambiente contenga los requerimientos mínimos necesarios.

APENDICE H - Riesgos Asociados

Los riesgos identificados fueron estudiados cada uno y se diseñó una estrategia de mitigación y un plan de contingencia. A continuación, se presenta cada riesgo detallado.

Tabla H.1. Detalle de Riesgos Requerimientos Inmaduros. Fuente: Elaboración Propia.

Requerimientos Inmaduros	
Magnitud:	Alta
Factor de Riesgo:	4
Descripción:	Debido a la cantidad de componentes y especificaciones que debe tener el sistema, es necesario conocer a cabalidad los requerimientos del mismo.
Impacto:	Omisión de componentes o parte de ellos.
Indicadores:	Información incompleta a la hora de hacer una consulta.
Estrategia de Mitigación:	Estudiar a fondo cada elemento, y establecer un mínimo de información por cada requerimiento.
Plan de Contingencia:	Dar prioridades a los requerimientos funcionales y a los requerimientos complementarios.

Tabla H.2. Detalle de Riesgos Requerimientos Mal Entendidos. Fuente: Elaboración Propia.

Requerimientos Mal Entendidos	
Magnitud:	Alta
Factor de Riesgo:	4
Descripción:	Debido a la cantidad de componentes y especificaciones que debe tener el sistema, es frecuente confundir los requerimientos.
Impacto:	Mala concepción de requerimientos, como consecuencia rediseñar y ajustar o ampliar el cronograma.
Indicadores:	En el momento de la implantación, funcionalidades no correspondientes.
Estrategia de Mitigación:	Durante el diseño, establecer puntos clave de cada requerimiento y verificar su correspondencia.
Plan de Contingencia:	Corroborar la información levantada con casos de estudios.

Tabla H.3. Detalle de Riesgos Alcance Inestable del Proyecto. Fuente: Elaboración Propia.

Alcance Inestable del Proyecto	
Magnitud:	Alta
Factor de Riesgo:	4
Descripción:	Cada requerimiento funcional se convierte en un caso de uso, el cual determina parte del alcance, pero para los requerimientos complementarios no es tan sencillo definir alcance.
Impacto:	Etapas mal concebidas durante la elaboración del proyecto.
Indicadores:	Al momento de la implantación
Estrategia de Mitigación:	Definir los hitos de cada etapa.

Plan de Contingencia:	Entregar un plan de proyecto por escrito, donde se especifiquen las actividades más importantes para la puesta en marcha del sistema.

Tabla H.4. Detalle de Riesgo Mala Estimación del Tiempo. Fuente: Elaboración Propia.

Mala Estimación del Tiempo	
Magnitud:	Alta
Factor de Riesgo:	4
Descripción:	Debido a los riesgos anteriores y más aún por aquellos que no están, se pueden presentar inconvenientes que retrasen las fases del desarrollo.
Impacto:	Ajustes o ampliaciones en el cronograma.
Indicadores:	Retrasos constantes en la entrega de partes o componentes relativos al sistema.
Estrategia de Mitigación:	Dejar un espacio de holgura entre las fases del desarrollo.
Plan de Contingencia:	Realizar un análisis de los tiempos reales de entrega y basar la nueva planificación en base a estos tiempos con un mínimo de holguras.

Tabla H.5. Detalle de los Riesgo Tecnología Inestable. Fuente: Elaboración Propia.

Tecnología Inestable	
Magnitud:	Moderado
Factor de Riesgo:	3
Descripción:	La tecnología propuesta no es lo suficientemente madura para llevar a cabo el ambiente. El ambiente debe ser interoperable por lo tanto las herramientas deben ser compatibles con los sistemas externos con que debe interactuar.
Impacto:	Ajustes o ampliaciones en el cronograma.
Indicadores:	Dificultad de implementar en ciertas actividades importantes del sistema. Independencia en el funcionamiento de módulos o componentes originalmente integrados.
Estrategia de Mitigación:	Se deben realizar pruebas donde se compruebe la interoperatividad del ambiente con los otros sistemas (previamente seleccionados).
Plan de Contingencia:	Evaluar herramientas similares y adquirir la seleccionada. En caso de tener una parte significativamente implementada, se deben migrar los datos o diseños a las nuevas herramientas.

Tabla H.6. Detalle de Riesgo Magnitud del Proyecto. Fuente: Elaboración Propia.

Magnitud del Proyecto	
Magnitud:	Moderada
Factor de Riesgo:	3
Descripción:	Dada la cantidad de requerimientos, es necesario hacer una abstracción de cada uno y delimitarlos.
Impacto:	Ajustes, ampliación o reducción de requerimientos. Ajustes o ampliación en el cronograma.
Indicadores:	Sobre la marcha, surgen más actividades de las que están programadas.
Estrategia de Mitigación:	Definir el proyecto por etapas y darle un alcance a cada una.
Plan de Contingencia:	Definición clara del alcance. Esto se logrará con los riesgos anteriores bien definidos.

APENDICE I - Caso del Negocio

En el Caso del Negocio se proporciona la descripción del producto, los objetivos de éste, y las restricciones del proyecto. Además se presenta el contexto en el que se desarrolla el producto y su pronóstico financiero. Adicionalmente, se encuentran los modelos de caso de usos del negocio y objeto del negocio que ayuda al lector involucrarse con la cultura organizacional.

Descripción del Producto

El producto a desarrollar es un ambiente, en el cual se evalúan aspectos inherentes a la arquitectura de software. Los usuarios podrán especificar el diseño de las arquitecturas de software y evaluar su calidad con respecto al sistema que se esté proponiendo. Está orientado especialmente a desarrolladores de software. Idealmente, se sugiere que en la fase de diseño de un sistema se evalúe la arquitectura a usar, para que esta propicie los requerimientos de calidad esperados en el sistema. Este ambiente provee al desarrollador una serie de opciones ya que las evaluaciones se podrán realizar por métodos y técnicas de evaluación, lenguajes formales y modelos de especificación de la calidad. El usuario de este ambiente puede especificar, mediante la diagramación, la arquitectura deseada y a su vez, emitir la documentación asociada al método de evaluación empleado.

Contexto del Negocio

El Laboratorio de Investigación en Sistemas de Información (LISI) es un grupo de investigación de la sección de Sistemas de Información y Gestión del Departamento de Procesos y Sistemas de la Universidad Simón Bolívar, cuyo interés principal es el estudio de las tecnologías que apoyan el Proceso de Desarrollo de los Sistemas de Información.

LISI se plantea como misión el cumplimiento de un conjunto de objetivos que conforman el norte de todas las actividades que se realicen:

- Estudio de las Tecnologías de la información que apoyan el proceso de desarrollo de los Sistemas de Información.
- Métodos y Metodologías para el desarrollo de los Sistemas de Información, utilizados en el mercado.
- Impacto de la Reingeniería de Procesos del Negocio y su vinculación con las Tecnologías de la Información.
- Guías de acción a las empresas del Sector Informático.
- Estructura Organizacional, Administración del Cambio y su relación con el proceso de desarrollo de los Sistemas de Información.
- Calidad en el Proceso de Desarrollo de los Sistemas de Información en Venezuela.

LISI establece relaciones dentro del sector productivo y empresas proveedoras de la plataforma tecnológica necesaria que le permite lograr el

cumplimiento de sus objetivos y llevar a cabo sus actividades de Investigación y Desarrollo.

Para comprender mejor el contexto donde se desenvuelve LISI, se muestran los modelos de negocio relacionados, tanto como el modelo de casos de uso del negocio, como el modelo objeto del negocio.

Modelos del Negocio

El Modelo de Casos de Uso presenta los principales procesos del negocio (LISI) , y sus principales actores.

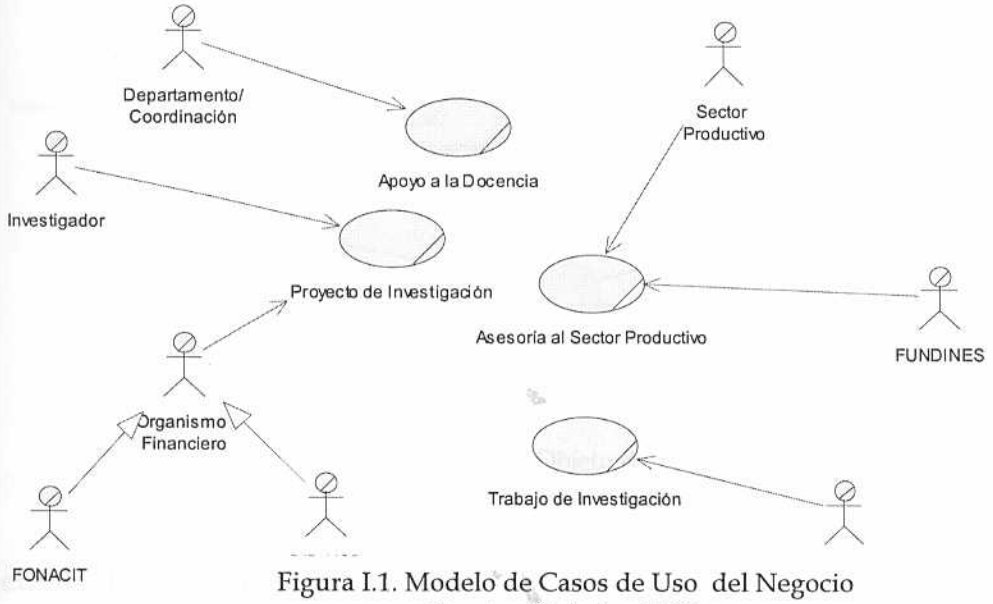

Figura I.1. Modelo de Casos de Uso del Negocio
Fuente: (Grimán, 2000)

La figura I.1. Presenta el Modelo de Casos de Uso del Negocio. Se puede enmarcar este Trabajo Especial de Grado bajo el actor "Tesista" realizando un "Trabajo de Investigación", dentro de la línea de investigación de arquitecturas de software.

El modelo de objetos del negocio, que se presenta a continuación describe la realización de los casos de uso del negocio. Sirve como una abstracción de cómo los trabajadores del negocio y las entidades del negocio necesitan ser relacionados y cómo ellos necesitan colaborar para lograr el desempeño del negocio.

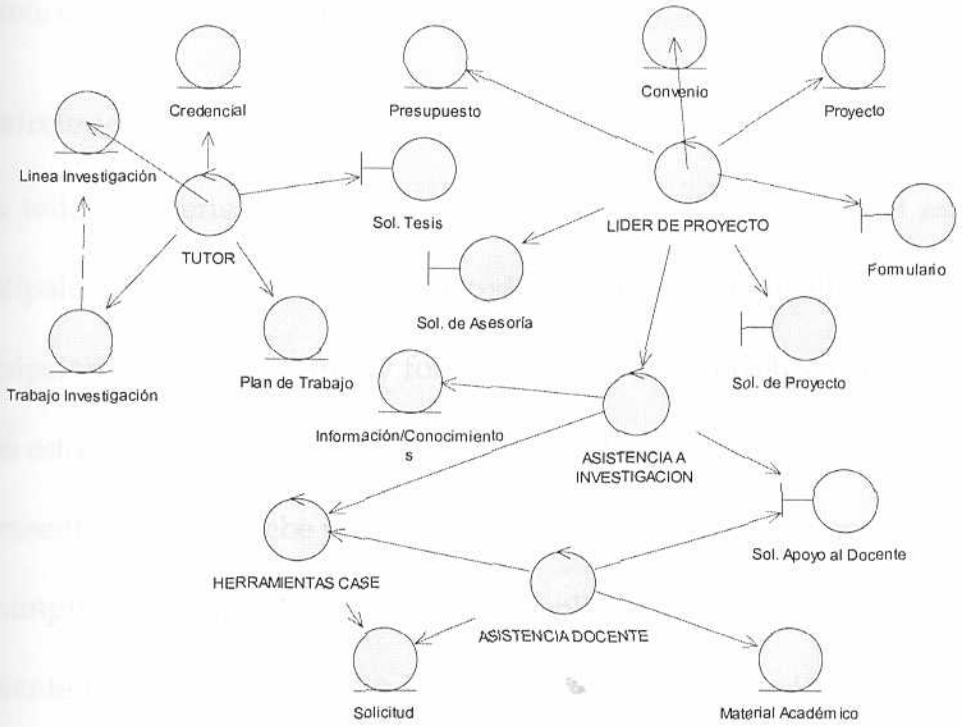

Figura I.2. Modelo Objeto del Negocio
Fuente: (Grimán, 2000)

Objetivos del Producto

El objetivo fundamental de este producto, es soportar el proceso de especificación y evaluación de las arquitecturas de software, para garantizar atributos de calidad a través de métodos, modelos y técnicas de evaluación, que permitan seleccionar una arquitectura idónea.

Pronóstico Financiero

Debido a que este proyecto de investigación forma parte de un Trabajo Especial de Grado, y sus fines son académicos no se estiman los costos que éste acarrea. En cuanto a la ganancia, ésta no se ve reflejada por el dinero que produzca; sin embargo, se obtiene un beneficio intangible al evaluar en etapas tempranas durante el proceso de desarrollo.

Restricciones

Ante toda esta serie de requerimientos y necesidades se inició el estudio de los principales modelos de calidad, métodos y técnicas de evaluación, lenguajes de descripción de la arquitectura y formas de diagramar objetos, con el fin de integrar todos estos elementos dentro de un ambiente de software.

El presente proyecto debe presentar una interfaz amigable y sencilla, sin necesidad de cumplir estrictamente un estándar de aplicación. En esta fase se ha definido el ambiente de trabajo, que las herramientas de apoyo son Rational Rose v.2002 y Power Designer 9.0, bajo las plataformas Windows 9X y Linux. Su desarrollo posterior será con JAVA debido a que contiene una gran cantidad de librerías gráficas que permite la manipulación de objetos, y SQL Server dado a que trabajan sin ningún problema bajo distintas plataformas.

APENDICE J - Plan Creativo de la Interfaz

En el plan creativo de la interfaz, se incluyeron aspectos para identificar estándares de colores, imágenes, tipología de letras, logos, etc.. El diseño de este ambiente incluyó la creación y elección del nombre, para ello se tomó las iniciales de las siglas Ambiente de Evaluación de Arquitecturas AMEVA.

El diseño del presenta un encabezado con el logotipo del ambiente, el logotipo del Laboratorio de Investigación de Sistemas de Información (LISI) y el logotipo de la Universidad Simón Bolívar, ya que el LISI es donde se desarrolla este Trabajo Especial de Grado y forma parte de mencionada universidad.

A continuación se presentan los diseños elaborados para el encabezado de dicho ambiente.

Figura J.1. Encabezado del Ambiente #1. Fuente: Elaboración Propia.

Figura J.2. Encabezado del Ambiente #2. Fuente: Elaboración Propia.

Figura J.3. Encabezado del Ambiente #3. Fuente: Elaboración Propia.

En lo que respecta al tamaño y tipo de letra, se utilizan mezclas de fuentes. A continuación se muestra los estándares de tipografía y color definidos para AMEVA.

Tabla J.1. Estándares de Interfaz. Fuente: Elaboración Propia.

Característica	Descripción
Estándares de tipografía y color	El tipo de letra base es Verdana, color negro , tamaño 10pts sin estilo, con color de fondo blanco o color blanco con el fondo negro. El texto se despliega preferiblemente justificado, sin decoración ni transformaciones. Los títulos son de color naranja si el fondo es blanco, o blancos si el fondo es negro, y los tamaños varían entre 16pts y 8pts.

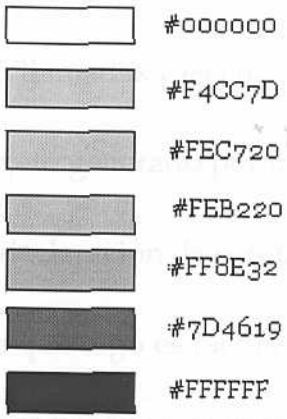

Figura J.4. Escala de Colores del Encabezado. Fuente: Elaboración Propia.

APENDICE K - Glosario

Se definió la terminología del dominio del problema, explicando los términos que pueden resultar poco familiares al lector de las descripciones de los Casos de Uso u otra secciones del proyecto.

Ambiente: es una configuración de computadora que se refiere al conjunto de arquitectura técnica, hardware y sistemas de software. El ambiente determina estándares para las aplicaciones que pueden ejecutarse en el computador.

Arquitectura: es la estructura o estructuras del sistema, la cual comprende componentes de software, las propiedades de estos componentes que son visibles externamente, y las relaciones entre ellos.

Atributo: se refiere a cada una de las cualidades o propiedades que definen a un elemento.

Calidad: Calidad es conformidad con los requisitos.

Componentes: son módulos activos, en el sentido de que reciben y generan eventos, realizan llamadas y ejecución de funciones.

Evento: es un objeto generado por una llamada a una acción.

Especificación: declaración de componentes o elementos de un objeto.

Factible: Se dice que algo es factible, cuando se puede realizar, si es probable que un hecho ocurra es porque éste es factible.

Herramienta: Es un programa que brinda a los programadores las facilidades necesarias para realizar una tarea.

Interfaces: un tipo interfaz describe la encapsulación de los componentes del sistema a nivel de usuario.

Métrica: medida o escala de valoración de un atributo de calidad.

Patrón: Es un modelo a seguir, diferentes objetos que cumplen con las mismas características cumplen con un patrón.

Patrones de Eventos: son expresiones que describen órdenes parciales de eventos.

Perfil: es un conjunto de escenarios, generalmente con alguna importancia relativa asociada con cada escenario.

Plataforma: ambiente de trabajo donde se agrupan los sistemas de una organización bajo un determinado sistema operativo.

Servidor: computador que proporciona servicios tales como acceso al sistema de archivos, la impresora o el sistema de correo electrónico a los usuarios de una red de área local. Usualmente es una combinación de software y hardware. Computador en un sistema de red compartido por múltiples usuarios.

Stakeholder: se refiere a todo el personal involucrado de alguna forma con el futuro sistema: desarrolladores, operadores, usuarios finales, clientes representativos, administradores de sistema, equipo de evaluación, gerentes, etc.

Respuesta: Contestar o Satisfacer a lo que se pregunta. En el sentido de la investigación es equivalente a la solución de un problema dado.

APÉNDICE L - Descripción de Casos de Uso Refinados

En esta sección se muestra la especificación de los casos de uso refinados.

Tabla L.1. Descripción de Casos de Uso Evaluar Arquitectura. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Evaluar Arquitectura	
Actor:	Arquitecto	
Descripción:	Este proceso abarca la evaluación completa de la arquitectura de software, basándose en los distintos modelos de especificación de la calidad, métodos y técnicas de evaluación y lenguajes formales	
Tipo de Relación:	Asociación	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- El Arquitecto solicita evaluar una arquitectura.	
		2.- El Ambiente despliega las tareas y/o pasos que el Arquitecto debe seguir.
	3.- El arquitecto activa cada paso (secuencialmente).	
		4.- El Ambiente arroja el resultado obtenido.
Curso Alterno:	Al paso 3: El Arquitecto omite un paso. El Ambiente muestra un mensaje de error y lo devuelve al paso anterior al omitido.	
Pre-Condiciones:	El ambiente debe haberse inicializado.	
Post-Condiciones:		

Tabla L.2. Descripción de Casos de Uso Aplicar ADL. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Aplicar del ADL	
Actor:	Arquitecto	
Descripción:	En este proceso el desarrollador aplicará la arquitectura a un lenguaje de descripción de arquitecturas.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- Se activa por parte del Arquitecto.	
		2.- El Ambiente dispone por lo menos de un ADL.
	3.- El Arquitecto especifica la arquitectura.	
		4.- El Ambiente hace la evaluación.
Curso Alterno:	Al paso 2: El ambiente no posee ningún ADL. El Arquitecto es devuelto al paso anterior y escoge otra técnica de evaluación.	
Pre-Condiciones:	Haber activado el proceso de evaluación de la arquitectura. El modelo de calidad, el método de evaluación y la técnica de evaluación se debieron escoger previamente.	
Post-Condiciones:		

Tabla L.3. Descripción de Casos de Uso Aplicar Técnica de Evaluación. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Aplicar Técnica de Evaluación	
Actor:	Arquitecto	
Descripción:	En este proceso el desarrollador aplicará la arquitectura a una técnica de evaluación.	
Tipo de Relación:	Incluye	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- El Arquitecto solicita la técnica para evaluar la Arquitectura.	
		2.- El Ambiente dispone de varias técnicas de evaluación.
	3.- El Arquitecto selecciona el deseado.	
		4.- El Ambiente solicita la definición de la técnica.
Curso Alterno:	Al paso 3: El Arquitecto solicitó evaluar con un ADL, entonces pasa al siguiente caso de uso.	
Pre-Condiciones:	Haber activado el proceso de evaluación de la arquitectura. El modelo de calidad y el método de evaluación se debieron escoger previamente.	
Post-Condiciones:	La definición de escenarios.	

Tabla L.4. Descripción de Casos de Uso Aplicar Método de Evaluación. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Aplicar Método de Evaluación	
Actor:	Arquitecto	
Descripción:	En este proceso el desarrollador aplicará la arquitectura a un método de evaluación.	
Tipo de Relación:	Incluye	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- El Arquitecto solicita el método para evaluar la Arquitectura.	
		2.- El Ambiente dispone de varios métodos de evaluación.
	3.- El Arquitecto selecciona el deseado.	
		4.- El Ambiente, según las especificaciones de calidad, comienza la evaluación.
		5.- El ambiente solicita la especificación de la técnica de evaluación a utilizar.
	6.- El Arquitecto introduce las especificaciones.	
		7.- El Ambiente procede a evaluar.
Curso Alterno:	Al paso 6: Al especificar la técnica, no queda bien especificada. El Ambiente devuelve un mensaje de error y devuelve al Arquitecto de nuevo al paso 6 de este caso de uso.	
Pre-Condiciones:	Haber activado el proceso de evaluación de la arquitectura. Se debió especificar la calidad previamente.	
Post-Condiciones:	Se deben mostrar las técnicas de evaluación.	

Tabla L.5. Descripción de Casos de Uso Especificar la Calidad. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Especificar la Calidad	
Actor:	Arquitecto	
Descripción:	En este proceso se darán las especificaciones de calidad requeridas.	
Tipo de Relación:	Incluye	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- El Arquitecto solicita la especificación de la calidad.	
		2.- El Ambiente dispone de varios modelos de especificación de calidad.
	3.- El Arquitecto selecciona el modelo deseado y los atributos para la evaluación..	
		4.- El Ambiente muestra sus especificaciones.
	5.- El Arquitecto basado en los requerimientos No Funcionales los selecciona.	
Curso Alterno:	NINGUNO	
Pre-Condiciones:	El ambiente debe haberse inicializado y el proceso de evaluación debe estar activado.	
Post-Condiciones:	La selección de atributos se guardan en un vector.	

Tabla L.6. Descripción de Casos de Uso Generar Documentación. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Generar Documentación	
Actor:	Arquitecto	
Descripción:	Su propósito es dar el resultado de la evaluación realizada. Aquí se verificará que la arquitectura propuesta para el sistema provee el funcionamiento óptimo.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- Se activa cuando el Arquitecto solicita una documentación sobre los resultados obtenidos de la evaluación.	2.- El Ambiente genera un documento (.txt) con el resultado obtenido y le da al Arquitecto la opción de imprimir.
Curso Alterno:	El Arquitecto no solicitó la documentación al terminar la evaluación. El Arquitecto debe buscar en el directorio principal un archivo de extensión .txt correspondiente a su evaluación.	
Pre-Condiciones:	El Ambiente evaluó recientemente una Arquitectura.	
Post-Condiciones:	El Ambiente regresa al usuario a la pantalla principal.	

Tabla L.7. Descripción de Casos de Uso Especificar Arquitectura. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Especificar Arquitectura	
Actor:	Arquitecto	
Descripción:	Su propósito es introducir y actualizar todo la información referente a las arquitecturas de software dentro del ambiente, sus características funcionales, principalmente.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- Se inicia cuando el Arquitecto solicita especificar la arquitectura.	2.- El Ambiente provee los elementos para su especificación. 3.- Según lo requerido por el usuario, el Ambiente mostrará funcionalidades y/o elementos.
Curso Alterno:	Al paso 3: que el requerimiento solicitado no se encuentra disponible en el ambiente.	
Pre-Condiciones:	El ambiente debe haberse inicializado y el proceso de evaluación activado.	
Post-Condiciones:	El ambiente pide la especificación de la Calidad.	

Tabla L.8. Descripción de Casos de Uso Mantenimiento del Sistema. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Mantenimiento del Sistema	
Actor:	Arquitecto	
Descripción:	Permite actualizar el sistema, dando origen a nuevas actividades dentro del ambiente.	
Tipo de Relación:	Asociación	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- El Arquitecto desea incluir, modificar y/o eliminar algún elemento del Ambiente.	2.- Se muestran las opciones: Modelos de Especificación de la Calidad, Método de Evaluación, Técnica de Evaluación, ADL.
	3.- El Arquitecto escoge el elemento.	4.- Para cada elemento especifico, el Ambiente desplegará las pantallas.
Curso Alterno:	Al paso 2: El Arquitecto solo puede escoger entre las opciones señaladas.	
Pre-Condiciones:	El ambiente debe haberse inicializado.	
Post-Condiciones:	Se muestran las opciones de mantenimiento.	

Tabla L.9. Descripción de Casos de Uso Administrar Modelo de Calidad. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Administrar Modelo de Calidad	
Actor:	Arquitecto	
Descripción:	Este proceso permite incluir un nuevo modelo de calidad al ambiente de evaluación, modificar o eliminar uno existente.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- El Arquitecto solicita ejecutar una operación de mantenimiento (Modelo de Calidad).	2.- El Ambiente le proporciona las opciones que tiene disponible.
	3.- Escoge la conveniente (agregar, modificar, eliminar) selecciona una y hace click aceptar.	
Curso Alterno:	NINGUNO.	
Pre-Condiciones:	El ambiente debe haberse inicializado haber activado la opción de Mantenimiento.	
Post-Condiciones	Se muestran las opciones de Administrar Modelo de Calidad.	

Tabla L.10. Descripción de Casos de Uso Ingresar Modelo de Calidad. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Ingresar Modelo de Calidad	
Actor:	Arquitecto	
Descripción:	Este proceso permite incluir un nuevo modelo de calidad al ambiente de evaluación.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- Se inicia cuando el Arquitecto solicita ingresar un modelo de calidad.	
		2.- El ambiente presenta una plantilla para agregar un modelo nuevo.
	3.- Rellena los campos solicitados y hace click en aceptar.	
		4.- Muestra un mensaje de confirmación.
	5.- El Arquitecto aprueba o desaprueba.	
		6.- Respuesta afirmativa: el Ambiente ingresa el modelo y actualiza en BD. Respuesta negativa: el Ambiente regresa al usuario a la pantalla principal sin actualizar la BD.
Curso Alterno:	Al paso 3: Los datos introducidos no son validos. El Ambiente muestra mensaje de error en introducción de datos.	
Pre-Condiciones:	La opción Administrar Modelos de Calidad se debió haber activado.	
Post-Condiciones	El Ambiente regresa al usuario a la pantalla principal.	

Tabla L.11. Descripción de Casos de Uso Modificar Modelo de Calidad. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Modificar Modelo de Calidad	
Actor:	Arquitecto	
Descripción:	Este proceso permite incluir un nuevo modelo de calidad al ambiente de evaluación, modificar o eliminar uno existente.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- Se inicia cuando el Arquitecto solicita modificar un modelo de calidad.	
		2.- El Ambiente le proporciona los modelos existentes.
	3.- Hace click sobre el escogido para modificarlo.	
		4.- Muestra los valores del modelo.
	5.- El Arquitecto modifica los valores convenientes y hace click en modificar.	
		6.- Muestra un mensaje de confirmación de la tarea.
	7.- El Arquitecto aprueba o desaprueba.	
		8.- Respuesta afirmativa: el Ambiente modifica el modelo y actualiza en BD. Respuesta negativa: el Ambiente regresa al usuario a la pantalla principal sin actualizar la BD.
Curso Alterno:	Al paso 3: Los datos modificados no son válidos. El Ambiente muestra un mensaje de error al modificar los datos. Al paso 4: Solo puede modificar la información almacenada en BD, no se pueden modificar los métodos asociados al Modelo de Calidad.	
Pre-Condiciones:	La opción Administrar Modelos de Calidad se debió haber activado.	
Post-Condiciones	El Ambiente regresa al usuario a la pantalla principal.	

Tabla L.12. Descripción de Casos de Uso Eliminar Modelo de Calidad. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Eliminar Modelo de Calidad	
Actor:	Arquitecto	
Descripción:	Este proceso permite eliminar un modelo de calidad al ambiente de evaluación.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- Se inicia cuando el Arquitecto solicita eliminar un Modelo de Calidad.	2.- El Ambiente muestra los modelos disponibles en el sistema.
	3.- El Arquitecto selecciona el modelo que desea eliminar y le da click al botón borrar.	4.- Muestra un mensaje de confirmación de la tarea.
	5.- El Arquitecto aprueba o desaprueba.	6.- Respuesta afirmativa: el Ambiente borra el modelo y actualiza en BD su existencia. Respuesta negativa: el Ambiente regresa al usuario a la pantalla principal sin efectuar la tarea.
Curso Alterno:	NINGUNO.	
Pre-Condiciones:	La opción Administrar Modelo de Calidad se debió haber activado.	
Post-Condiciones	El Ambiente regresa al usuario a la pantalla principal.	

Tabla L.13. Descripción de Casos de Uso Administrar Método de Evaluación. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Administrar Método de Evaluación	
Actor:	Arquitecto	
Descripción:	Este proceso permite incluir un nuevo método de evaluación al ambiente, modificar o eliminar uno existente.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- El Arquitecto solicita ejecutar una operación de mantenimiento (Método de Evaluación).	
		2.- El Ambiente le proporciona las opciones que tiene disponible.
	3.- Escoge la conveniente (agregar, modificar, eliminar) y hace click en aceptar.	
Curso Alterno:	NINGUNO.	
Pre-Condiciones:	El ambiente debe haberse inicializado haber activado la opción de Mantenimiento.	
Post-Condiciones:	Se muestran las opciones de Administrar Método de Evaluación.	

Tabla L.14. Descripción de Casos de Uso Ingresar Método de Evaluación. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Ingresar Método de Evaluación	
Actor:	Arquitecto	
Descripción:	Este proceso permite incluir un nuevo método de evaluación al ambiente.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- Se inicia cuando el Arquitecto solicita ingresar un método de evaluación.	
		2.- El ambiente presenta una plantilla para agregar un método nuevo.
	3.- Rellena los campos solicitados y hace click en aceptar.	
		4.- Muestra un mensaje de confirmación.
	5.- El Arquitecto aprueba o desaprueba.	
		6.- Respuesta afirmativa: el Ambiente ingresa el método y actualiza en BD. Respuesta negativa: el Ambiente regresa al usuario a la pantalla principal sin actualizar la BD.
Curso Alterno:	Al paso 3: Los datos introducidos no son validos. El Ambiente muestra mensaje de error en introducción de datos.	
Pre-Condiciones:	La opción Administrar Modelos de Calidad se debió haber activado.	
Post-Condiciones:	El Ambiente regresa al usuario a la pantalla principal.	

Tabla L.15. Descripción de Casos de Uso Modificar Métodos de Evaluación. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Modificar Método de Evaluación	
Actor:	Arquitecto	
Descripción:	Este proceso permite modificar método de evaluación al ambiente.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- Se inicia cuando el Arquitecto solicita modificar un método de evaluación.	
		2.- El Ambiente le proporciona los métodos existentes.
	3.- Hace click sobre el escogido para modificarlo.	
		4.- Muestra los valores del modelo.
	5.- El Arquitecto modifica los valores convenientes.	
		6.- Muestra un mensaje de confirmación de la tarea.
	7.- El Arquitecto aprueba o desaprueba.	
		8.- Respuesta afirmativa: el Ambiente modifica el método y actualiza en BD. Respuesta negativa: el Ambiente regresa al usuario a la pantalla principal sin actualizar la BD.
Curso Alterno:	Al paso 3: Los datos modificados no son válidos. El Ambiente muestra un mensaje de error al modificar los datos. Al paso 4: Solo puede modificar la información almacenada en BD, no se pueden modificar los métodos asociados al Método de Evaluación.	
Pre-Condiciones:	La opción Administrar Método de Evaluación se debió haber activado.	
Post-Condiciones:	El Ambiente regresa al usuario a la pantalla principal.	

Tabla L.16. Descripción de Casos de Uso Eliminar Método de Evaluación. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Eliminar Método de Evaluación	
Actor:	Arquitecto	
Descripción:	Este proceso permite eliminar un método de evaluación al ambiente.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- Se inicia cuando el Arquitecto solicita eliminar un Método de Evaluación.	2.- El Ambiente muestra las métodos disponibles en el sistema.
	3.- El Arquitecto selecciona el método que desea eliminar y le da click al botón borrar.	4.- Muestra un mensaje de confirmación de la tarea.
	5.- El Arquitecto aprueba o desaprueba.	6.- Respuesta afirmativa: el Ambiente borra el método y actualiza en BD su existencia. Respuesta negativa: el Ambiente regresa al usuario a la pantalla principal sin efectuar la tarea.
Curso Alterno:	NINGUNO.	
Pre-Condiciones:	La opción Administrar Método de Evaluación se debió haber activado.	
Post-Condiciones:	El Ambiente regresa al usuario a la pantalla principal.	

Tabla L.17. Descripción de Casos de Uso Administrar Técnica de Evaluación. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Administrar Técnica de Evaluación	
Actor:	Arquitecto	
Descripción:	Este proceso permite incluir una nueva técnica de evaluación, modificar o eliminar una existente.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- El Arquitecto solicita ejecutar una operación de mantenimiento: Técnica de Evaluación.	2.- El Ambiente le proporciona las opciones que tiene disponible.
	3.- Escoge la conveniente (agregar, modificar, eliminar) y hace click en aceptar.	
Curso Alterno:	NINGUNO.	
Pre-Condiciones:	El ambiente debe haberse inicializado haber activado la opción de Mantenimiento.	
Post-Condiciones:	Se muestran las opciones de Administrar Técnica de Evaluación.	

Tabla L.18. Descripción de Casos de Uso Ingresar Técnica de Evaluación. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Ingresar Técnica de Evaluación	
Actor:	Arquitecto	
Descripción:	Este proceso permite incluir una nueva técnica de evaluación al ambiente.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- Se inicia cuando el Arquitecto solicita ingresar una técnica de evaluación.	
		2.- El ambiente presenta una plantilla para agregar una técnica nuevo.
	3.- Rellena los campos solicitados y hace click en aceptar.	
		4.- Muestra un mensaje de confirmación.
	5.- El Arquitecto aprueba o desaprueba.	6.- Respuesta afirmativa: el Ambiente ingresa el técnica y actualiza en BD. Respuesta negativa: el Ambiente regresa al usuario a la pantalla principal sin actualizar la BD.
Curso Alterno:	Al paso 5: Los datos introducidos no son validos. El Ambiente muestra mensaje de error en introducción / modificación de datos.	
Pre-Condiciones:	La opción Administrar Técnica de Evaluación se debió haber activado.	
Post-Condiciones:	El Ambiente regresa al usuario a la pantalla principal.	

Tabla L.19. Descripción de Casos de Uso Eliminar Técnica de Evaluación. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Eliminar Técnica de Evaluación	
Actor:	Arquitecto	
Descripción:	Este proceso permite eliminar técnica una de evaluación.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- Se inicia cuando el Arquitecto solicita eliminar una Técnica de Evaluación.	2.- El Ambiente muestra las técnicas disponibles en el sistema.
	3.- El Arquitecto selecciona la técnica que desea eliminar y le da click al botón borrar.	4.- Muestra un mensaje de confirmación de la tarea.
	5.- El Arquitecto aprueba o desaprueba.	6.- Respuesta afirmativa: el Ambiente borra la técnica y actualiza en BD su existencia. Respuesta negativa: el Ambiente regresa al usuario a la pantalla principal sin efectuar la tarea.
Curso Alterno:	NINGUNO.	
Pre-Condiciones:	La opción Administrar Técnica de Evaluación se debió haber activado.	
Post-Condiciones:	El Ambiente regresa al usuario a la pantalla principal.	

Tabla L.20. Descripción de Casos de Uso Modificar Técnica de Evaluación. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Modificar Técnica de Evaluación	
Actor:	Arquitecto	
Descripción:	Este proceso permite modificar una nueva técnica de evaluación.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- Se inicia cuando el Arquitecto solicita modificar una técnica de evaluación.	
		2.- El Ambiente le proporciona las técnicas existentes.
	3.- Hace click sobre el escogido para modificarlo.	
		4.- Muestra los valores del modelo.
	5.- El Arquitecto modifica los valores convenientes y hace click en modificar.	
		6.- Muestra un mensaje de confirmación de la tarea.
	7.- El Arquitecto aprueba o desaprueba.	
		8.- Respuesta afirmativa: el Ambiente modifica la técnica y actualiza en BD. Respuesta negativa: el Ambiente regresa al usuario a la pantalla principal sin actualizar la BD.
Curso Alterno:	Al paso 3: Los datos modificados no son válidos. El Ambiente muestra un mensaje de error al modificar los datos. Al paso 4: Solo puede modificar la información almacenada en BD, no se pueden modificar los métodos asociados a la Técnica de Evaluación.	
Pre-Condiciones:	La opción Administrar Técnica de Evaluación se debió haber activado.	
Post-Condiciones:	El Ambiente regresa al usuario a la pantalla principal.	

Tabla L.21. Descripción de Casos de Uso Administrar ADL. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Administrar ADL	
Actor:	Arquitecto	
Descripción:	Este proceso permite incluir un nuevo ADL, modificar o eliminar uno existente.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- El Arquitecto solicita ejecutar una operación de mantenimiento ADL.	2.- El Ambiente le proporciona las opciones que tiene disponible.
	3.- Escoge la conveniente (agregar, eliminar) y hace click en aceptar.	
Curso Alterno:	NINGUNO.	
Pre-Condiciones:	El ambiente debe haberse inicializado haber activado la opción de Mantenimiento.	
Post-Condiciones:	Se muestran las opciones de Administrar ADL.	

Tabla L.22. Descripción de Casos de Uso Ingresar ADL. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Ingresar ADL	
Actor:	Arquitecto	
Descripción:	Este proceso permite incluir un nuevo ADL al Ambiente.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- Se inicia cuando el Arquitecto solicita ingresar un ADL.	2.- El Ambiente presenta tiene disponible.
	3.- El usuario carga el ADL y le hace click en aceptar.	4.- Muestra un mensaje si se desea ejecutar la operación indicada.
	5.- El Arquitecto aprueba o desaprueba.	6.- Respuesta afirmativa: el Ambiente incluye el nuevo componente y actualiza la BD de su existencia. Respuesta negativa: el Ambiente regresa al usuario a la pantalla principal sin actualizar la BD ni el ambiente.
Curso Alterno:	Al paso 3: El ADL no es compatible con el Ambiente. Se muestra un mensaje de error y no se efectúa la operación.	
Pre-Condiciones:	La opción Administrar ADL se debió haber activado.	
Post-Condiciones:	El Ambiente regresa al usuario a la pantalla principal.	

Tabla L.23. Descripción de Casos de Uso Eliminar ADL. Fuente: Elaboración Propia.

Nombre del Caso de Uso:	Eliminar ADL	
Actor:	Arquitecto	
Descripción:	Este proceso permite eliminar un ADL de ambiente.	
Tipo de Relación:	Extiende	
Curso Normal:	Acciones del Actor	Respuesta del Ambiente
	1.- Se inicia cuando el Arquitecto solicita eliminar un ADL.	
		2.- El Ambiente muestra los ADLs disponibles en el sistema.
	3.- El Arquitecto selecciona el ADL que desea eliminar y le da click al botón borrar.	
		4.- Muestra un mensaje de confirmación de la tarea.
	5.- El Arquitecto aprueba o desaprueba.	
		6.- Respuesta afirmativa: el Ambiente borra el ADL y actualiza en BD su existencia. Respuesta negativa: el Ambiente regresa al usuario a la pantalla principal efectuar la tarea.
Curso Alterno:	NINGUNO	
Pre-Condiciones:	La opción Administrar ADL se debió haber activado. Debe existir al menos un ADL.	
Post-Condiciones:	El Ambiente regresa al usuario a la pantalla principal.	

APÉNDICE M – Diagramas de Secuencia

A continuación se muestran los diagramas de secuencia asociados a los casos de uso del sistema.

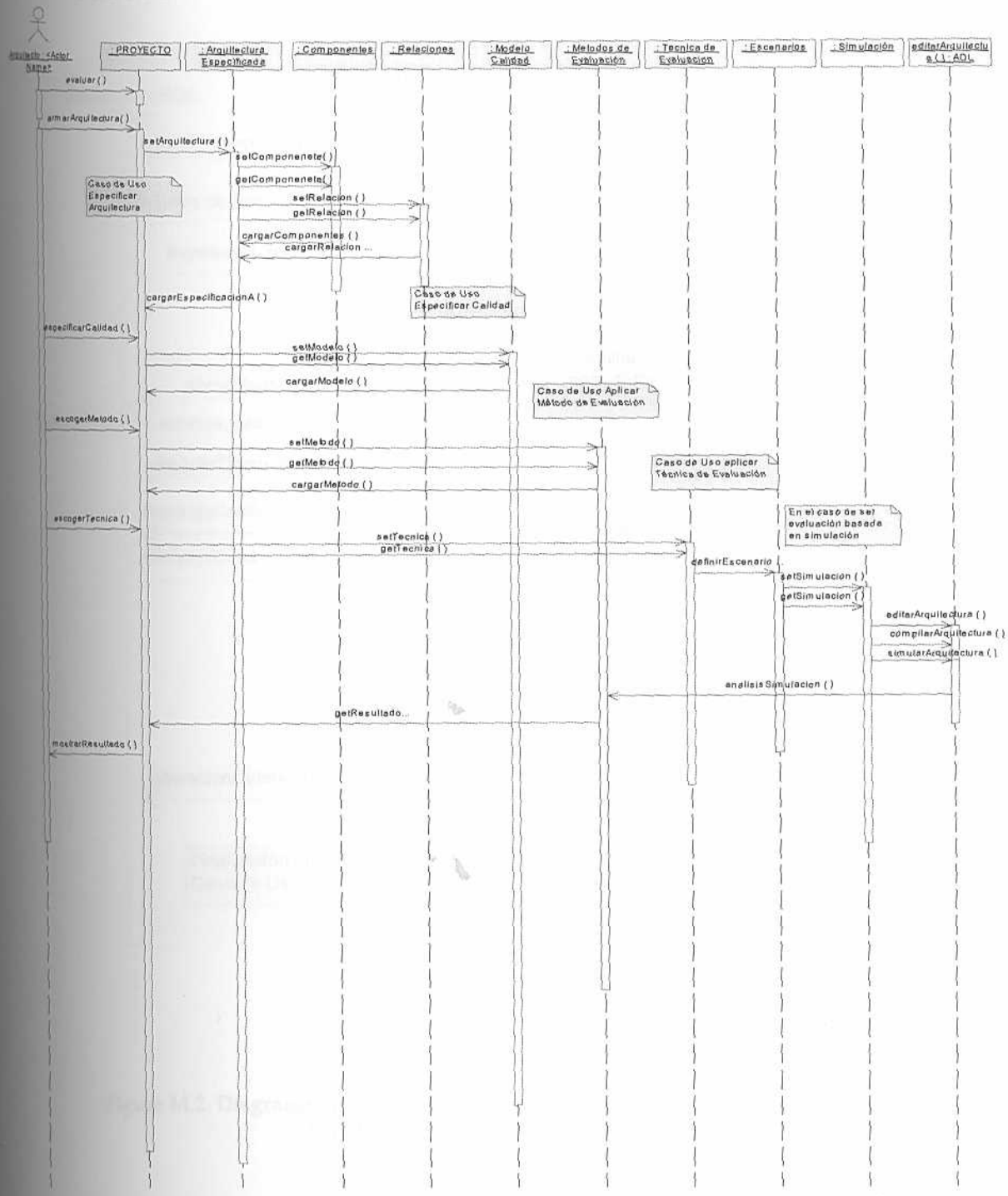

Figura M.1. Diagrama de Secuencia. C.U. Evaluar Arquitectura. Fuente: Elaboración Propia.

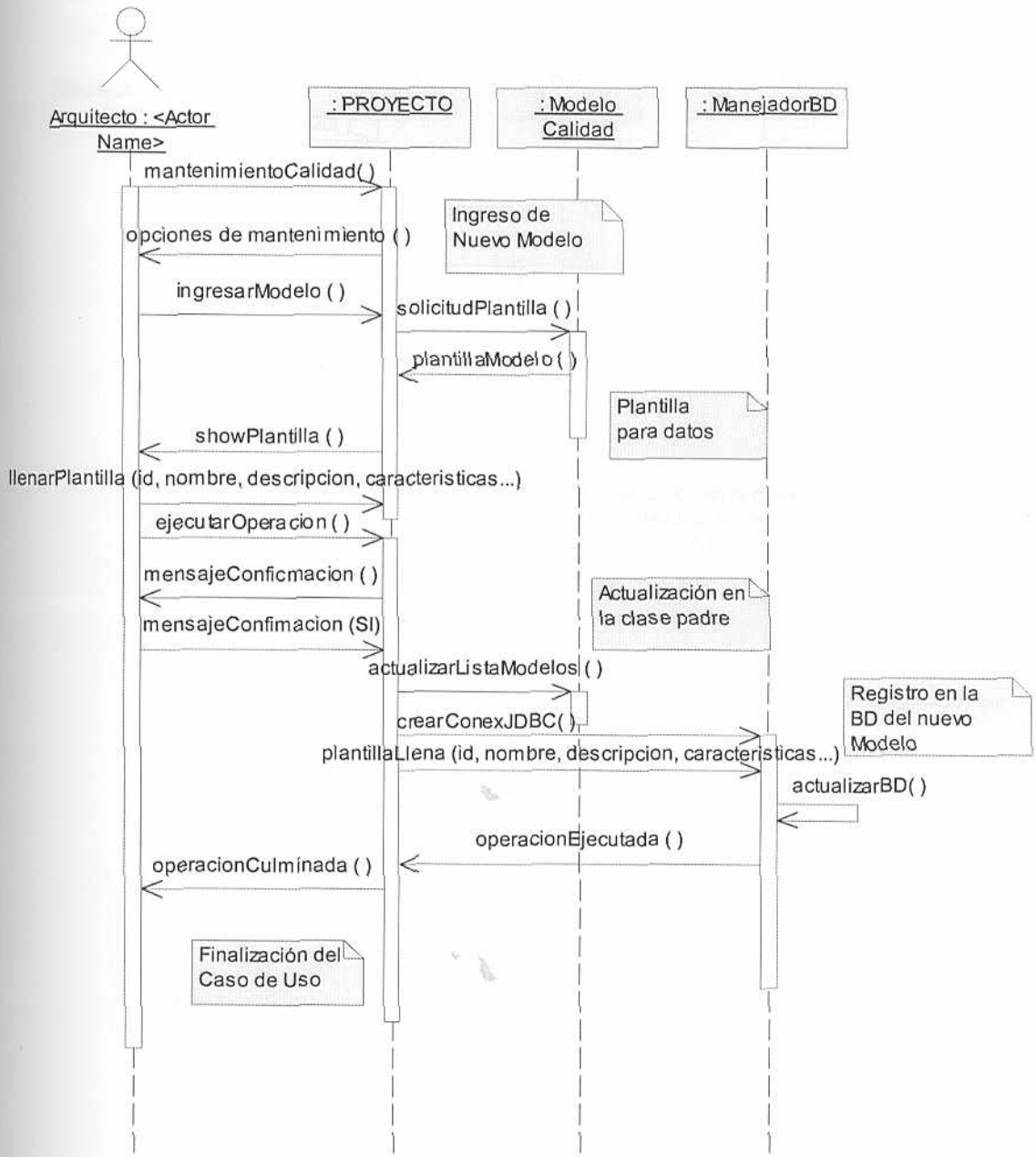

Figura M.2. Diagrama de Secuencia. C.U. Agregar Modelo de Calidad.
Fuente: Elaboración Propia.

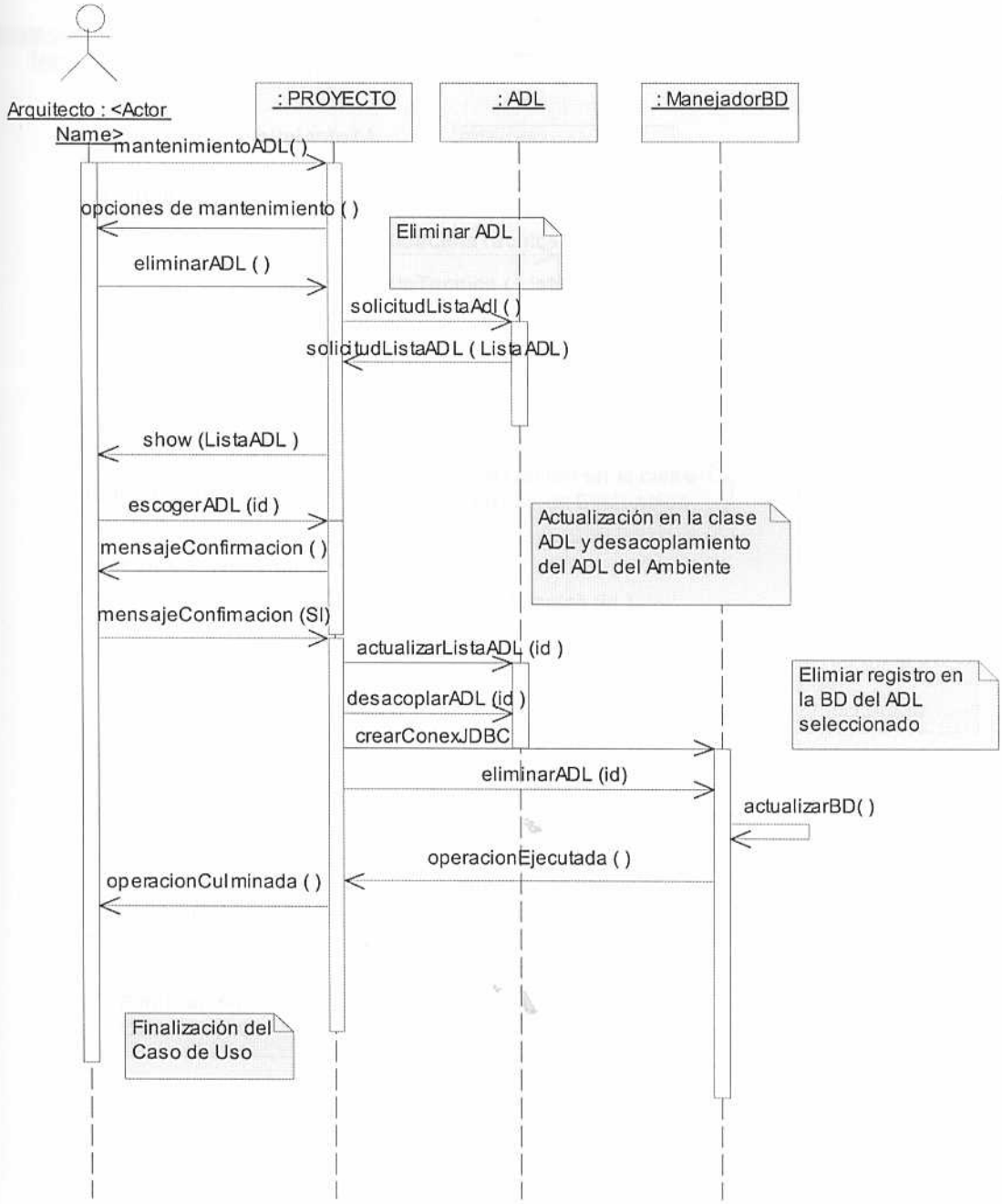

Figura M.3. Diagrama de Secuencia. C.U. Eliminar ADL.
Fuente: Elaboración Propia.

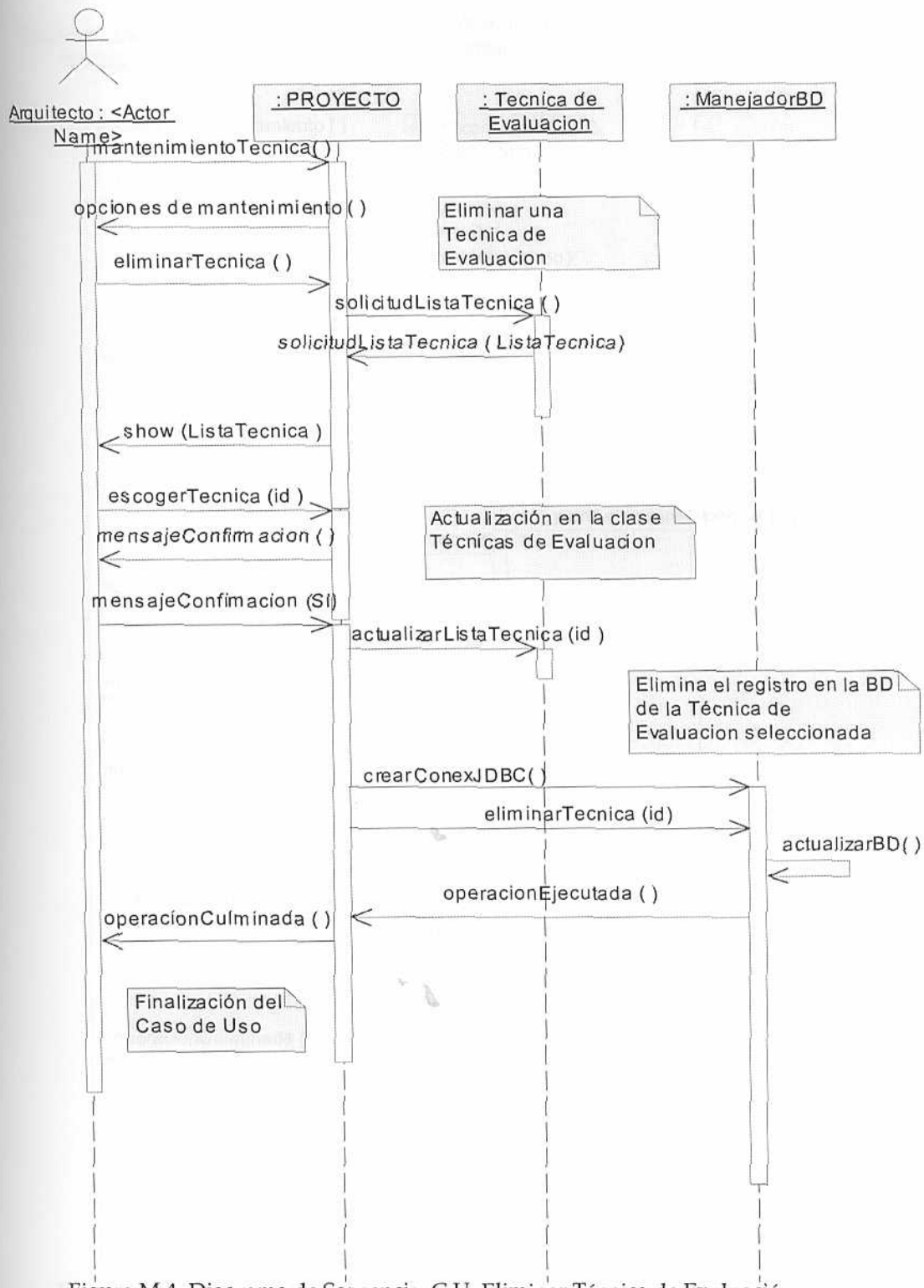

Figura M.4. Diagrama de Secuencia. C.U. Eliminar Técnica de Evaluación.
Fuente: Elaboración Propia.

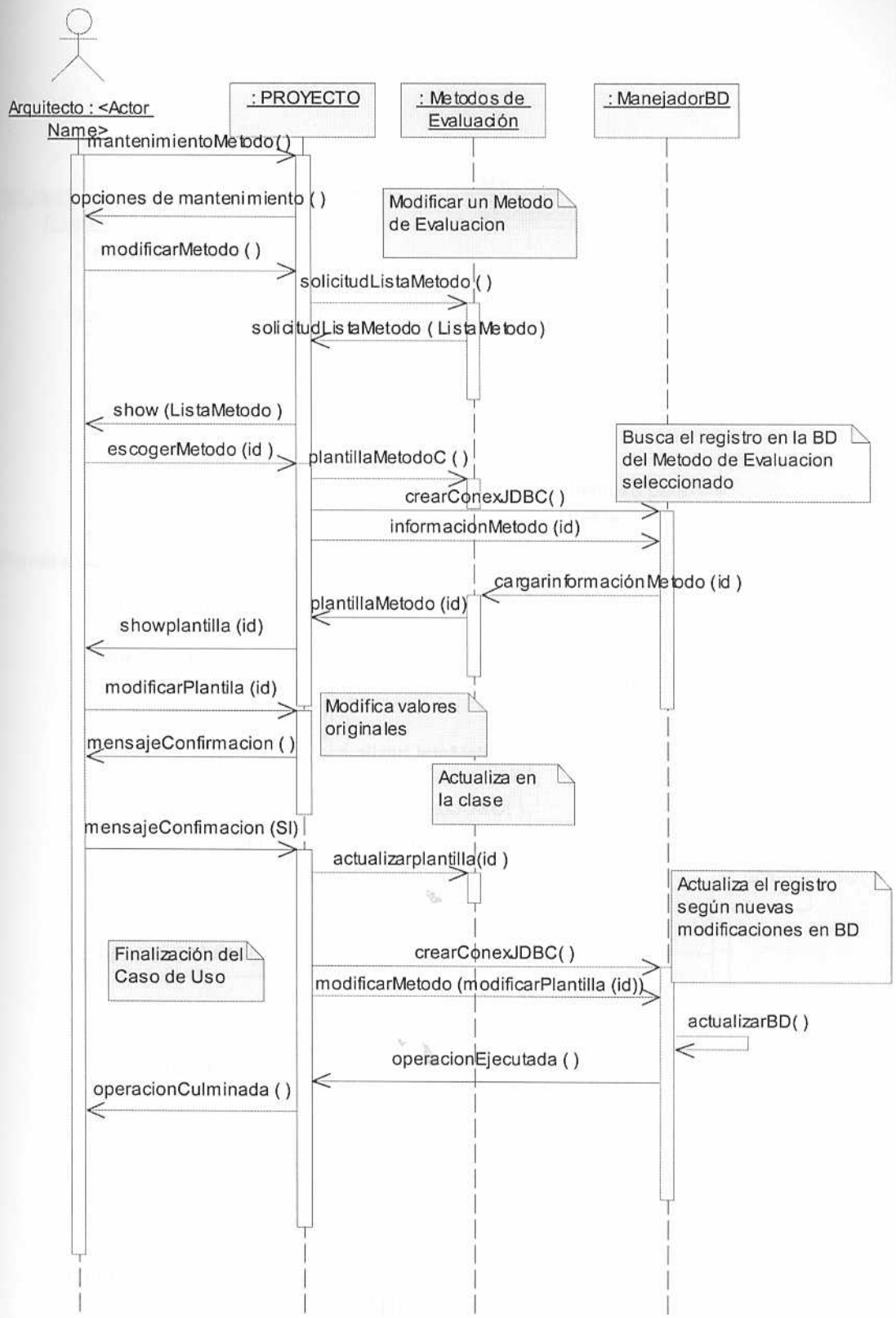

Figura M.5. Diagrama de Secuencia. C.U. Modificar Método de Evaluación.
Fuente: Elaboración Propia.

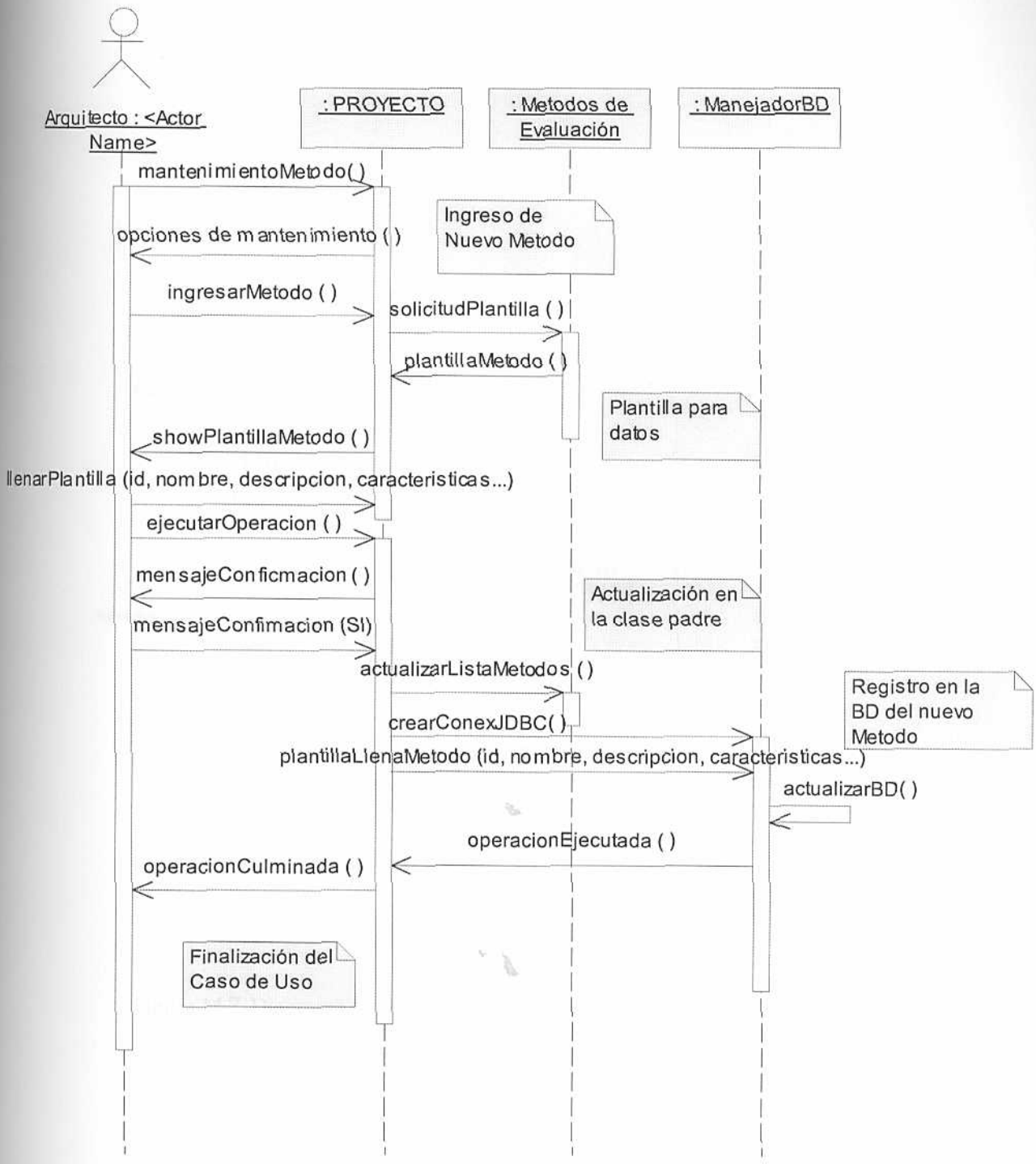

Figura M.6. Diagrama de Secuencia. C.U. Agregar Método de Evaluación.
Fuente: Elaboración Propia.

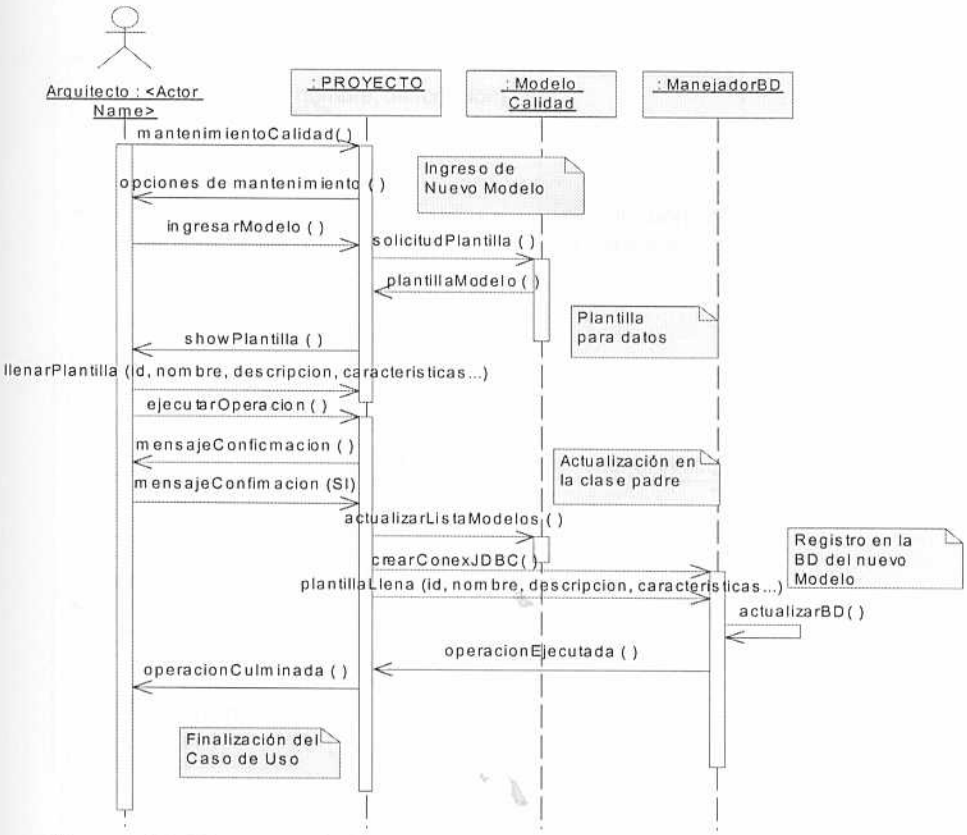

Figura M.7. Diagrama de Secuencia de Ingresar Modelo de Calidad.
Fuente: Elaboración Propia.

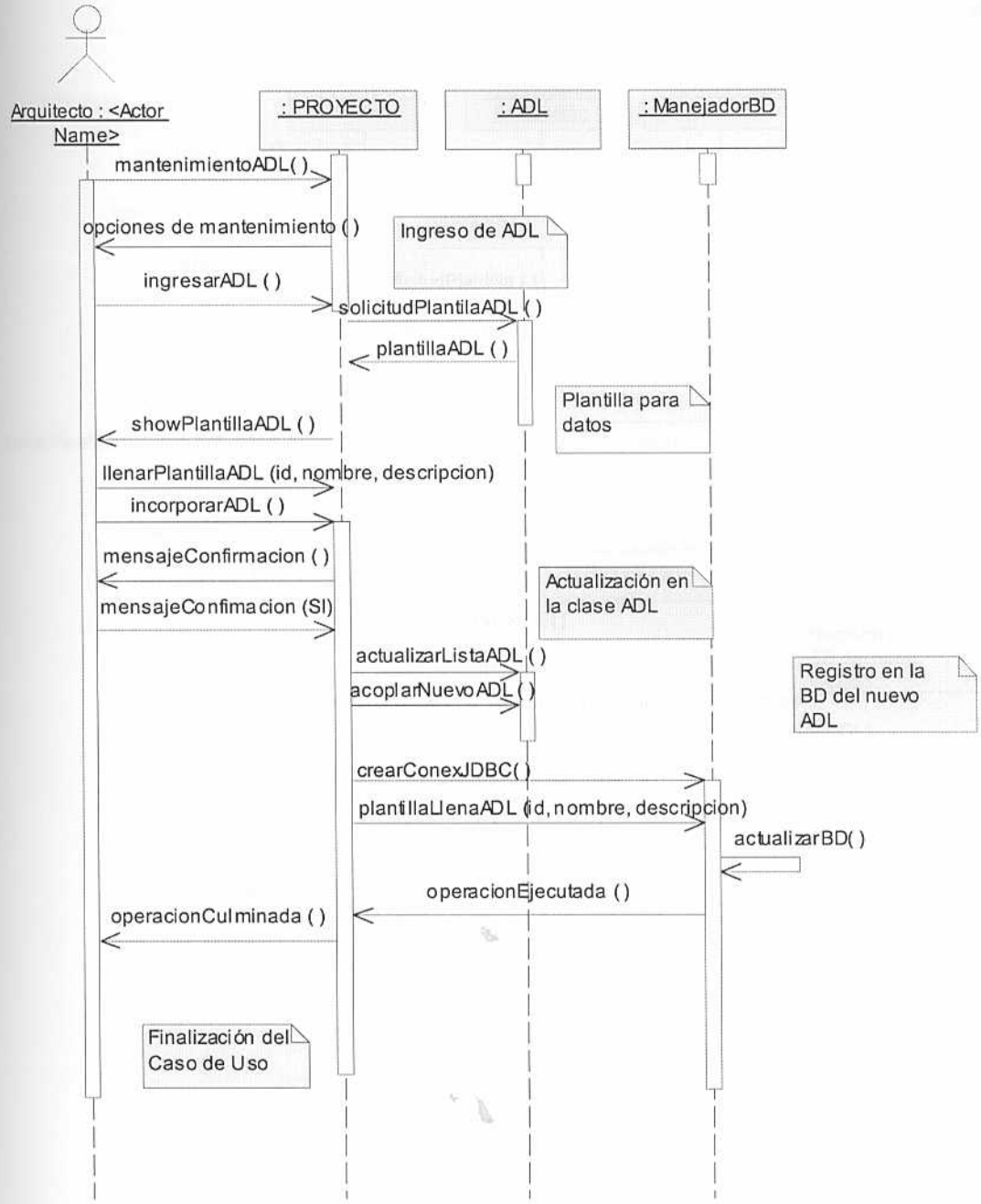

Figura M.8. Diagrama de Secuencia de Ingresar ADL.
Fuente: Elaboración Propia.

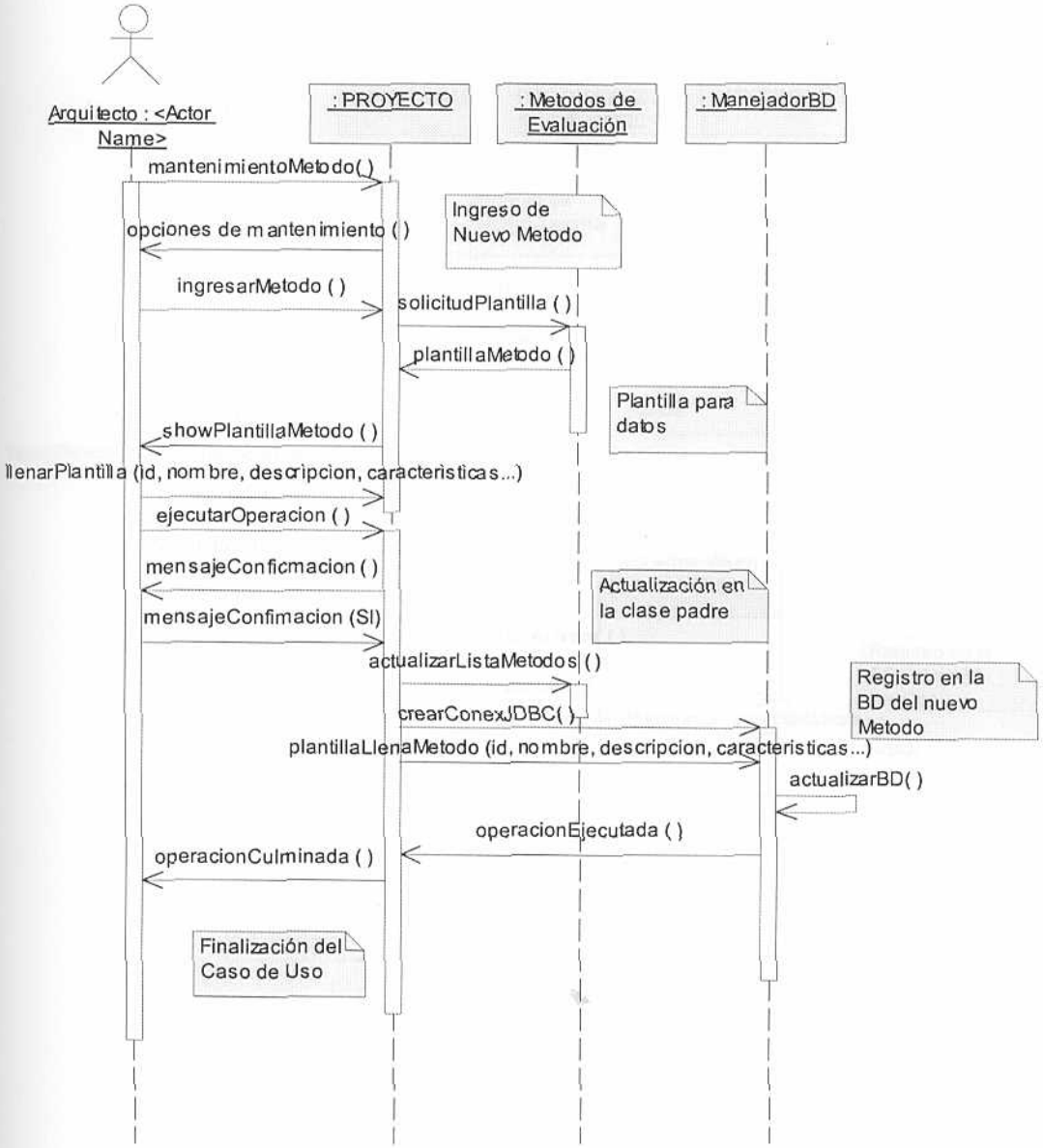

Figura M.9. Diagrama de Secuencia de Ingresar Método de Evaluación.
Fuente: Elaboración Propia.

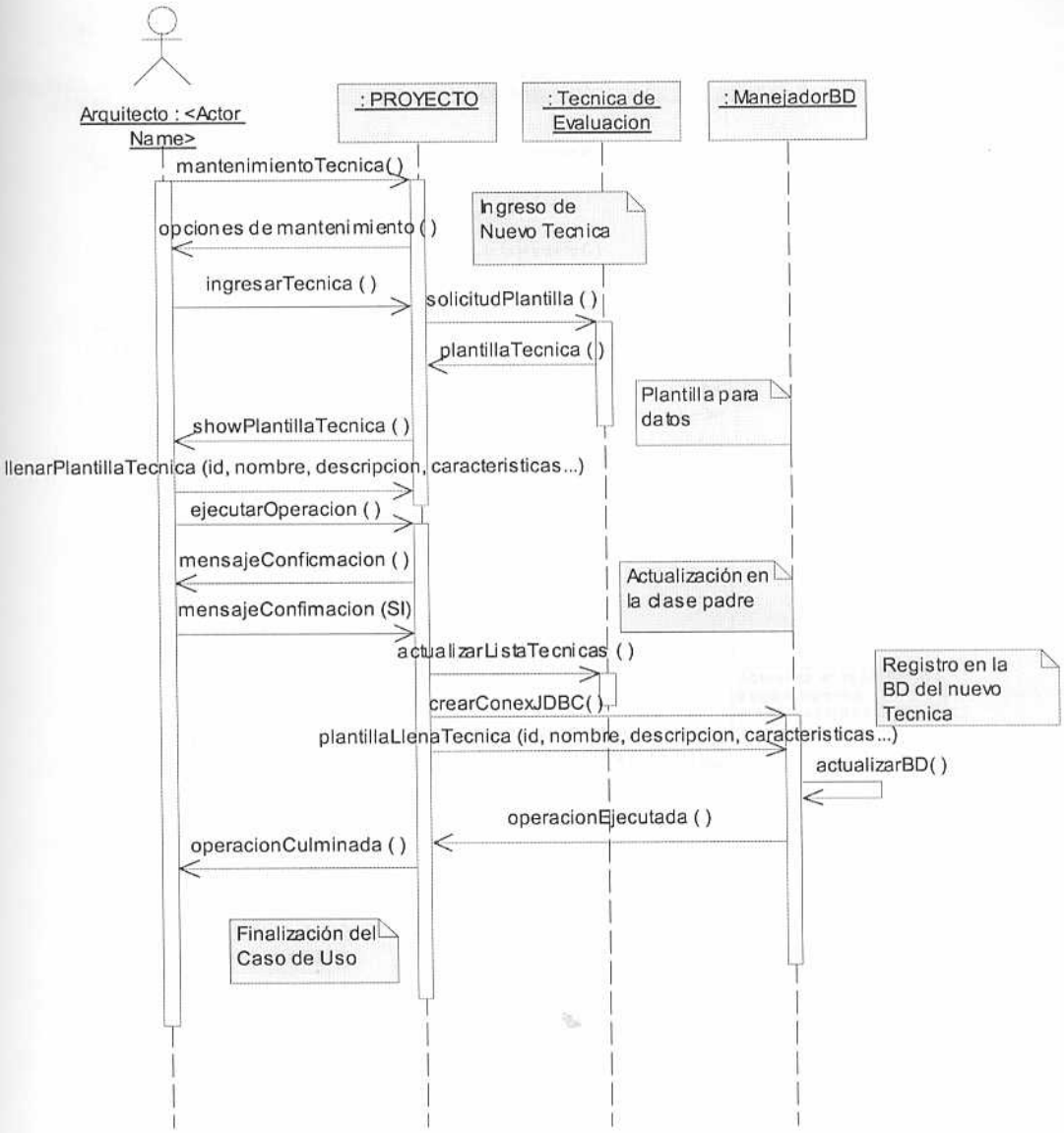

Figura M.10. Diagrama de Secuencia de Ingresar Técnica de Evaluación.
Fuente: Elaboración Propia.

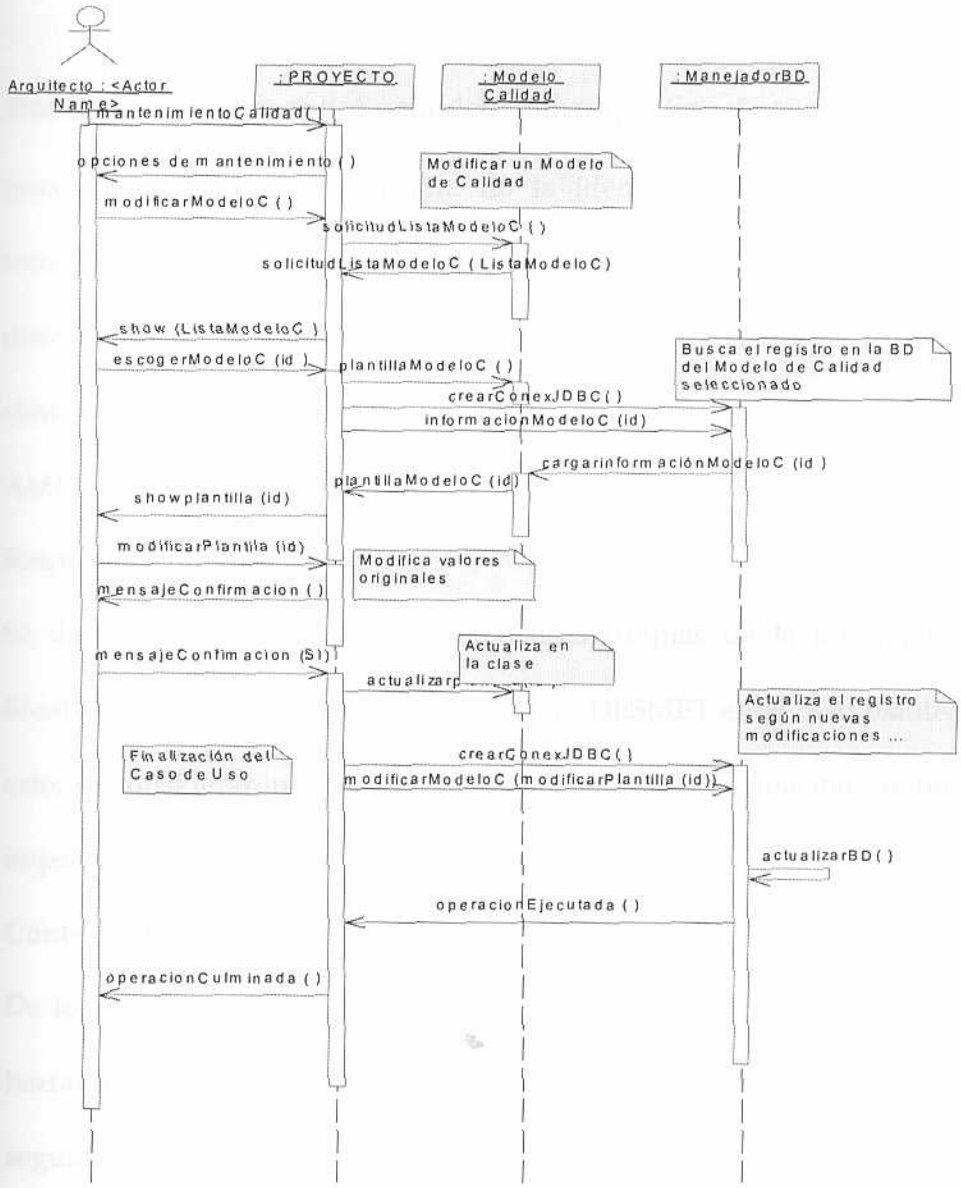

Figura M.11. Diagrama de Secuencia de Modificar Modelo de Calidad.
Fuente: Elaboración Propia.

APÉNDICE N - Identificación de Requerimientos DESMET

Esta metodología se aplicó para determinar que tipo de evaluación es propicia para este tipo de investigación. En la identificación de requerimientos se tomaron los técnicos y prácticos para la determinación del método según la descripción de la Metodología en el Anexo D - *Metodología de DESMET*. A continuación se muestra los requerimientos que fueron identificados para AMEVA.

Requerimientos Técnicos

Se definieron una serie de características propias de la investigación, para identificar qué tipo de evaluación sugiere DESMET en el caso planteado. Para esto, se consideraron aspectos como contexto de la evaluación, naturaleza del impacto, alcance del impacto, entre otros.

Contexto de Evaluación

De los cuatro contextos existentes, se consideró que "Selección de métodos y herramientas para una organización que pueden proyectar alternativas iniciales seguidas de una evaluación más detallada de uno o mas métodos o herramientas", es el contexto más apropiado para esta investigación. La selección de métodos y herramientas para una organización pueden proyectar alternativas iniciales seguidas de una evaluación más detallada del ambiente propuesto (modelo conceptual) .

Naturaleza del Impacto

La naturaleza del impacto fue considerada como "Cualitativa", ya que si bien la evaluación debería estar apoyada en mediciones cuantitativas, la mayor

importancia de dichas mediciones se fundamentó en un análisis cualitativo de la presencia o no de las características que se definieron al evaluar.

Naturaleza del Objeto de Evaluación

Se consideró al ambiente como una combinación herramienta / métodos. Si el objeto es una herramienta soportada por varios métodos, DESMET sugiere que sea evaluado dicho objeto como un método.

Alcance del Impacto

El alcance del impacto se consideró como "Extensión del Impacto", ya que aquí se identifica al ambiente (modelo conceptual) que puede causar el efecto de un filtro sobre el ciclo de vida del proyecto o del producto.

Esta opción es la que se adecua al fin del ambiente, ya que éste sería parte esencial del ciclo de vida del producto de software. En este contexto, DESMET asume que el desarrollo de un producto envuelve cierto número de etapas (adaptables a cualquier ciclo de vida), y que en la práctica estas actividades suelen solaparse; también afirma que existen métodos y herramientas que pueden ser específicas a una sola etapa. Esto hace que DESMET identifique, para ayudar a esclarecer con una mayor profundidad el alcance del impacto, para esto propone cinco opciones, de las cuales la seleccionada fue "Restringido a una actividad asociada primariamente con una etapa específica y observable / cuantificable inmediatamente a la culminación de la actividad", ya que la intención del ambiente es evaluar la arquitectura y el diseño de la misma abarca solo una etapa del ciclo de vida.

Madurez del Ambiente

"No usado en proyectos comerciales, todavía en desarrollo" es el más

apropiado, dado que aún no se encuentra el ambiente en la etapa funcional, por lo cual se considera un producto inmaduro.

Tiempo de Aprendizaje

La opción "Tiempo requerido para entender los principios de la arquitectura" fue la seleccionada, ya que el tiempo requerido es simplemente el necesario para comprender cómo se especificó la arquitectura del ambiente.

Madurez de la Organización

La madurez de la organización fue seleccionada bajo "Capacidad de Evaluación cualitativa y cuantitativa", ya que LISI tiene estándares para el desarrollo de software definidos. A pesar de esto, pudieran no ser considerados definidos ya que la manera de aplicar dichos estándares entre los distintos profesores que conforman el LISI, no está restringida al hecho de seguir paso a paso un conjunto de instrucciones previamente establecidas en el Departamento para la aplicación de los mismos. En tal sentido, se pueden conseguir dos instancias de un estándar para el desarrollo de software que difieran por el toque personal que pudieran haber agregado sus participantes.

A continuación se presenta un breve resumen de las conclusiones que se han logrado obtener tras el análisis de los tres requerimientos técnicos para la selección del Método de DESMET.

En cuanto a los requerimientos técnicos, se pudo observar que en general la evaluación apunta a la cualitativa. Por lo tanto, para definir cuál evaluación cualitativa se debía aplicar, se identificaron los requerimientos prácticos, ya que son complemento de la aplicación de DESMET. En la tabla N.1. Resumen de Requerimientos Técnicos de los Métodos de Evaluación, que se muestra a

continuación, se encuentra un resumen de los requerimientos técnicos identificados.

Tabla N.1. Resumen de Requerimientos Técnicos de los Métodos de Evaluación.
Fuente: Elaboración Propia.

Requerimiento	Tipo	Método Recomendado
Contexto de Evaluación	Selección de Métodos y Herramientas para una organización que pueden proyectar alternativas iniciales seguidas de una evaluación más detallada de uno o más métodos o herramientas	Ninguno en especial
Naturaleza Del Impacto	Cualitativa	Análisis de Características en: Estudio de un Caso Experimento Modo Pantalla Medición Análisis de Efectos Cualitativos
Naturaleza del Objeto de Evaluación	Comparaciones de herramienta soportada por diferentes métodos	Evaluación Cuantitativa
Alcance del Impacto	Impacto extendido que puede que puede afectar una o varias etapas del ciclo de desarrollo	Cualquier Cualitativo Estudio de un Caso Experimento
Madurez Del Método	No usado en proyectos comerciales, todavía en desarrollo	Resumen Cuantitativo Resumen Cualitativo
Tiempo de Aprendizaje	Tiempo requerido para entender los principios de la arquitectura	No Cuantitativo
Madurez de la Organización	Capacidad de Evaluación cualitativa y cuantitativa	Ninguno en especial

Requerimientos Prácticos

En cuanto a los requerimientos prácticos, se estudiaron la confianza de los resultados, los tiempos disponibles y los costos de la evaluación. También se estudiaron las características de los métodos y se identificó cuál era el que se ajustaba al tipo de investigación realizada.

A continuación se presentan los requerimientos prácticos identificados.

Tabla N.2. Características de los métodos de Evaluación de DESMET para su selección. Fuente: Elaboración Propia.

Método de Evaluación	Características Favorables	Características No Favorables	% Fav.
Experimentos Cuantitativos	• Tiempo de aprendizaje relativamente pequeño. • Personal disponible para tomar parte del experimento. • Deseo hacer contribuciones de contexto independiente al método.	• Método relacionado con una pregunta sencilla. • Beneficios directamente observables desde los resultados de las tareas. • Beneficios claramente cuantificables.	50%
Casos de Estudio Cuantitativo	• Procedimientos de desarrollo estables. • Personal con experiencia métrica. • Tiempo para la evaluación proporcional al tiempo transcurrido del proyecto de tamaño normal.	• Beneficios cuantificables previo al retiro del producto. • Beneficios cuantificables en un proyecto sencillo.	60%
Resumen Cuantitativo	• Beneficios no cuantificables en un solo proyecto.	• Existencia de una base de datos del proyecto con información de productividad, calidad, información de métodos. • Proyectos con experiencia de uso del método.	33%
Selección Cualitativa		• Gran número de métodos a evaluar. • Tiempo corto para el ejercicio de evaluación	0%
Caso de Estudio Cualitativo	• Beneficios difíciles de cuantificar. • Procedimientos de desarrollo estables. • Tiempo para la evaluación proporcional al tiempo transcurrido del proyecto de tamaño normal.	• Beneficios observables en un simple proyecto.	75%
Experimento Cualitativo	• Beneficios difíciles de cuantificar. • Tiempo de evaluación pequeño.	• Población de usuarios del método variada. • Beneficios observables en un proyecto sencillo.	50%
Resumen Cualitativo	• Beneficios difíciles de cuantificar. • Beneficios no observables en un proyecto sencillo.	• Población de usuarios del método variada. • Proyectos realizados para aprender del método.	50%
Análisis de Efectos Cualitativos	• Requerimiento de unión de métodos. • Interés en evaluación de métodos genéricos. • Disposición de opiniones expertas acerca del método.	• Carencia de estabilidad en el proceso de desarrollo.	75%
Benchmarking		• Método no intensivo al nivel humano. • Salidas del método capaz de ser comparadas con un ranking en términos de qué tan buenos son.	0%

Con respecto a los tiempos asociados, se identificó el tiempo como largo, dado al seguimiento que se ha hecho durante la investigación.

Tabla N.3. Tiempo Relativo asociado a la investigación.
Fuente: Elaboración Propia.

Riesgo Relativo	Método de Evaluación	Comentarios
Largo (Más de 3 Meses)	Estudio de un Caso (Cuantitativo y Análisis de Características)	Un estudio de un Caso puede durar lo que dura el desarrollo del proyecto.

Con respecto a la confianza de los resultados, se determinó con la magnitud del riesgo asociado a la investigación.

Tabla N.4. Riesgos asociados a la investigación.
Fuente: Elaboración Propia.

Riesgo Relativo	Método de Evaluación	Comentarios
Alto	Estudio de un Caso Cuantitativo	La evaluación del impacto está basada en una comparación de dos proyectos diferentes.
	Análisis de Características, Estudio de un Caso o Modo de Proyección	La evaluación esta basada e la evaluación de una persona y el criterio de evaluación es subjetivo.

Con respecto a los costos asociados, se presenta los seleccionados.

Tabla N.5. Costos asociados con la investigación.
Fuente: Elaboración Propia.

Riesgo Relativo	Método de Evaluación	Comentarios
Medio	Estudio de un Caso	Los costos estarán relacionados más que todo con los trabajos adicionales para realizar las evaluaciones.
	Análisis de las Características, Medición	Los costos de introducir, enviar y recibir una medición es algo que requiere su esfuerzo, a lo se une el tiempo y costo derivado del entendimiento de aquella persona que debe rellenar el cuestionario.
	Análisis de las Características. Modo Proyección	Esto puede ser realizado por una persona durante unas semanas. El costo se verá incrementado si es necesario entender una especificación de requerimientos detallados de los requerimientos de los usuarios potenciales de dichos métodos.

Considerando la escala de tiempos relativos se nota que si bien se dispone solamente de esta última fase de la investigación para realizar la evaluación

según DESMET, se ha trabajado desde la primera fase en el entendimiento, análisis, comprensión y diferenciación, de los distintos enfoques en los que se basa cada uno de los casos de estudio. El tiempo de esta fase fue considerado mayor a tres meses, lo que por recomendación de DESMET corresponde a usar como método de evaluación: Estudio de un Caso, ya sea de manera cuantitativa o a través de un Análisis de Características.

En cuanto a la confiabilidad de los resultados, el hecho de la evaluación se realizó por una sola persona, lo que infiere criterios de evaluación subjetivos. En tal sentido, DESMET sugiere el Análisis de Características en el Estudio de un Caso. También es importante indicar, que esta evaluación incluye el impacto de los casos de estudio sobre la investigación, donde DESMET sugiere la aplicación del Estudio de un Caso Cuantitativo. Para ambos métodos se representa un riesgo de magnitud alta.

Con respecto a los costos de la evaluación, a pesar de que esta investigación no acarrea costo alguno, se determinó que los costos están relacionados más que todo con los trabajos adicionales para realizar las evaluaciones que corresponde según DESMET al Estudio de un Caso, al igual que la evaluación puede ser realizada por una persona durante unas semanas y cuyos costos se ven incrementados cuando es necesario entender una especificación de requerimientos detallados de los usuarios potenciales. También, se enfocó hacia el esfuerzo, costo y tiempo que puede representar para la persona aplicar dicha evaluación, por lo que DESMET sugiere el método de Análisis de Características, Medición.

Si se aprecian las coincidencias de los métodos sugeridos, tras el análisis de los

requerimientos prácticos, en la selección del Método de Evaluación de DESMET que mejor se adapta a la evaluación, se puede notar una reiteración de un Análisis de Características por el Estudio de un Caso.

A continuación se presentan las conclusiones que se lograron obtener tras el análisis de los tres requerimientos prácticos para la selección del Método de DESMET.

Tabla N.6. Resumen de Requerimientos Prácticos de los Métodos de Evaluación. Fuente: Elaboración Propia.

Criterio Práctico	Método de Evaluación Sugerido
Tiempo disponible	Estudio de un Caso: cuantitativa o a través de Análisis de Características.
Confianza de los resultados	Análisis de Características en el Estudio de un Caso.
Costos de evaluación	Análisis Característico. Estudio de un Caso.

El resultado de aplicar DESMET determinó que tanto los requerimientos técnicos de esta investigación como los requerimientos prácticos, pueden ser satisfechos por la evaluación cualitativa. Por otra parte, en la tabla N.2. "Características de los métodos de Evaluación de DESMET para su selección" se pudo observar claramente que se obtuvo la mayor puntuación (75%) para los métodos Caso de Estudio Cualitativo y Análisis de Efectos Cualitativos de lo que se interpretó, que el mejor método que se adapta a los requerimientos establecidos a esta investigación es la aplicación de Análisis de Características del Estudio de un Caso, dada su practicidad y el antecedente de los casos de estudio. Aunado a esto, tuvo mayor ocurrencia dentro de los resultados el Estudio de un Caso. No se aplicó el método de Análisis de Efectos Cualitativos, dado que su evaluación incluye juicio de expertos.

mantenimiento, instrumentación entre otros, ya que para estos, se necesita un nivel de abstracción muy alto que no ofrece el modelo conceptual.

También fue necesario aplicar métricas para la valoración de las características, donde el grado de ayuda ofrecido se debe medir o juzgar en una escala ordinal, del cero (0) al cinco (5).

A continuación se muestran las características definidas para la validación, donde se presenta su nombre, definición conceptual y definición operacional.

Tabla O.1. Característica Completitud. Fuente: Elaboración Propia.

Característica	Completitud
Definición conceptual	Es el grado en que se ha logrado definir de forma clara y concisa todos los conceptos referentes al dominio del problema y sus relaciones relevantes de jerarquía, asociación y agregación.
Definición Operacional	
01. Presencia de todos los conceptos o entidades de información requeridos.	
02. Precisión conceptual de términos	
03. Completitud de relaciones entre conceptos.	
04. Niveles de cardinalidad.	
05. Definición de requisitos que justifican la funcionalidad.	
06. Presencia de sistemas externos con los que debe interactuar.	

Tabla O.2. Característica Consistencia. Fuente: Elaboración Propia.

Característica	Consistencia
Definición conceptual	Es el grado en que las entidades representan de forma única y no contradictoria los conceptos del dominio del problema.
Definición operacional	
07. Correspondencia de conceptos definidos en el diagrama.	
08. Correspondencia en la nomenclatura empleada.	
09. Consistencia referida al principio de instanciación (subconjuntos de funciones).	
10. Complementabilidad de los conceptos con su subconjunto.	
11. Consistencia en las relaciones establecidas.	
12. No redundancia en las relaciones definidas.	
13. Niveles de abstracción.	

Tabla O.3. Característica Correctitud. Fuente: Elaboración Propia.

Característica	Correctitud
Definición conceptual	Es el grado de claridad en la presentación de los elementos que conforman el modelo conceptual.
Definición operacional	
14. Comprensión de conceptos y relaciones.	
15. Correctitud jerárquica.	
16. Representación de la funcionalidad definida por la interacción de los conceptos.	
17. Representación de la funcionalidad en general con respecto al modelo conceptual.	

Tabla O.4. Característica Complejidad. Fuente: Elaboración Propia.

Característica	Complejidad
Definición conceptual	Es el grado de claridad en la presentación de los elementos que describen el contexto y la funcionalidad del sistema.
Definición operacional	
18. Presencia del encapsulamiento de conceptos.	
19. Presencia de modularidad.	
20. Número de elementos en el modelo adecuado a los conceptos establecidos.	
21. Existencia de elementos que no contribuyen al entendimiento del modelo conceptual.	
22. Existencia de relaciones que no contribuyen al entendimiento del modelo conceptual.	

Para la medición de las características, se utilizó la escala ordinal mostrada a continuación:

Tabla O.5. Escala Ordinal de las Métricas. Fuente: (Casal, 2003).

Valor otorgado	Descripción	Definición
5	Excelente	La evaluación de la característica supera y sobresale las expectativas.
4	Muy buena	La evaluación de la característica supera las expectativas.
3	Buena	La evaluación de la característica cubre todas las expectativas.
2	Deficiente	La evaluación de la característica no cubre las expectativas.
1	Muy deficiente	La evaluación de la característica no cubre las expectativas y está lejos de hacerlo.
0	No aplica	La evaluación de la característica no se puede realizar ya que no aplica para la característica evaluada.

Selección de los Proyectos Piloto

Estos casos de estudio son los presentados como antecedentes, y aunado a estos, se incorporó el modelo conceptual de AMEVA especificado en la Vista Lógica de la Fase de Elaboración. El proyecto piloto considerado para

A.T.A.M. es el Proyecto DID-KMS; para M.E.L. también se usó la aplicación del Proyecto DID-KMS. El caso que representa el método de Bosch es el Proyecto del Call Center y la Casa de Futuro fue el proyecto seleccionado del método A.B.D.M.. en total fueron cinco casos de estudios para esta etapa de DESMET. Para mayor detalle ver el Apéndice C - *Casos de Estudio.*

Prueba en cada Proyecto Piloto

La prueba consistió en verificar la existencia de las características propuestas en los modelos conceptuales y calificar con una escala del 0 al 5 de acuerdo a las ocurrencia o no dentro del modelo.

Evaluación de las Características en cada Caso de Estudio

Se aplicó la evaluación según las características propuestas a los modelos conceptuales de los casos de estudio. A continuación se presenta la aplicación a cada caso de estudio.

Caso de Estudio 1. Sistema de Gestión del Conocimiento para Proyectos de Investigación. (Aplicación de A.T.A.M.)

Tabla O.6. Característica Completitud (A.T.A.M.).Fuente: Elaboración Propia.

Característica	Completitud
Definición conceptual	Es el grado en que se ha logrado definir de forma clara y concisa todos los conceptos referentes al dominio del problema y sus relaciones relevantes de jerarquía, asociación y agregación.

Definición Operacional	Puntuación
01. Presencia de todos los conceptos o entidades de información requeridos.	4
02. Precisión conceptual de términos	5
03. Completitud de relaciones entre conceptos.	5
04. Niveles de cardinalidad.	5
05. Definición de requisitos que justifican la funcionalidad.	4
06. Presencia de sistemas externos con los que debe interactuar.	4
SUMA	27

Tabla O.7. Característica Consistencia (A.T.A.M.). Fuente: Elaboración Propia.

Característica	Consistencia
Definición conceptual	Es el grado en que las entidades representan de forma única y no contradictoria los conceptos del dominio del problema.
Definición operacional	Puntuación
07. Correspondencia de conceptos definidos en el diagrama.	5
08. Correspondencia en la nomenclatura empleada.	5
09. Consistencia referida al principio de instanciación (subconjuntos de funciones).	5
10. Complementabilidad de los conceptos con su subconjunto.	4
11. Consistencia en las relaciones establecidas.	5
12. No redundancia en las relaciones definidas.	5
13. Niveles de abstracción.	0
SUMA	29

Tabla O.8. Característica Correctitud (A.T.A.M.). Fuente: Elaboración Propia.

Característica	Correctitud
Definición conceptual	Es el grado de claridad en la presentación de los elementos que conforman el modelo conceptual.
Definición operacional	Puntuación
14. Comprensión de conceptos y relaciones.	5
15. Correctitud jerárquica.	4
16. Representación de la funcionalidad definida por la interacción de los conceptos.	4
17. Representación de la funcionalidad en general con respecto al modelo conceptual.	5
SUMA	18

Tabla O.9. Característica Complejidad (A.T.A.M.). Fuente: Elaboración Propia.

Característica	Complejidad
Definición conceptual	Es el grado de claridad en la presentación de los elementos que describen el contexto y la funcionalidad del sistema.
Definición operacional	Puntuación
18. Presencia de la encapsulamiento de conceptos.	5
19. Presencia de modularidad.	3
20. Número de elementos en el modelo adecuado a los conceptos establecidos.	5
21. Existencia de elementos que no contribuyen al entendimiento del modelo conceptual.	0
22. Existencia de relaciones que no contribuyen al entendimiento del modelo conceptual.	0
SUMA	13

Caso de Estudio 2. Sistema de Gestión del Conocimiento para Proyectos de Investigación. (Aplicación de M.E.L.)

Tabla O.10. Característica Completitud (M.E.L.). Fuente: Elaboración Propia.

Característica	Completitud	
Definición conceptual	Es el grado en que se ha logrado definir de forma clara y concisa todos los conceptos referentes al dominio del problema y sus relaciones relevantes de jerarquía, asociación y agregación.	
Definición Operacional		Puntuación
01. Presencia de todos los conceptos o entidades de información requeridos.		5
02. Precisión conceptual de términos		5
03. Completitud de relaciones entre conceptos.		5
04. Niveles de cardinalidad.		5
05. Definición de requisitos que justifican la funcionalidad.		4
06. Presencia de sistemas externos con los que debe interactuar.		4
	SUMA	28

Tabla O.11. Característica Consistencia (M.E.L.). Fuente: Elaboración Propia.

Característica	Consistencia	
Definición conceptual	Es el grado en que las entidades representan de forma única y no contradictoria los conceptos del dominio del problema.	
Definición operacional		Puntuación
07. Correspondencia de conceptos definidos en el diagrama.		5
08. Correspondencia en la nomenclatura empleada.		5
09. Consistencia referida al principio de instanciación (subconjuntos de funciones).		4
10. Complementabilidad de los conceptos con su subconjunto.		4
11. Consistencia en las relaciones establecidas.		5
12. No redundancia en las relaciones definidas.		5
13. Niveles de abstracción.		0
	SUMA	28

Tabla O.12. Característica Correctitud (M.E.L.). Fuente: Elaboración Propia.

Característica	Correctitud	
Definición conceptual	Es el grado de claridad en la presentación de los elementos que conforman el modelo conceptual.	
Definición operacional		Puntuación
14. Comprensión de conceptos y relaciones.		4
15. Correctitud jerárquica.		5
16. Representación de la funcionalidad definida por la interacción de los conceptos.		4
17. Representación de la funcionalidad en general con respecto al modelo conceptual.		4
	SUMA	17

Tabla O.13. Característica Complejidad (M.E.L.). Fuente: Elaboración Propia.

Característica	Complejidad
Definición conceptual	Es el grado de claridad en la presentación de los elementos que describen el contexto y la funcionalidad del sistema.
Definición operacional	Puntuación
18. Presencia de la encapsulamiento de conceptos.	3
19. Presencia de modularidad.	4
20. Número de elementos en el modelo adecuado a los conceptos establecidos.	4
21. Existencia de elementos que no contribuyen al entendimiento del modelo conceptual.	0
22. Existencia de relaciones que no contribuyen al entendimiento del modelo conceptual.	0
SUMA	11

Caso de Estudio 3. Sistema para la Casa del Futuro . (Aplicación de A.B.D.M.)

Tabla O.14. Característica Completitud (A.B.D.M.). Fuente: Elaboración Propia.

Característica	Completitud
Definición conceptual	Es el grado en que se ha logrado definir de forma clara y concisa todos los conceptos referentes al dominio del problema y sus relaciones relevantes de jerarquía, asociación y agregación.
Definición Operacional	Puntuación
01. Presencia de todos los conceptos o entidades de información requeridos.	5
02. Precisión conceptual de términos	4
03. Completitud de relaciones entre conceptos.	4
04. Niveles de cardinalidad.	5
05. Definición de requisitos que justifican la funcionalidad.	3
06. Presencia de sistemas externos con los que debe interactuar.	0
SUMA	21

Tabla O.15. Característica Consistencia (A.B.D.M.). Fuente: Elaboración Propia.

Característica	Consistencia
Definición conceptual	Es el grado en que las entidades representan de forma única y no contradictoria los conceptos del dominio del problema.
Definición operacional	Puntuación
07. Correspondencia de conceptos definidos en el diagrama.	5
08. Correspondencia en la nomenclatura empleada.	5
09. Consistencia referida al principio de instanciación (subconjuntos de funciones).	3
10. Complementabilidad de los conceptos con su subconjunto.	4
11. Consistencia en las relaciones establecidas.	4
12. No redundancia en las relaciones definidas.	4
13. Niveles de abstracción.	0
SUMA	25

Tabla O.16. Característica Correctitud (A.B.D.M.). Fuente: Elaboración Propia.

Característica	Correctitud
Definición conceptual	Es el grado de claridad en la presentación de los elementos que conforman el modelo conceptual.
Definición operacional	Puntuación
14. Comprensión de conceptos y relaciones.	4
15. Correctitud jerárquica.	4
16. Representación de la funcionalidad definida por la interacción de los conceptos.	4
17. Representación de la funcionalidad en general con respecto al modelo conceptual.	4
SUMA	16

Tabla O.17. Característica Complejidad (A.B.D.M.). Fuente: Elaboración Propia.

Característica	Complejidad
Definición conceptual	Es el grado de claridad en la presentación de los elementos que describen el contexto y la funcionalidad del sistema.
Definición operacional	Puntuación
18. Presencia de la encapsulamiento de conceptos.	4
19. Presencia de modularidad.	4
20. Número de elementos en el modelo adecuado a los conceptos establecidos.	4
21. Existencia de elementos que no contribuyen al entendimiento del modelo conceptual.	0
22. Existencia de relaciones que no contribuyen al entendimiento del modelo conceptual.	0
SUMA	12

Caso de Estudio 4. Sistema de Emergencias, Call Center. (Aplicación de Bosch)

Tabla O.18. Característica Completitud (Bosch). Fuente: Elaboración Propia.

Característica	Completitud
Definición conceptual	Es el grado en que se ha logrado definir de forma clara y concisa todos los conceptos referentes al dominio del problema y sus relaciones relevantes de jerarquía, asociación y agregación.
Definición Operacional	Puntuación
01. Presencia de todos los conceptos o entidades de información requeridos.	5
02. Precisión conceptual de términos	5
03. Completitud de relaciones entre conceptos.	5
04. Niveles de cardinalidad.	5
05. Definición de requisitos que justifican la funcionalidad.	4
06. Presencia de sistemas externos con los que debe interactuar.	0
SUMA	24

Tabla O.19. Característica Consistencia (Bosch). Fuente: Elaboración Propia.

Característica	Consistencia	
Definición conceptual	Es el grado en que las entidades representan de forma única y no contradictoria los conceptos del dominio del problema.	
Definición operacional		Puntuación
07. Correspondencia de conceptos definidos en el diagrama.		5
08. Correspondencia en la nomenclatura empleada.		5
09. Consistencia referida al principio de instanciación (subconjuntos de funciones).		4
10. Complementabilidad de los conceptos con su subconjunto.		4
11. Consistencia en las relaciones establecidas.		5
12. No redundancia en las relaciones definidas.		5
13. Niveles de abstracción.		0
	SUMA	28

Tabla O.20. Característica Correctitud (Bosch). Fuente: Elaboración Propia.

Característica	Correctitud	
Definición conceptual	Es el grado de claridad en la presentación de los elementos que conforman el modelo conceptual.	
Definición operacional		Puntuación
14. Comprensión de conceptos y relaciones.		5
15. Correctitud jerárquica.		5
16. Representación de la funcionalidad definida por la interacción de los conceptos.		5
17. Representación de la funcionalidad en general con respecto al modelo conceptual.		5
	SUMA	20

Tabla O.21 Característica Complejidad (Bosch). Fuente: Elaboración Propia.

Característica	Complejidad	
Definición conceptual	Es el grado de claridad en la presentación de los elementos que describen el contexto y la funcionalidad del sistema.	
Definición operacional		Puntuación
18. Presencia de la encapsulamiento de conceptos.		5
19. Presencia de modularidad.		4
20. Número de elementos en el modelo adecuado a los conceptos establecidos.		5
21. Existencia de elementos que no contribuyen al entendimiento del modelo conceptual.		0
22. Existencia de relaciones que no contribuyen al entendimiento del modelo conceptual.		0
	SUMA	14

Caso de Estudio 5. Ambiente de Evaluación de Arquitecturas de Software, AMEVA.

Tabla O.22. Característica Completitud (AMEVA). Fuente: Elaboración Propia.

Característica	Completitud
Definición conceptual	Es el grado en que se ha logrado definir de forma clara y concisa todos los conceptos referentes al dominio del problema y sus relaciones relevantes de jerarquía, asociación y agregación.
Definición Operacional	Puntuación
01. Presencia de todos los conceptos o entidades de información requeridos.	4
02. Precisión conceptual de términos	5
03. Completitud de relaciones entre conceptos.	5
04. Niveles de cardinalidad.	5
05. Definición de requisitos que justifican la funcionalidad.	5
06. Presencia de sistemas externos con los que debe interactuar.	5
SUMA	29

Tabla O.23. Característica Consistencia (AMEVA). Fuente: Elaboración Propia.

Característica	Consistencia
Definición conceptual	Es el grado en que las entidades representan de forma única y no contradictoria los conceptos del dominio del problema.
Definición operacional	Puntuación
07. Correspondencia de conceptos definidos en el diagrama.	5
08. Correspondencia en la nomenclatura empleada.	5
09. Consistencia referida al principio de instanciación (subconjuntos de funciones).	4
10. Complementabilidad de los conceptos con su subconjunto.	5
11. Consistencia en las relaciones establecidas.	5
12. No redundancia en las relaciones definidas.	5
13. Niveles de abstracción.	0
SUMA	29

Tabla O.24. Característica Correctitud (AMEVA). Fuente: Elaboración Propia.

Característica	Correctitud
Definición conceptual	Es el grado de claridad en la presentación de los elementos que conforman el modelo conceptual.
Definición operacional	Puntuación
14. Comprensión de conceptos y relaciones.	5
15. Correctitud jerárquica.	5
16. Representación de la funcionalidad definida por la interacción de los conceptos.	4
17. Representación de la funcionalidad en general con respecto al modelo conceptual.	4
SUMA	18

Tabla O.25. Característica Complejidad (AMEVA). Fuente: Elaboración Propia.

Característica	Complejidad
Definición conceptual	Es el grado de claridad en la presentación de los elementos que describen el contexto y la funcionalidad del sistema.
Definición operacional	Puntuación
18. Presencia de la encapsulamiento de conceptos.	5
19. Presencia de modularidad.	5
20. Número de elementos en el modelo adecuado a los conceptos establecidos.	5
21. Existencia de elementos que no contribuyen al entendimiento del modelo conceptual.	0
22. Existencia de relaciones que no contribuyen al entendimiento del modelo conceptual.	0
SUMA	15

Análisis de la Puntuación y Presentación de Resultados

En base a los resultados obtenidos en la evaluación aplicada a los casos de estudio se observó lo siguiente:

Tabla O.26. Totales de cada Característica. Fuente: Elaboración Propia.

Caso de Estudio	Completitud	Consistencia	Correctitud	Complejidad
ATAM	27	29	18	13
MEL	28	28	17	11
ABDM	21	25	16	12
Bosch	24	28	20	14
AMEVA	29	29	18	15

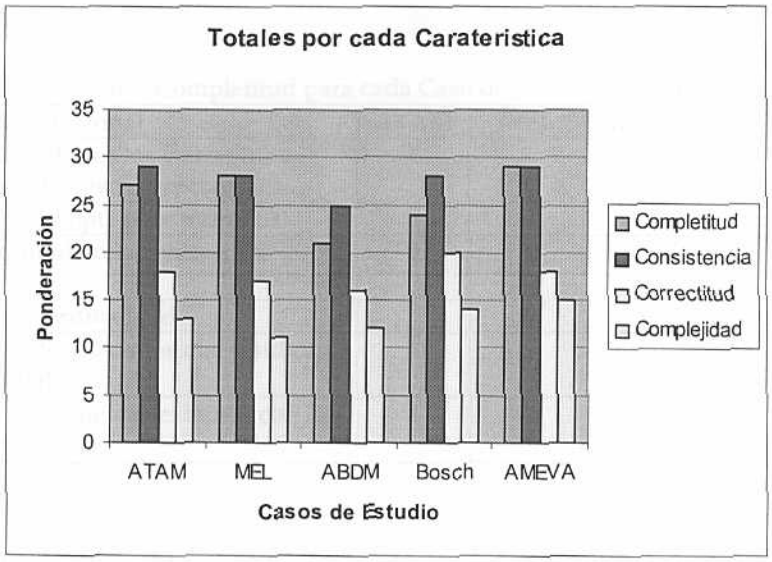

Figura O.1. Totales por Características.
Fuente: Elaboración Propia.

Los valores correspondientes a los totales por cada característica en los casos de estudio presentados en a tabla O.26. Totales de cada Característica, muestran una madurez debido a que obtuvieron una calificación semejante. Esto implica, que los modelos conceptuales de cada caso se encuentran especificados de una forma coherente y comprensible. Desde el punto de vista cuantitativo, sobre la base de los puntajes obtenidos por cada caso de estudio y aunque no todos obtuvieron el mayor puntaje para cada característica, los valores presentados satisfacen la evaluación.

Para determinar, en cuáles métricas se demostró debilidad por parte de los casos de estudio, se analizó cada característica en particular.

Para la característica Completitud se determinó la puntuación en cada caso de estudio como se muestra a continuación:

Tabla O.27. Característica Completitud para cada Caso de Estudio. Fuente: Elaboración Propia.

Definición Operacional	ATAM	MEL	ABDM	BOSCH	AMEVA
01. Presencia de todos los conceptos o entidades de información requeridos.	4	5	5	5	4
02. Precisión conceptual de términos	5	5	4	5	5
03. Completitud de relaciones entre conceptos.	5	5	4	5	5
04. Niveles de cardinalidad.	5	5	5	5	5
05. Definición de requisitos que justifican la funcionalidad.	4	4	3	4	5
06. Presencia de sistemas externos con los que debe interactuar.	4	4	0	0	5

En la figura O.2. Característica Completitud para cada Caso de Estudio, se encuentra representada mediante una gráfica los valores obtenidos por cada caso de estudio para operación propuesta.

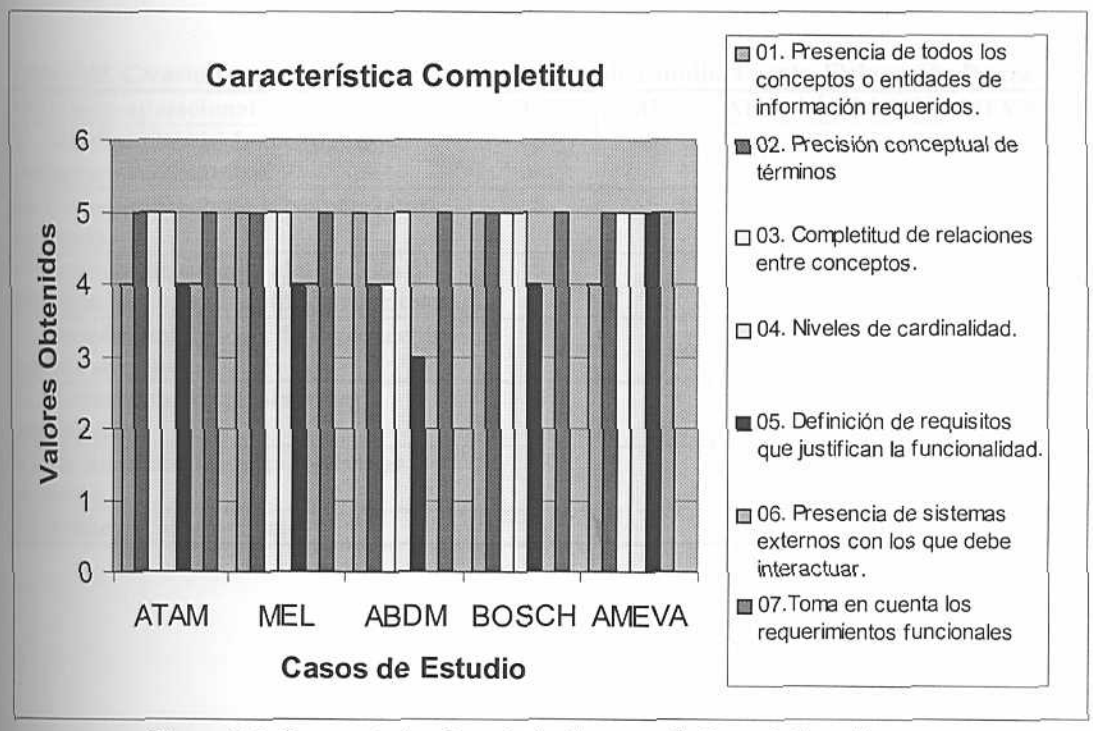

Figura O.2. Característica Completitud para cada Caso de Estudio.
Fuente: Elaboración Propia.

En esta característica, se presentó uniformidad en cuanto a los resultados obtenidos. La métrica "Presencia de sistemas externos con los que se debe

interactuar", se vio reflejada solo el caso de estudio del modelo conceptual de la aplicación de ATAM y el de MEL para el caso del modelo de especificación de la calidad, dado que se instanció el modelo de Ortega y como es un conjunto de métricas se puede considerar un sistema; y en AMEVA, por el hecho de incluir un ADL en su conceptualización, y los diferentes modelos de especificación de la calidad que se puedan instanciar. Para esta característica, AMEVA presento la puntuación más elevada, lo cual significa que satisface las necesidades plantadas mejor que los otros casos evaluados.

En cuanto a la Característica Consistencia, se colocaron en la tabla que se encuentra a continuación, acompañada del gráfico respectivo.

Tabla O.28. Característica Consistencia para cada Caso de Estudio. Fuente: Elaboración Propia.

Definición operacional	ATAM	MEL	ABDM	BOSCH	AMEVA
07. Correspondencia de conceptos definidos en el diagrama.	5	5	5	5	5
08. Correspondencia en la nomenclatura empleada.	5	5	5	5	5
09. Consistencia referida al principio de instanciación (subconjuntos de funciones).	5	4	3	4	4
10. Complementabilidad de los conceptos con su subconjunto.	4	4	4	4	5
11. Consistencia en las relaciones establecidas.	5	5	4	5	5
12. No redundancia en las relaciones definidas.	5	5	4	5	5
13. Niveles de abstracción.	0	0	0	0	0

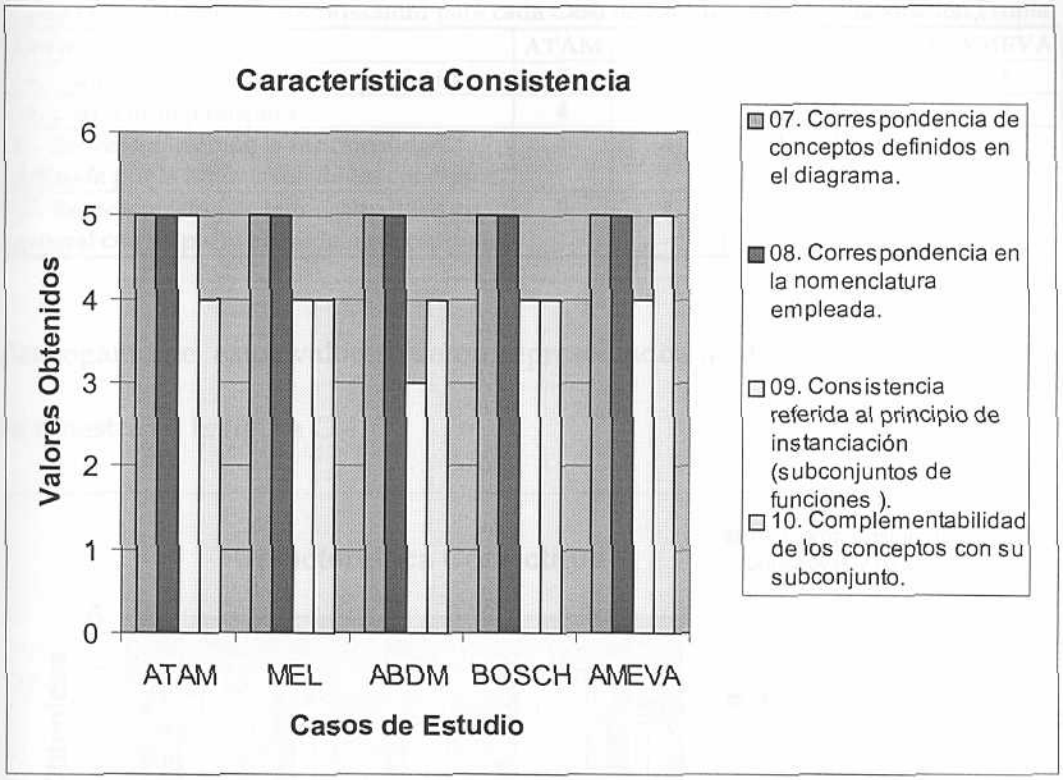

Figura O.3. Característica Consistencia para cada Caso de Estudio.
Fuente: Elaboración Propia.

Con respecto a la característica Consistencia, también se mostraron valores similares. Para la métrica "Consistencia referida al principio de instanciación (subconjunto de funciones)", el caso particular del modelo conceptual de ABDM, obtuvo la ponderación más baja debido a que no existe instanciación alguna que permita introducir o validar bajo un estándar los requerimientos de calidad. La métrica "Niveles de Abstracción" no se hizo presente en ninguno de los casos mostrados, dado que se valuaron modelos conceptuales. A pesar de llegar hasta cierto nivel de detalle, esta modalidad de diagramas permite solo aplicar una abstracción parcial del dominio del problema o si se quiere, de la realidad.

Los valores obtenidos para la Característica Correctitud, se encuentran en la tabla O.9. Característica Correctitud para cada Caso de Estudio.

Tabla O.29. Característica Correctitud para cada Caso de Estudio. Fuente: Elaboración Propia.

Definición operacional	ATAM	MEL	ABDM	BOSCH	AMEVA
14. Comprensión de conceptos y relaciones.	5	4	4	5	5
15. Correctitud jerárquica.	4	5	4	5	5
16. Representación de la funcionalidad definida por la interacción de los conceptos.	4	4	4	5	4
17. Representación de la funcionalidad en general con respecto al modelo conceptual.	5	4	4	5	4

Análogamente, estos valores fueron representados mediante una gráfica como se muestra en la figura O.4.

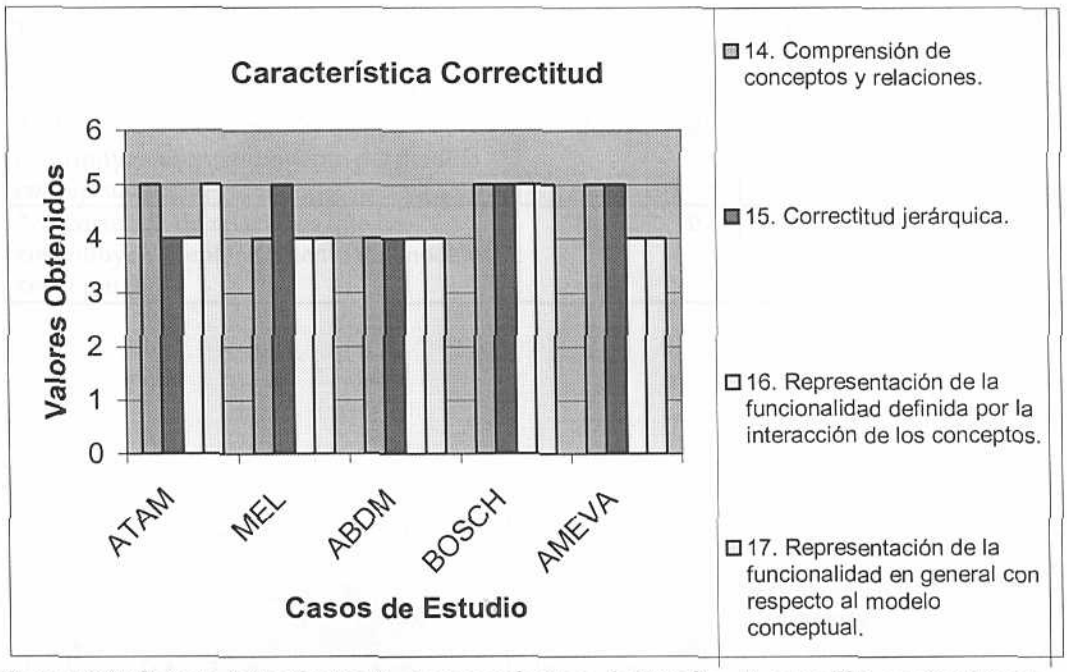

Figura O.4. Característica Correctitud para cada Caso de Estudio. Fuente: Elaboración Propia.

De los valores obtenidos para la característica Correctitud, se apreció análogamente como en los casos anteriores, la homogeneidad en las métricas propuestas. Para el modelo conceptual que representa el caso de estudio de la aplicación de Bosch, los valores obtenidos superaron las expectativas de la evaluación. La presencia de la métrica "Correctitud Jerárquica" en AMEVA en eminente dado a la estructura diseñada en el modelo conceptual. Para los otros

casos, esta métrica se encuentra presente pero no de manera tan obvia como el caso antes mencionado.

Finalmente, la Característica Complejidad se muestra en la tabla a continuación con su gráfico respectivo.

Tabla O.30. Característica Complejidad para cada Caso de Estudio. Fuente: Elaboración Propia.

Definición operacional	ATAM	MEL	ABDM	BOSCH	AMEVA
18. Presencia de la encapsulación de conceptos.	5	3	4	5	5
19. Presencia de modularidad.	3	4	4	4	5
20. Número de elementos en el modelo adecuado a los conceptos establecidos.	5	4	4	5	5
21. Existencia de elementos que no contribuyen al entendimiento del modelo conceptual.	0	0	0	0	0
22. Existencia de relaciones que no contribuyen al entendimiento del modelo conceptual.	0	0	0	0	0

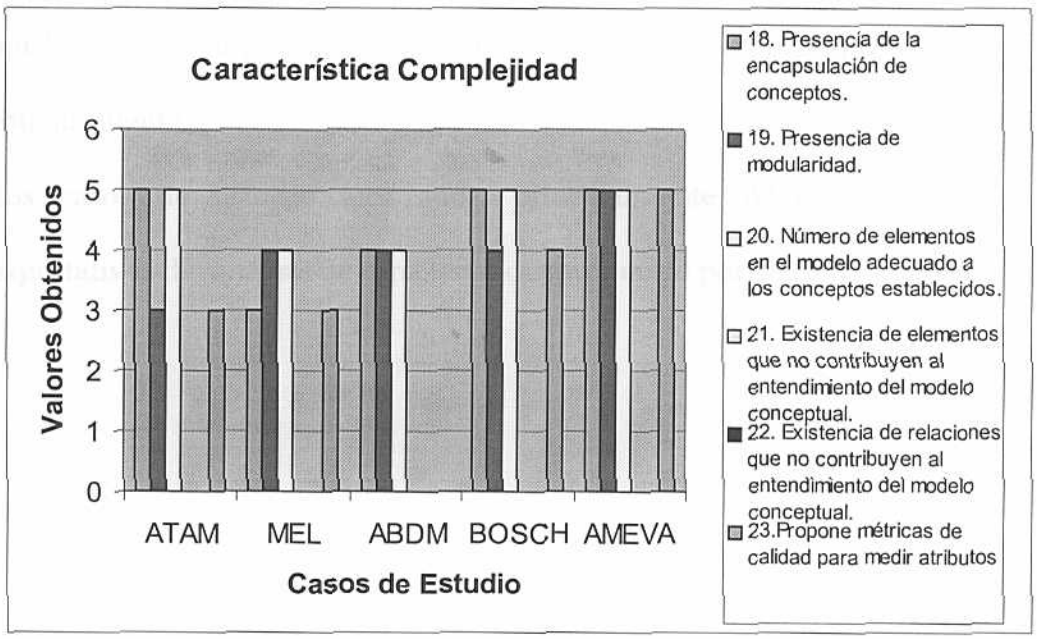

Figura O.5. Característica Complejidad para cada Caso de Estudio. Fuente: Elaboración Propia.

Finalmente, la característica Complejidad mostró los valores menos

homogéneos durante la evaluación. Esto se debe a que en los modelos conceptuales estudiados no es tan obvio la presencia de la métrica "Presencia de Modularidad". En los modelos existen un número de conceptos con características comunes pero no fueron representados de forma modular, lo cual tampoco era necesario, pero sí podía dar una mayor simplicidad en la comprensión de los mismos. El modelo conceptual del caso de estudio de AMEVA, si muestra cierta modularidad que permite una identificación rápida de los componentes del ambiente. Las métricas "Existencia de elementos que no contribuyen al entendimiento del modelo conceptual" y "Existencia de relaciones que no contribuyen al entendimiento del modelo conceptual" obtuvieron un resultado equivalente a "no aplica", dado que estos modelos conceptuales fueron refinados en etapas anteriores de esta investigación con la finalidad de presentarlos de la forma más clara y evidente para su comprensión, por lo tanto es utópica la aparición de relaciones o elementos que dificulten su entendimiento.

Los casos de estudio mostrados, principalmente AMEVA, llenaron las expectativas del análisis de características propuesto por DESMET.

APÉNDICE P - Especificaciones del Diagrama de Clases

En algunas las clases se definieron vectores que son la representación de listas para almacenar propiedades, características, etc., referente a cada clase. A continuación se da una breve descripción de las mismas.

Lista de vector atributo: es un vector compuesto por una dupla (atributo, bool) donde el atributo tendrá valor "true" cuando sea especificado por el arquitecto como requerimiento de calidad para evaluar. Es decir, se especifican los requerimientos de calidad para evaluar la arquitectura.

Lista métricas: es un vector compuesto por cinco campos (nombre atributo, subcaraterística, nombre métrica, valor inferior, valor superior) en el caso de tener métricas.

Lista modelo: es un vector compuesto por duplas (id, nombre modelo) que se actualiza cuando un nuevo modelo es ingresado.

Lista método: es un vector compuesto por duplas (id, nombre método) que se actualiza cuando un nuevo método es ingresado.

Lista técnica: es un vector compuesto por triplas (id, nombre técnica, especificación).

Lista de Escenarios: esta lista, le entran parámetros. Por esto, es una clase parametrizada que usa como entrada el vector atributo definido en el programa principal. Si todos los atributos poseen valor falso, devuelve un error y los escenarios no se crean.

Otras consideraciones fueron tomadas en cuenta para la elaboración del diagrama de clases.

Dado las bondades que posee el lenguaje de programación JAVA, no se considera el modelamiento de clases con respecto a la portabilidad del sistema, ni a la conexión con la base de datos, ya que JAVA ofrece librerías y funciones predeterminadas.

La clase Escenario es una clase parametrizada. Una clase parametrizada se denota con un subcuadro en el extremo superior de la clase, en donde se especifican los parámetros que deben ser pasados a la clase para que esta pueda ser instanciada.

La clase Herramienta es una interfaz o clase abstracta y se denota con el nombre de la clase y de los métodos con letra "itálica". Esto indica que la clase definida no puede ser instanciada pues posee métodos abstractos (aún no han sido definidos, es decir, sin implementación). La única forma de utilizarla es definiendo subclases, que implementan los métodos abstractos definidos.

Se toma la estructura de JAVA beans para las clases, donde se tiene métodos set (lee el valor de la propiedad) y get (cambia el valor de la propiedad).

El ambiente se descompuso en subsistemas, por lo tanto para incluir una nueva clase, se debe colocar en el subdirectorio donde pertenece. Si la nueva clase es hija se debe colocar en la sentencia "extends padre" , el nombre de la clase padre.

Los modelos y métodos, has sido definidos en clases. Es decir, una sola clase define

todas las propiedades y métodos pertenecientes a un modelo de calidad. La justificación es la siguiente:

- Sistema débilmente acoplado, significa cada clase (modelo/método) es independiente. La clase padre tiene definido los métodos comunes, pero los específicos de cada método/modelo están definidos en su clase correspondiente. Al momento de introducir un nuevo método/modelo, solo de debe agregar la sentencia a la clase "extends padre" para que herede los métodos del padre.
- Estandarización de clases. Si se descompone cada método/modelo en clases, a la hora de introducir un método/modelo nuevo con sus clases pertinentes, se deberán crear nuevas clases y modificar el código existente en el diagrama para relacionar nuevas clases con las clases existentes anteriormente. Los beneficios que ofrece esta estandarización de clases, es que facilita la implementación de nuevos componentes.
- Basados en el análisis de antecedentes y en los requerimientos especificados en los objetivos, se determinó que:
    - Los modelos de calidad idóneos para incluir en el ambiente son el Modelo de Calidad de Ortega y el Modelo de ISO 9126-1 para arquitectura , por lo tanto, cada modelo será especificado en una clase.
    - Los método d evaluación idóneos para incluir en el ambiente son ATAM, ML y Método de Bosch, por lo tanto, cada método será especificado en una clase.

- El ADL seleccionado como idóneo para incluir en el ambiente fue RAPIDE, por lo tanto, fue especificada una clase genérica ADL con métodos genéricos. A parte, la clase del ADL se definió como una estructura simplificada de las rutinas propias de RAPIDE que son invocadas por la aplicación. Contiene un conjunto de métodos para la gestión del mismo dentro del Ambiente. No es posible entrar en mayor nivel de detalle debido a que este componente de software es externo al sistema y su funcionamiento escapa al control del mismo. En el Anexo JJ - *RAPIDE*, se describe con mayor detalle el funcionamiento del ADL seleccionado.

APÉNDICE Q - Impacto de los Atributos sobre la Arquitectura

Basándose en los perfiles, se muestra el impacto de cada atributo sobre la arquitectura propuesta.

Tabla Q.1. Análisis del Impacto de la Fiabilidad. Fuente: Elaboración Propia.

Escenario	Soporte que ofrece la arquitectura	%
C1: El ambiente requiere acceder a la BD.	Acceso a base de datos. Existe una clase que permite establecer la conexión con el manejador y actualizar los cambios efectuados.	100
C2: El ambiente requiere una estrategia de integración con elementos adicionales.	Existencia de un programa principal que contiene la opción mantenimiento, la cual permite incluir nuevos elementos al ambiente, es decir, los mecanismos ya están estipulados. Estos elementos están restringidos a los ya existentes en el ambiente.	100
C3: El ambiente requiere una estrategia de mantenimiento.	En el programa principal esta opción se encuentra disponible. Todas las clases principales, están provistas de métodos capaces de ofrecer mantenimiento a la misma.	95
C4: El ambiente se encuentra disponible 150 horas a mes.	La arquitectura provee condiciones para que el ambiente se encuentre disponible. A pesar de esto, no provee contingencia si hay una caída del ambiente durante su ejecución.	25
C5: El ambiente anula una operación incorrecta.	Dentro de las clases se definen métodos de validación. Si el usuario intenta ejecutar una operación incorrecta, el ambiente está en capacidad de notificarle al usuario anular la operación.	100
C6: El ambiente necesita que su tiempo de recuperación sea menor a 10 segundos.	No existe un mecanismo dentro de la arquitectura que soporte la recuperación del ambiente.	0
C7: El ambiente necesita recuperar datos en caso de falla.	Se encuentra definido un método en la clase del Manejador que se encarga de respaldar los datos. Se debe especificar que sólo se creará el back up de la BD.	100

Tomando como base el análisis del impacto para la fiabilidad se observó que la arquitectura propuesta cubre con las expectativas de los escenarios C1, C2, C3, C5 y C7, debido a que existen estructuras o mecanismos que soportan dichos escenarios. El escenario C4, no satisface del todo ya que no se contemplan mecanismos de contingencia y recuperación del ambiente. Aunque la probabilidad

de que falle es poca, se deben tomar en cuenta procesos de recuperación. Por el mismo motivo, el escenario C6 no cubre ninguna expectativa con respecto a este perfil. También se debe resaltar que el escenario C4 es el de mayor frecuencia dentro del ambiente especificado por este perfil.

Tomando en cuenta los pesos relativos definidos en los perfiles de cada uno de los escenarios y el porcentaje de soporte de la arquitectura, el valor de la Fiabilidad en el ambiente es calculado:

(0.17*(100/100)) + (0.121*(100/100)) + (0.195*(95/100)) + (0.21*(25/100)) + (0.17*(100/100)) + (0.0731*(0/100)) + (0.0487*(100/100)) = 0.7474

Por lo que se puede decir que el porcentaje de la Fiabilidad en AMEVA es aproximadamente 75%.

Este es el análisis de impacto para la Mantenibilidad:

Tabla Q.2. Análisis del Impacto de la Mantenibilidad. Fuente: Elaboración Propia.

Escenario	Soporte que ofrece la arquitectura	%
C8: El ambiente requiere independencia funcional de los módulos.	Existe un programa principal que hace las llamadas a estos módulos. Sin embargo, el funcionamiento depende únicamente de ellos mismos.	100
C9: El ambiente requiere la posibilidad de incorporar nuevos Modelos de Especificación de la Calidad, modificarlos o eliminarlos.	El subsistema de Modelos de Calidad posee una clase que tiene métodos que permiten efectuar estas operaciones. Después de efectuar dichas operaciones se actualiza la BD.	100
C10: El ambiente requiere la posibilidad de incorporar nuevos Métodos de Evaluación, modificarlos o eliminarlos.	El subsistema de Métodos de Evaluación posee la clase Métodos de Evaluación que tiene métodos que permiten efectuar estas operaciones. Después de efectuar dichas operaciones se actualiza la BD.	100
C11: El ambiente requiere la posibilidad de incorporar nuevas Técnicas de Evaluación, modificarlos o eliminarlos.	El subsistema de Escenarios posee una clase Técnicas de Evaluación que tiene métodos que permiten efectuar estas operaciones. Después de efectuar dichas operaciones se actualiza la BD.	100

C12: El ambiente requiere la posibilidad de incorporar nuevos ADLs o eliminarlos.	El subsistema de Escenarios posee una clase ADL que tiene métodos que permiten efectuar estas operaciones. Después de efectuar dichas operaciones se actualiza la BD.	100
C13: El ambiente requiere una medida de interconexión entre los módulos o un nivel de acoplamiento.	Se toma módulos débilmente acoplados, dado a que en cada módulo o subsistema, se define una clase padre, y al incorporar un nuevo elemento se hace como una clase hija.	100
C14: El ambiente requiere que sus elementos se encuentren relacionados para llevar a cabo las funcionalidades propias.	El nivel de cohesión se mantiene ya que las clases padres están relacionadas entre sí, y al introducir un nuevo elemento (siempre relacionado a una clase padre), las relaciones se heredan.	100
C15: El ambiente requiere que los nombres de los módulos y los identificadores tengan significado dentro del contexto de la aplicación.	La aplicación esta dividida en subsistemas que pueden ser considerados como módulos. Cada módulo se encuentra identificado con las clases definidas en su interior. Esto no es obstáculo para relacionar las clases entre si.	100

Como se pudo observar, la arquitectura propuesta soporta los escenarios definidos para este perfil. Los mecanismos de mantenibilidad fueron tomados en cuenta a la hora del diseño. Las expectativas de mantenibilidad fueron cubiertas a cabalidad. Para determinar el porcentaje del perfil Mantenibilidad se toman los pesos relativos definidos en los perfiles y el porcentaje por escenario.

(0.155*(100/100)) + (0.103*(100/100)) + (0.103*(100/100)) + (0.103*(100/100)) + (0.103*(100/100)) + (0.155*(100/100)) + (0.155*(100/100)) + (0.120*(100/100))= 0.997

Por lo tanto, se consideró que la Mantenibilidad se encuentra 100% presente en la arquitectura propuesta para AMEVA.

Este es el análisis para la Interoperatividad:

Tabla Q.3. Análisis del Impacto de la Interoperatividad. Fuente: Elaboración Propia.

Escenario	Soporte que ofrece la arquitectura	%
C16: El ambiente requiere compatibilidad de un elemento externo para aplicar la simulación (ADL).	La presencia de la clase ADL indica que se toma en cuenta este elemento. Sin embargo, la compatibilidad no se puede asegurar hasta el momento de la implementación. De acuerdo al ADL y al lenguaje de programación seleccionados, se estima que no habrá conflicto.	75

C17: El ambiente requiere ejecutar operaciones de mantenimiento al ADL.	Existe una clase definida que se encarga de acoplar o desacoplar el ADL dentro del ambiente. Es importante acotar, que estas operaciones de mantenimiento, solamente pueden ser de agregar o eliminar un ADL, ya que el funcionamiento del mismo, escapa de las manos del implementador y no puede ejecutar operaciones de modificación sobre el mismo.	100

Los mecanismos de interoperatividad se ven soportados por la arquitectura. A pesar de ello, no se encuentra claro la forma de implementación, y por eso el escenario C16 no satisface por completo el perfil. El valor obtenido según los pesos y el porcentaje definidos en los perfiles fue:

$(0.5*(75/100)) + (0.5*(100/100)) = 0.875$

La Interoperatividad de AMEVA obtuvo un valor aproximado de 88%, lo que determina que este perfil se encuentra presente en el ambiente especificado, es decir, fue considerado el hecho de incluir un elemento externo al ambiente mientras se diseñaba el mismo.

ANEXO A - Modelos de Especificación de la Calidad

Modelo ISO 9126-1

Este modelo sugiere que la calidad está descompuesta es seis factores: Funcionalidad, Fiabilidad, Eficiencia, Usabilidad, Mantenibilidad y Portabilidad, y que a su vez, cada componente de la calidad del software debe ser descrito en términos de uno o varios de estos seis factores, los cuales están soportados por un conjunto de atributos que pueden ser refinados en muchos niveles de subcaracterísticas (ISO/IEC 9126-1, 1998).

ISO 9126-1 (1998) define estos seis factores de la siguiente manera:

- **Funcionalidad:** es el conjunto de atributos relacionados con la existencia de funciones y sus propiedades, las cuales satisfacen las necesidades expresadas o implicadas en el desarrollo del software
- **Fiabilidad:** Es el conjunto de atributos relacionados con la capacidad del software para mantener su nivel de rendimiento bajo condiciones establecidas por un período de tiempo establecido.
- **Usabilidad:** Es el conjunto de atributos relacionados con el esfuerzo necesitado para usar el software y con la evaluación individual de tal uso por un número de usuarios establecidos o implicados.
- **Eficiencia:** Es el conjunto de atributos asociados con las relaciones entre el nivel de rendimiento del software y la cantidad de recursos utilizados bajo

condiciones establecidas.

- **Mantenibilidad:** Es el conjunto de atributos relacionados con el esfuerzo necesitado para realizar modificaciones específicas.
- **Portabilidad:** Es el conjunto de atributos relacionados con la habilidad del software para ser transferido de un ambiente a otro.

ISO 9126-1 (1998) define la calidad como un conjunto de características del producto. Las características externas de calidad son aquellas que gobiernan la forma en que el producto trabaja en su ambiente. Por ejemplo: Usabilidad, Fiabilidad. Las características internas de calidad son las relacionadas a cómo el producto fue desarrollado. Este modelo es uno de los más completos en cuanto a identificación de características, internas y externas, de calidad del producto.

Tabla A.1. Características y Subcaracterísticas de Calidad. Fuente: (ISO/IEC 9126-1, 1998).

CARACTERÍSTICAS	SUBCARACTERISTICAS
Funcionalidad	Adecuación. Precisión. Interoperabilidad, Conformidad Seguridad.
Fiabilidad	Madurez. Tolerancia a fallas. Recuperabilidad.
Eficiencia	Comportamiento del tiempo. Comportamiento de los recursos.
Usabilidad	Entendimiento. Capacidad de aprendizaje, Operabilidad.
Mantenibilidad	Capacidad para ser analizado. Capacidad para ser modificado. Estabilidad. Capacidad para ser validado.
Portabilidad	Adaptabilidad. Facilidad de instalación. Conformidad. Capacidad para remplazar.

La primera etapa de este modelo viene con la definición de los requisitos de

calidad , que toma como entrada un sistema de necesidades indicadas o implicadas, documentación técnica relevante y el estándar de ISO sí mismo y produce una especificación de requisito de calidad (ISO/IEC 9126-1, 1998).

La segunda etapa es la preparación de la evaluación, que implica la selección de las métricas apropiadas, de la definición del nivel y de la definición de los criterios. La calidad se refiere a las necesidades dadas, que varían a partir de una evaluación a otra. La definición de los criterios implica el elaborar de un procedimiento (métricas) para resumir los resultados de la evaluación y toma, como entrada, criterios de la gerencia específicos a un ambiente particular que pueda influenciar la importancia relativa diversas características de la calidad y las características secundarias. Esta definición, es también por lo tanto específico a la evaluación particular (ISO/IEC 9126-1, 1998).

La etapa final es el procedimiento de la evaluación, que se refina en tres pasos: medida, grado y cierre. Los primeros dos son intuitivo directos: en la medida a, las métricas (las cuales se pueden instanciar de otro modelo) seleccionadas se aplican el producto de software y los valores en las escalas de la métrica obtenida. Posteriormente, para cada valor medido, se determina el nivel de grado. El cierre es el paso final del proceso de la evaluación del software, donde se resume el sistema de niveles clasificados. El resultado es un resumen de la calidad del software visto como producto. La calidad en resumen, es comparada con otros aspectos tales como tiempo y costo, y la decisión final es tomada, basada en

criterios gerenciales(ISO/IEC TR 15504-2, 1998).

Modelo de Calidad de Producto con Enfoque Sistémico

El modelo planteado por Ortega (2000) se basa en el Modelo de Calidad Sistémica de Callaos y Callaos. Para poder desarrollar este modelo, se estudiaron los diferentes modelos de producto y se identificaron las características de alto nivel de calidad del producto. Luego se seleccionaron los elementos que permitieron obtener un modelo de calidad del producto con un enfoque sistémico. Los componentes que son tomados en cuenta en el modelo de calidad del producto son los siguientes:

La eficiencia y efectividad del producto como calidad parcial del modelo de Calidad Sistémica de Callaos (1996).

Las características de calidad del modelo de Dromey (1996) y el estándar internacional ISO/IEC 9126-1 (ISO/IEC 9126-1, 1998): Eficiencia, Fiabilidad, Funcionalidad, Mantenibilidad, Portabilidad y Usabilidad.

La relación utilizada en el modelo de McCall (Gillies, 1977) entre los atributos y calidad de las métricas.

Este modelo es mostrado en la Figura xxx y plantea, sobre la base de las 6 características de calidad del estándar internacional ISO/IEC 9126-1 (1998), un conjunto de sub-características y métricas asociadas que miden la calidad de un producto de software con un enfoque sistémico. Estas métricas (999 en total) propuestas por Ortega (2000), son el resultado del análisis de las cuatro dimensiones de la Matriz de Calidad Sistémica de Callaos que están asociadas a

cada característica.

Aunque el modelo que propone Ortega (2000) soporta el enfoque sistémico propuesto por Callaos y Callaos, el modelo está orientado únicamente a la estimación de la calidad del producto de software dejando de un lado la calidad del proceso. A pesar de ello, el modelo menciona algunas sub-características de cada métricas asociadas al proceso de desarrollo.

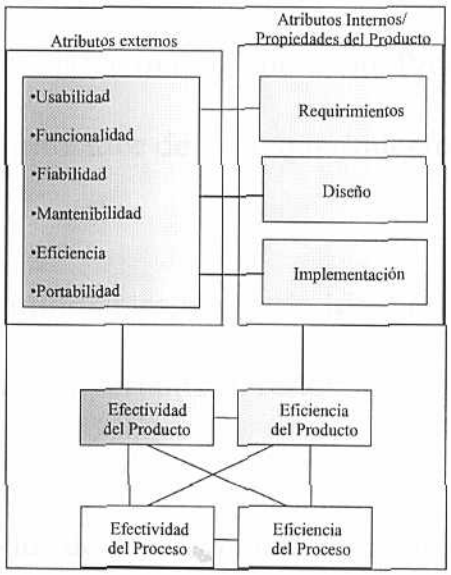

Figura A.1. Modelo de Calidad del Producto de Software
Fuente: (Ortega et al.,2000).

Las 6 características del modelo, aunados a cada una de las sub-características, hacen del modelo un instrumento de medición de gran valor ya que cubre con todas los aspectos necesarios e imprescindibles para medir directamente la calidad del producto de software; esto se debe a su similitud con el estándar internacional ISO/IEC 9126.

Cabe acotar que, según en método propuesto por Ortega y sus colaboradores para

aplicar el modelo, no todas las métricas se usan en todos los casos, sino que el modelo debe ser adaptado dependiendo de las características del producto deseado.

Árbol de Utilidad (Utility Tree)

El árbol de utilidad es un modelo de calidad basado en escenarios, que provee un mecanismo de top-down donde se traducen los distintos controladores de un sistema en escenarios de atributos de calidad específicos (Kazman, 1999).

Un árbol de utilidad es un AHP (Analitic Hierarchy Process) Proceso de Jerarquía Analítica como modelo controlador de los requerimientos de atributos específicos (Kazman, 1999).

Para generar un árbol de utilidad, es necesario identificar, dar prioridad y refinar los atributos de calidad más importantes o resaltantes del sistema. El nodo raíz es la Utilidad. Generalmente el desempeño, la modificabilidad, seguridad y disponibilidad son los nodos de alto nivel (atributos predominantes) e inmediatos a la utilidad, aunque los diferentes grupos de "stakeholders" pueden agregar ellos mismos más atributos de calidad o los mismos bajo otros nombres. Los escenarios son las hojas de árbol de utilidad.

Es muy importante entender la importancia relativa al alcance del sistema, con respecto a los atributos de calidad, tal como el desempeño, para determinar cuando enfocar la atención durante la evaluación de la arquitectura. Los árboles de utilidad ayudan a concretizar y priorizar rasgos relacionados a la calidad (Kazman, 1999).

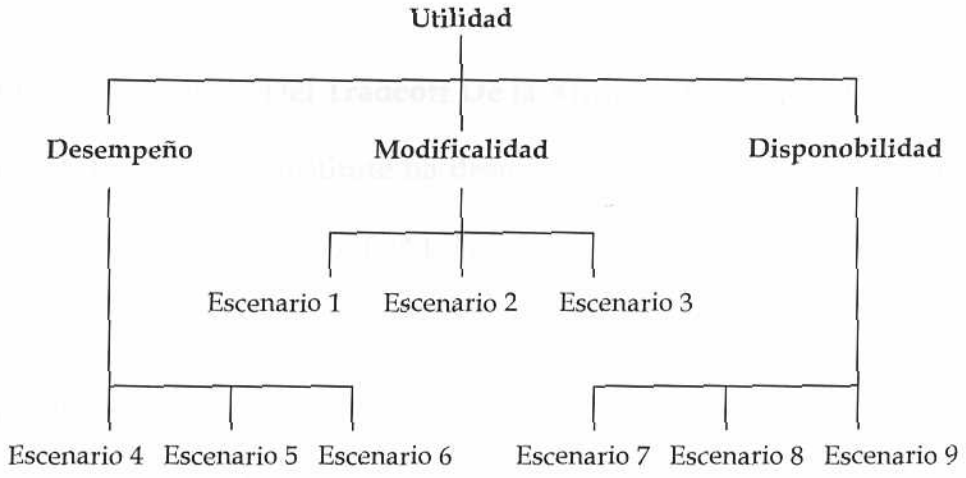

Figura A.2. Ejemplo de Árbol de Utilidad sin prioridades.
Fuente: (Kazman, 1999)

El resultado de la aplicación de este modelo es un resumen con las prioridades de los requerimientos del sistema basado en atributos de calidad específicos (Kazman, 1999). El sistema de los escenarios generados y del subconjunto que traza sobre la arquitectura. Un sistema de atributos de calidad específicos. Las preguntas específicas que fueron aplicadas a la arquitectura y a las respuestas a estas preguntas. Un sistema de riesgos identificados y un sistema de no-riesgos identificados (Kazman, 1999).

Al igual que los Perfiles, los Árboles de Utilidad son una herramienta usada para identificar escenarios.

ANEXO B - Métodos de Evaluación

El Método Del Análisis Del Tradeoff De la Arquitectura (ATAM)

El Software Engineering Institute ha desarrollado un Método para el Análisis del Tradeoff de la Arquitectura (ATAM) cuyo propósito es evaluar las consecuencias de las alternativas de decisión arquitectural a la luz de los requerimientos de los atributos de calidad. Este método permite: descubrir riesgos derivados de las decisiones arquitecturales, descubrir puntos de sensibilidad y descubrir tradeoffs (Kazman, 1999). El ATAM se basa sobre un sistema de medidas de atributos específicos del sistema, basados sobre modelos formales y algún cualitativo, basado sobre las inspecciones formales.

Los atributos de calidad de los grandes sistemas de programas son determinados principalmente por la arquitectura del software. ATAM es un método para evaluar la arquitectura del software considerando múltiples atributos de calidad (Kazman, 1999).

ATAM es un modelo en espiral dividido en cuatro fases principales donde cada una de ellas realiza uno o más aportes al entendimiento del sistema, reduciendo riesgos y modificando el diseño (Klein et al., 1999). En la Fase I este instrumente evalúa la recolección de escenarios y la recolección de requerimientos, restricciones y ambiente de la arquitectura. En la Fase II, se hace una descripción de las diferentes vistas arquitecturales y se construyen los escenarios. En la Fase III, en base al análisis de los atributos específicos de calidad, se hace el diseño del modelo.

Finalmente en la Fase IV, se identifican los puntos de sensibilidad y por consiguiente se identifican los Tradeoffs de la arquitectura (Kazman et al, 1999).

En ATAM se evalúa la arquitectura del software considerando múltiples atributos de calidad (Kazman, 1999). Este es el esquema más representativo de los pasos antes mencionados :

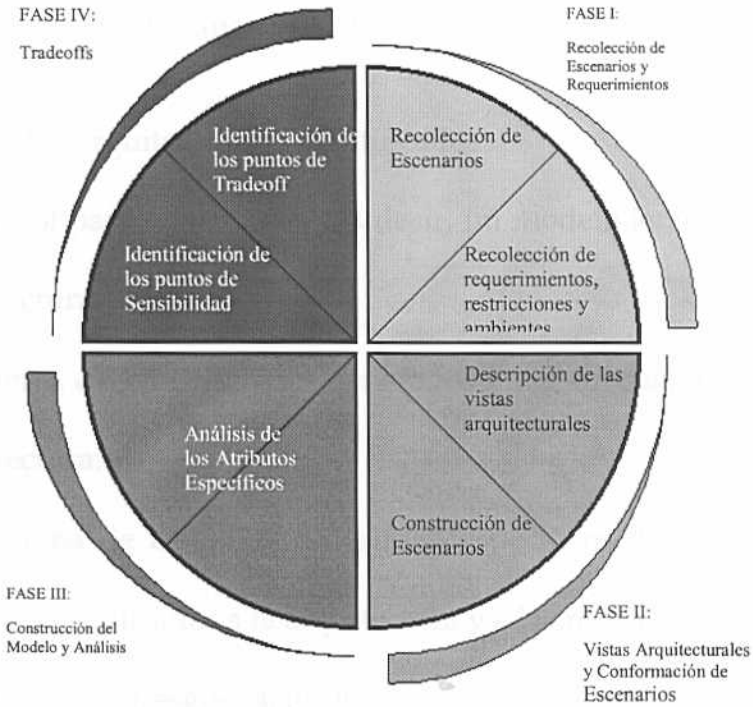

Figura B.1. Pasos del Método ATAM
Fuente: (Kazman, 1999)

Los resultados más importantes de aplicar ATAM, son arquitecturas mejoradas. La ayuda de ATAM en obtener sistemas de atributos de calidad a lo largo de múltiples dimensiones, analizar los efectos de cada atributo en el aislamiento, y después entender las interacciones de estos atributos (Klein et al., 1999). Ambos procesos implican aspectos técnicos y sociales. Los aspectos técnicos tratan de cómo se acopia, se representa, y se analiza la información arquitectural. Los

aspectos sociales se ocupan de las interacciones entre los usuarios de áreas específicas del sistema, permitiendo que se comuniquen en un marco común, para hacer las asunciones implícitas en sus análisis explícitos, y para proporcionar una base objetiva para manejar los tradeoffs inevitables de la arquitectura. La salida de un ATAM es un informe escrito que incluye los principales resultados de la evaluación. Éstos son (Kazman et al, 1999):

- Los estilos arquitecturales identificados .
- Un "árbol para uso general" es decir, un modelo jerárquico de los atributos arquitecturales.
- El sistema de los escenarios generados y del subconjunto que traza sobre la arquitectura.
- Un sistema de atributos de calidad específicos. Las preguntas específicas que fueron aplicadas a la arquitectura y a las respuestas a estas preguntas.
- Un sistema de riesgos identificados.
- Un sistema de no-riesgos identificados.

El Método de Diseño Basado en la Arquitectura (ABDM)

Este método de diseño basado en la arquitectura (ABDM) es usado para diseñar una arquitectura de alto nivel del software para una línea de productos o un sistema duradero. Diseñar una arquitectura con tales expectativas es difícil porque los requisitos detallados no se saben por adelantado. El método de ABD satisface la funcionalidad, la calidad, y los requisitos del negocio en un nivel de la abstracción

que permite la variación necesaria al producir productos específicos (Bachmann et al., 2000).

El método ABD tiene cuatro fases, las cuales están determinadas por un conjunto de actividades, principalmente la primera fase, donde se define la Vista Lógica. En la segunda fase, se define la Vista de procesos, seguido de la definición de la Vista de Desarrollo en la tercera fase, y en la última fase la verificación con escenarios de calidad y restricciones. En la siguiente figura se muestra el esquema de este modelo.

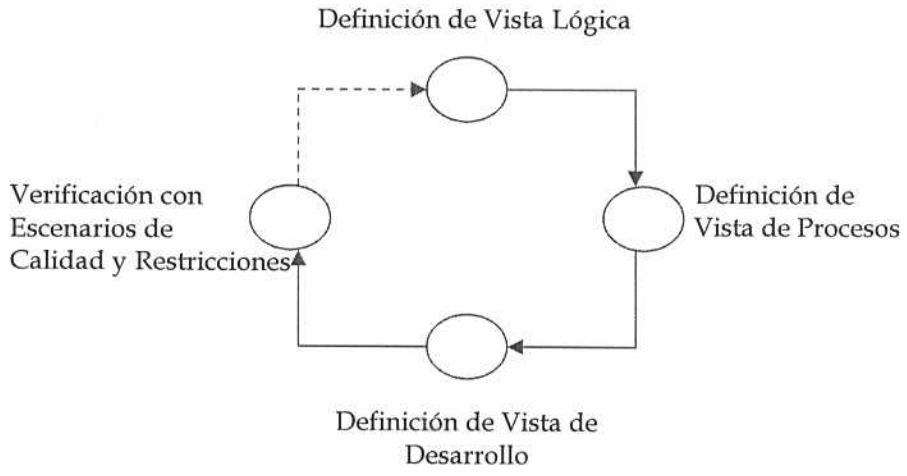

Figura B.2. Pasos del Método ABD.
Fuente: (Bachmann et al., 2000).

En resumen, todas las tareas que se deben llevar a cabo para la aplicación de este método son las siguientes:

- Identificación de conectores arquitectónicos.
- Ecapsulamiento de la funcionalidad.
- Selección de estilos arquitectónicos.
- Adecuación de la funcionalidad a los estilos arquitectónicos.

- Refinación de los templates.
- Verificación de la funcionalidad.
- Generación de la vista de procesos.
- Generación de la vista desarrollo.
- Verificación de escenarios de calidad.

El método proporciona una serie de pasos para diseñar la arquitectura conceptual del software. La arquitectura conceptual del software proporciona la organización de las funciones y los requerimientos funcionales (Bachmann et al., 2000). En la siguiente figura se muestra las actividades más importantes del ABDM, las cuales se encuentran en la primera fase.

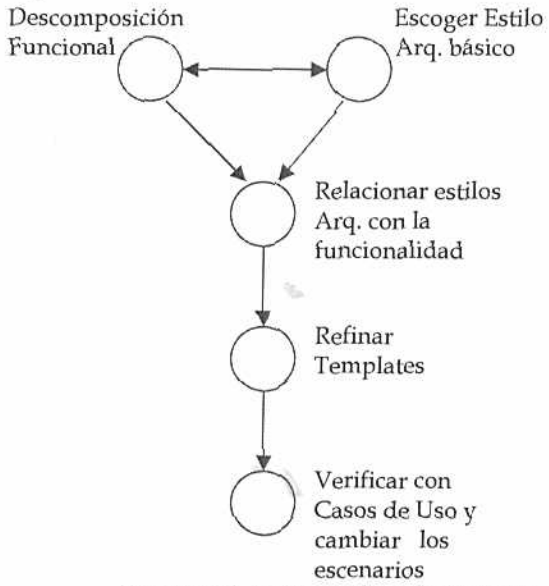

Figura B.3. Primera fase de ABDM.
Fuente: (Bachmann et al., 2000).

Para relacionar los elementos funcionales con un estilo arquitectónico se deben estudiar estos requerimientos. Los elementos tienen un conjunto de responsabilidades que se deben separar en grupos. El objetivo de separar estos

grupos, es para que cada uno represente un elemento diferente dentro de la arquitectura (Bachmann et al., 2000). El criterio de división dependerá de atributos de calidad son importantes en un elemento en particular. Existen varios criterios como el de coherencia funcional, niveles de abstracción semejantes, situación de responsabilidad y patrones de datos o comportamiento similares. Estos criterios son fijados por el arquitecto de acuerdo a sus necesidades y las necesidades del sistema (Bachmann et al., 2000).

Una de las salidas más importantes del método es una colección de plantillas o templates del software que obligan la puesta en práctica de los componentes de diversos tipos. Las plantillas del software incluyen una descripción de cómo los componentes obran recíprocamente con servicios compartidos y también incluyen las responsabilidades de los componentes (Bachmann et al., 2000).

Método de Evaluación Arquitectónica de Losavio (M.E.L)

M.E.L. (Losavio et. al, 2002), es un método de evaluación de la calidad de arquitecturas de software. Para llevar a cabo esta evaluación es necesario realizar un conjunto de tareas o actividades que dan como resultado de cada una, una serie de elementos que permiten especificar la arquitectura y evaluarla.

A continuación se presentan los pasos del método de evaluación M.E.L.

Tabla B.1. Flujos de Trabajo y Artefactos de M.E.L. Fuente: (Losavio et al, 2002)

Requerimientos de Flujo de Trabajo	Artefactos
A.- Captura de los requerimientos funcionales a través de los Casos de Uso.	Modelo de Casos de Uso: • Diagrama de Casos de Uso. • Especificación de Casos de Uso.
B.- Captura de Requerimientos No Funcionales: B.1.- Especificación de los requerimientos de calidad. B.2.- Especificación de restricciones tecnológicas. B.3.- Especificación de los requerimientos de calidad para cada caso de uso.	• Lista de características de calidad basadas en el Modelo de Calidad ISO 9126-1. • Lista de características. • Especificación de los casos de uso con una lista de características de calidad.
C.- Prioridad de los Casos de Uso.	• Lista de los Casos de Uso más relevantes para la arquitectura.
D.- Detalle de los Casos de Uso.	• Especificación de los casos de uso incluyendo la lista de características de calidad, subcaracterísticas, medidas y prioridad.
E.- Identificar Arquitecturas Candidatas.	Lista de Arquitecturas Candidatas: • Diagrama de componentes y conectores o especificación de la arquitectura.
F.- Aplicación de la estructura para la evaluación de las arquitecturas candidatas.	• Tabla de Evaluación para las arquitecturas candidatas.
G.- Análisis de la tabla de Evaluación para seleccionar la arquitectura candidata.	Arquitectura Candidata: • Análisis del reporte técnico. • Especificación de la arquitectura.

Estas actividades o pasos, se pueden llevar en el mismo orden de aparición. Sin embargo, la realización de algunas tareas en paralelo está permitido.

Método para el diseño de Arquitecturas de Software BOSCH

Este método propone tres fases para la evaluación de diseño de arquitecturas. De cada fase se obtienen una serie de artefactos que ayudan a modelar la arquitectura candidata. En la figura xxx que se presenta a continuación se muestra un esquema de este método.

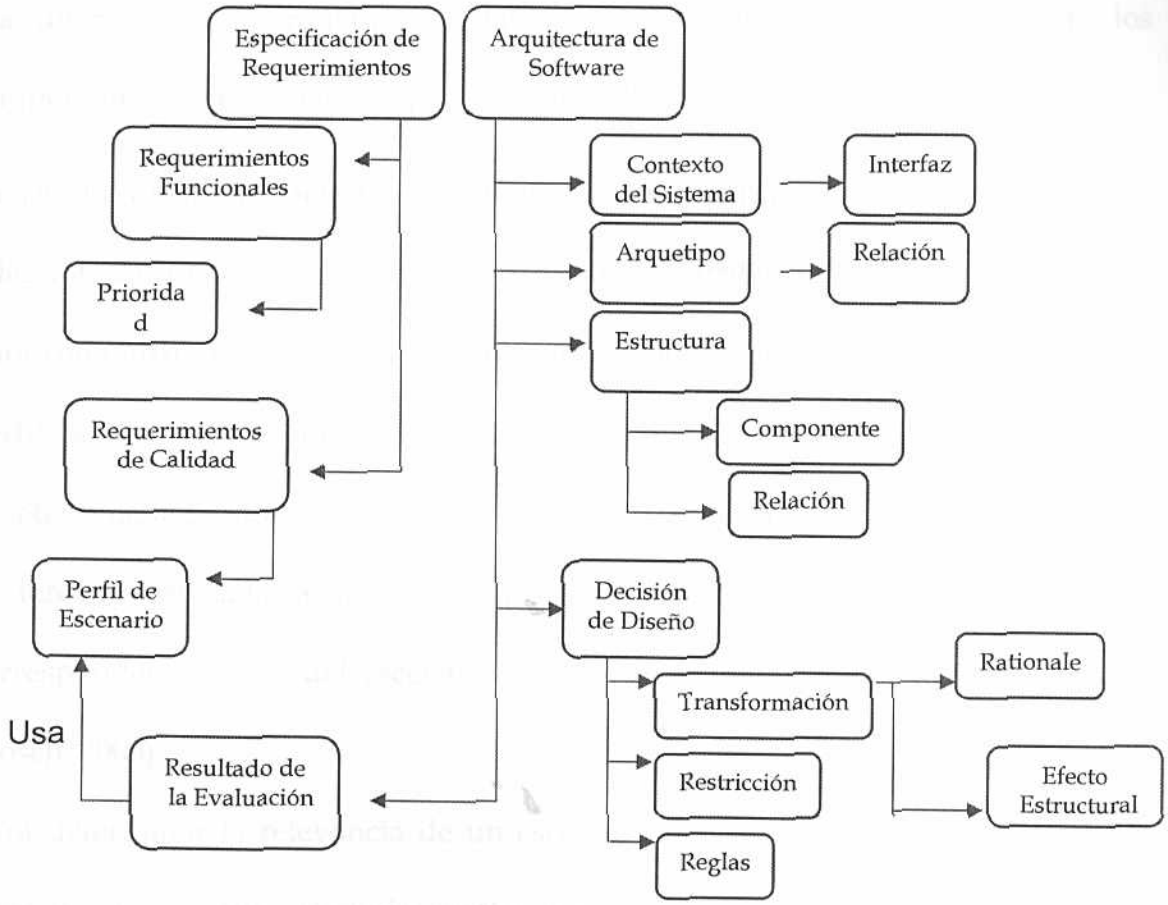

Figura B.4. Método de Diseño de Arquitecturas de Software Orientado a Atributos de Calidad
Fuente: (Bosch, 2002)

En la primera fase, diseño arquitectónico basado en funcionalidad se busca diseñar una primera versión de la arquitectura basada en sus requerimientos funcionales.

De acuerdo a Bosch (2000), la primera fase del método de diseño de Arquitectura de Software comprende la determinación del contexto del sistema con cada interfaz, descomponer el sistema en sus componentes principales e identificar las relaciones entre ellos e identificar los arquetipos.

La culminación de esta fase, es la base para las dos fases posteriores; al terminarla se debe contar con los siguientes artefactos: el contexto de sistema, en términos de sus interfaces; los arquetipos y las relaciones entre ellos; y finalmente los componentes y sus relaciones que conforman la estructura.

La segunda fase del método consiste en la evaluación de la arquitectura y en la selección de un perfil, el cual, tiene asociado un atributo de calidad. Los perfiles para cada atributo se especifican en lo que se denomina tabla de presentación de perfil, la misma esta conformada por cinco columnas, la primera de ellas referida a la categoría; la segunda, los posibles puntos a evaluar por cada categoría (métrica); la tercera, asociada a los escenarios; finalmente las dos últimas columnas corresponden al peso del escenario y al peso relativo de este dentro del perfil (Bosch, 2000).

Para determinar la relevancia de un escenario dentro de una categoría se deben asignar pesos que posteriormente son normalizados y para realizar la normalización, a cada escenario, se le asigna un valor que se da de acuerdo al soporte que proporcionan para el sistema en cuestión y luego se suman las puntuaciones de todos los escenarios para obtener un total. Haciendo uso de este total se obtiene el peso relativo de cada escenario, dividiendo el peso asignado a

éste entre el total. Este resultado es un número entre 0 y 1 que indicará la importancia relativa del escenario dentro del perfil. La suma de todos los pesos relativos debe dar como resultado 1 (Bosch, 2000).

Para realizar la evaluación, en la tablas de presentación de los perfiles, es necesario incluir a cada escenario una métrica de calidad que sustentase la medición de la relevancia de éstos dentro del perfil, con la idea de contar con un mayor nivel de detalle al realizar la evaluación en base a la técnica seleccionada y posteriormente al elaborar los resultados.

Bosch (2000) propone dos alternativas al momento de elegir los perfiles. Estos pueden ser: seleccionados (que no contienen todos los escenarios) o completos de acuerdo a los escenarios que componen el perfil.

La evaluación puede ser; cualitativa, que es cuando se realiza la comparación de arquitecturas; cuantitativa, expresa en número los atributos de calidad de la arquitectura evaluada; y máximo teórico, determina el máximo teórico de una arquitectura para un atributo en particular.

Se debe seleccionarse la técnica de evaluación a utilizar, Bosch (2000) propone la elección de una de las siguientes técnicas: evaluación basada en escenarios, evaluación basada en simulación, evaluación basada en modelos matemáticos, evaluación basada en experiencias.

En esta fase los artefactos obtenidos son los que tienen que ver con el resultado de la evaluación para cada atributo y luego son seleccionadas las transformaciones apropiadas.

Durante la tercera fase, la noción de transformación en una arquitectura mejora uno o más de sus atributos de calidad, en este caso se identifican cuatro categorías y cuatro tipos de transformaciones de arquitecturas que pueden ser usadas para elegir las propiedades del sistema (Bosch, 2000).

Las cuatro categorías de transformaciones arquitectónicas son (Bosch, 2000): imposición de un estilo arquitectónico, en éste se pueden unir los estilos arquitectónicos, además éstos tienden a ser predominantes en la arquitectura y la unión debe ser hecha con cuidado; imposición de un patrón arquitectónico, un patrón arquitectónico difiere en un estilo arquitectónico en que no es predominante, pero pueden ser unidos con la mayoría de los estilos y otros patrones arquitectónicos; aplicación de un patrón de diseño, es el uso de un patrón de diseño, los patrones de diseño no tienen un impacto amplio sobre la arquitectura, pero tiene efectos más específicos (localizados); conversión de los atributos de calidad a funcionalidad, esta categoría extiende la Arquitectura de Software con nuevas funcionalidades, restricciones y reglas.

Los pasos a realizar para el proceso de transformación son (Bosch, 2000):

1. Identificar cuales requerimientos de calidad no se cumplieron por la versión actual de la Arquitectura de Software.
2. Identificar para el atributo de calidad actual, en cuál componente o lugar dentro de la arquitectura, el atributo de calidad está inhibido.
3. Seleccionar una transformación que resolverá el problema identificado en la arquitectura.

4. Ejecutar la transformación, significa reorganizar la funcionalidad de acuerdo al estilo o patrón seleccionado, y qué descripción de arquitectura es actualizada para incorporar los cambios.

Estos cuatro pasos se realizan de forma iterada para cada atributo de calidad, finalizando en el paso cuatro y retornando en el paso uno.

Los artefactos desarrollados son las decisiones de diseño y en particular las transformaciones. Otros de los artefactos que se obtienen en esta fase son las subsecuentes versiones de la Arquitectura de Software.

ANEXO C – Metodología de Rational Unified Process

La metodología Rational Unified Process (RUP), es un proceso de Ingeniería de Software que provee un enfoque disciplinado para asignar tareas y responsabilidades dentro de cualquier desarrollo de una organización. Su objetivo es garantizar la producción de un software de alta calidad que satisfaga los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos. El proceso cubre el ciclo de vida de desarrollo de software y está estructurado de tal manera que puede ser adaptado y extendido para cubrir las necesidades de la organización que lo adopte.

RUP agrupa buena parte de las mejores prácticas en el modelo de desarrollo de software, en particular incluye las siguientes:

- Desarrollo de software en forma iterativa.
- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modela el software visualmente.
- Verifica la calidad del software.
- Controla los cambios.

El proceso tiene dos estructuras o dimensiones:

- El eje horizontal representa el tiempo y presenta las diversas fases del ciclo de vida del proyecto.
- El eje vertical representa los flujos de trabajo del proceso, los cuales agrupan

las actividades de acuerdo a su naturaleza.

La primera dimensión representa la parte dinámica del proceso y está expresada en términos de ciclos, fases e iteraciones. La segunda dimensión representa la parte estática del proceso y se describe en términos de componentes, actividades, flujos de trabajo, artefactos y actores. A continuación se muestra un gráfico que explica esta situación.

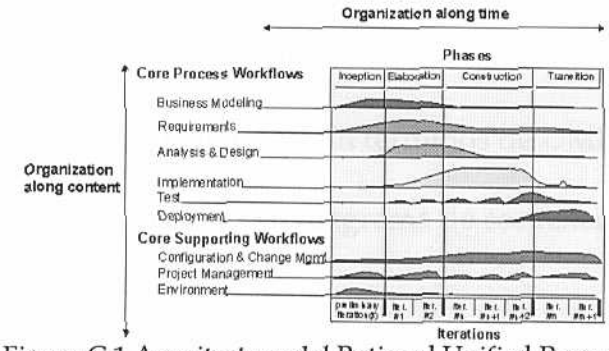

Figura C.1 Arquitectura del Rational Unified Process
Fuente: (Kruchten,2000)

Modelo de Rational Unified Process

RUP se representa usando cuatro elementos principales de modelaje: Trabajadores, Actividades, Artefactos y Flujos de Trabajo.

- **Trabajadores:** Este término se refiere a los roles que definen cómo deberían trabajar los individuos. Un trabajador desempeña uno o más roles y es el propietario de un conjunto de artefactos.
- **Actividades:** Una actividad de un trabajador específico es una unidad de trabajo que un individuo en ese rol puede realizar. La actividad tiene un objetivo claro, usualmente expresado en términos de la creación y modificación de artefactos, tales como un modelo o una clase. Cada

actividad está asignada a un trabajador específico.

- **Artefactos:** Un artefacto es una pieza de información que es producida, modificada o usada por un proceso. Son productos tangibles del proyecto: las cosas que el proyecto produce o usa mientras se trabaja hacia el producto final. Los artefactos se usan como entradas para los trabajadores para realizar una actividad y son el resultado o la salida de esas actividades.
- **Flujos de Trabajo:** Un flujo de trabajo es una secuencia de actividades que producen un resultado notable. En términos de UML, un flujo de trabajo puede ser expresado como un diagrama de secuencia o un diagrama de actividades.

Revisión de las fases de RUP

El proceso iterativo de RUP se organiza en fases, cada fase se concluye con un punto de chequeo o hito (milestones) que brinda información que permite decidir si continuar, abortar o cambiar el curso del desarrollo. A continuación se describen las cuatro fases que conforman el proceso:

Fase de Concepción

Esta fase se concentra principalmente en el correcto entendimiento de los requerimientos y la determinación del alcance del proyecto, como se muestra en la tabla C.1.

Tabla C.1. Descripción de la Fase de Concepción. Fuente: (Rational,2000)

Objetivos
• Establecer el alcance del proyecto y las condiciones límites. • Discriminar los casos de uso del sistema. • Estimar costos y tiempos del proyecto. • Estimar riesgos potenciales.
Artefactos
• Un documento con la visión del proyecto. • El modelo de casos de uso con una lista de todos los casos de uso y los

actores que puedan ser identificados. • Un glosario inicial del proyecto. • Un caso inicial del negocio el cual incluye: contexto del negocio, criterios de éxito y planificación financiera. • Un estudio inicial de riesgos. • Requerimientos complementarios. • Un plan del proyecto el cual muestre las fases y las iteraciones.
Hito: Objetivo del ciclo de vida.
Criterios de Evaluación
• Consenso de las partes interesadas en la definición del alcance y la estimación de costos y tiempos. • Fidelidad en los casos de uso primarios. • Credibilidad en los costos y tiempos estimados en el proceso de desarrollo. • Alcance del prototipo de arquitectura desarrollado. • Gastos reales contra gastos planificados.

Si el proyecto no alcanza la concreción del hito entonces debe ser cancelado o rediseñado de manera considerable.

Fase de Elaboración

El propósito de esta fase es analizar el dominio del problema, establecer una base arquitectónica sólida, desarrollar el plan de proyecto y eliminar los elementos de más alto riesgo. Para cumplir con el propósito se debe tener una visión clara del sistema tanto a lo largo como a lo ancho.

En la tabla C.2. se definen los objetivos, artefactos, actividades esenciales y criterios de evaluación correspondientes a esta fase.

Tabla C.2. Descripción de la Fase de Elaboración. Fuente: (Rational,2000)

Objetivos
• Definir, validar y delinear la arquitectura tan rápido como sea posible. • Refinar la visión. • Delinear un plan de alta fidelidad para la fase de construcción. • Demostrar que la arquitectura delineada soportará la visión del negocio, por un costo y tiempo razonables.
Actividades esenciales
• Elaborar la visión y establecer un sólido entendimiento de los casos de uso más críticos que conducen las decisiones de arquitectura y planificación. • Elaborar el proceso, la infraestructura y el ambiente de desarrollo. • Elaborar la arquitectura y seleccionar los componentes.
Artefactos
• Un modelo de casos de uso (completo en al menos un 80%), con todos los actores identificados y la mayor parte de las descripciones de casos de uso. • Requerimientos complementarios: los no funcionales o no asociados con ningún caso de uso.

• Descripción de la arquitectura del software. • Prototipo ejecutable de la arquitectura. • Una lista revisada de riesgos. • Plan de proyecto incluyendo iteraciones y criterios de evaluación para cada iteración. • Manual preliminar del usuario.
Hito: Arquitectura del ciclo de vida.
Criterios de Evaluación
• Estabilidad en la visión del producto. • Estabilidad de la arquitectura. • Nivel de detalle y exactitud del plan de la fase de construcción. Respaldos de las estimaciones. • Acuerdo de las partes interesadas acerca de que el plan logre que se pueda alcanzar la visión al desarrollar el sistema con la arquitectura seleccionada.

Si el proyecto no alcanza la concreción del hito entonces debe ser cancelado o rediseñado de manera considerable.

Fase de Construcción

Durante esta fase se evoluciona desde un prototipo inicial a un producto operacional, todos los componentes y características de la aplicación son desarrolladas e integradas al producto y probadas de manera rigurosa. Es un proceso de manufactura en donde el énfasis está en gerenciar recursos y controlar operaciones.

Tabla C.3. Descripción de la Fase de Construcción. Fuente: (Rational,2000)

Objetivos
• Minimizar los costos de desarrollo. • Lograr la calidad adecuada. • Lograr versiones que pueden ser usadas (alfa, beta, etc).
Actividades Esenciales
• Gerencia de recursos, control de recursos y optimización de procesos. • Desarrollo completo de componentes y pruebas contra los criterios de evaluación definidos. • Desarrollo de versiones del producto.
Artefactos
• El producto de software integrado sobre la plataforma adecuada. • Los manuales del usuario. • Una descripción de la versión actual.
Hito: Capacidad operativa inicial.
Criterios de Evaluación
• Estabilidad y madurez de la versión del producto, capacidad para ser enviada a los usuarios. • Los gastos reales de recursos comparados con los gastos planificados.

Si el proyecto no alcanza la concreción del hito entonces la fase de transición debe ser pospuesta como un *release* (versión).

Fase de Transición

En esta fase lo principal es asegurar que el sistema posee el nivel adecuado de calidad para alcanzar los objetivos; se deben eliminar los errores, entrenar a los usuarios, ajustar las funciones y añadir los elementos faltantes. Se produce y envía un producto final. El propósito de esta fase es efectuar la transición del producto a la comunidad de usuarios.

Tabla C.4. Descripción de la Fase de Transición. Fuente: (Rational,2000)

Objetivos
• Lograr que el usuario se auto-soporte. • Lograr un producto final tan rápidamente y costo efectivo como sea posible.
Actividades esenciales
• Ajustes, incluyendo corrección de errores y mejoramiento para el desempeño y usabilidad. • Empaque, envío, producción, venta y entrenamiento personal.
Piedra de milla: Versión del producto.
Criterios de Evaluación
• Usuarios satisfechos. • Gastos de recursos actuales contra gastos planificados

Es importante destacar que, esta metodología permite dentro de sus fases cierta flexibilidad en el cumplimiento y la realización de las actividades ya que, si bien se especifica qué se debe hacer en algunos casos, no se establece de qué manera ha de hacerse. Además RUP permite trabajar con otras técnicas de modelaje sin que esto impida el cumplimiento de cada una de las fases estipuladas.

ANEXO D - Metodología DESMET

La metodología DESMET ayuda al evaluador de una organización a planear y ejecutar un ejercicio de evaluación que sea imparcial y fiable (Kitchenham, 1996). Este surge de la necesidad de los Ingenieros de Software para evaluar métodos y herramientas utilizadas en esta área. Uno de los problemas en las empresas con respecto a la evaluación de las mismas es que así como se tienen dificultades técnicas existen factores sociológicos y administrativos que desvían un ejercicio de evaluación. A excepción de los experimentos formales, las evaluaciones propuestas en DESMET son dependientes del contexto, lo que revela su carácter sistémico lo que significa que un método no pueda ser considerado el mejor para todas las circunstancias (Kitchenham, 1996).

El método DESMET identifica nueve métodos de evaluación (Kitchenham, 1996):

- **Experimentos Cuantitativos:** una investigación del impacto cuantitativo del método organizado como un experimento formal.
- **Caso de Estudio Cuantitativo:** una investigación del impacto cuantitativo del método organizado como un caso de estudio.
- **Resumen Cuantitativo:** una investigación del impacto cuantitativo del método organizado como un resumen.
- **Selección Cualitativa:** una evaluación basada en características de un simple individuo, quien no solo determina las características a ser valoradas y su rango de escala.

- **Experimentos Cualitativos:** una evaluación basada en características de un grupo de usuarios potenciales quienes esperan probar las salidas del método y herramienta en tareas típicas antes de hacer las evaluaciones.
- **Caso de Estudio Cualitativo:** una evaluación basada en características ejecutadas por quien ha usado el método/herramienta en un proyecto real.
- **Resumen Cualitativo:** una evaluación basada en características donde personas que tienen gran experiencia en le uso de método/herramienta, o tiene estudio del mismo.
- **Análisis de Efectos Cuantitativos:** una valoración subjetiva del cuantitativo del método/herramienta, basado en la opinión de un experto.
- **Benchmarking:** un proceso para correr un número de pruebas de estándares usando métodos/herramientas alternativas y valorando la ejecución relativa de la herramienta contra las pruebas.

Cada uno de estos métodos aplica dependiendo de los requerimientos de la investigación. Estos requerimientos están organizados en dos categorías. Requerimientos prácticos y requerimientos tecnológicos.

Requerimientos Técnicos:

DESMET identifica los requerimientos técnicos como criterios específicos que los Métodos de Evaluación usan para determinar las circunstancias de la evaluación. Estos criterios son:

Contexto de Evaluación: según Kitchenham (1996), DESMET es un ejercicio comparativo y con la excepción de los experimentos formales, las evaluaciones

son dependientes del contexto, es decir, que la identificación del mejor método de evaluación pudiera derivar resultados de distinto orden dependiendo de la manera e la cual se desarrolle. Es importante mencionar esto ya que, en cualquier caso distinto a un experimento formal, se debe tener claro que los resultados de la evaluación son directamente proporcionales a las capacidades de las evaluaciones aplicadas.

En cuanto al contexto de evaluación existen cuatro principales:

- Selección de un conjunto de métodos/herramientas para un proyecto individual.
- Selección de métodos y herramientas para una organización que pueden proyectar alternativas iniciales seguidas de una evaluación más detallada de uno o más métodos o herramientas.
- Monitoreo de cambios como parte de un programa que puede incluir la evaluación de un cambio propuesto y/o la evaluación del efecto de adoptar un nuevo método en una organización.
- Evaluación de métodos/herramientas para la re-venta como parte de una línea de producción.

Naturaleza del Impacto: DESMET identifica dos categorías mayores del impacto:

- Cuantitativa.
- Cualitativa.

Naturaleza del Objeto de Evaluación: el objeto a ser evaluado puede ser

clasificado, según DESMET, usando el siguiente orden:

- Herramienta.
- Método.
- Combinación Método/Herramienta.
- Ambiente de Soporte de Proyecto.

Alcance del Impacto: el alcance del impacto tiene dos dimensiones:

- Granularidad del producto. La granularidad del impacto se identifica con los métodos que aplica al desarrollo o al mantenimiento de un producto de software como un todo; las partes individuales del producto como lo son los módulos o documentos.
- Extensión del impacto. Aquí se identifica el método que puede causar el efecto de un filtro sobre el ciclo de vida del proyecto o del producto.

Madurez del Método: el concepto de madurez se puede visualizar a través de las siguientes categorías:

- No usado en proyectos comerciales, todavía en desarrollo.
- Usado en productos realizados por la misma organización.
- En uso extendido en la propia organización.

Tiempo de Aprendizaje: DESMET presenta el factor método divido en os aspectos:

- Tiempo requerido para entender los principios del método.
- Tiempo requerido para ser experto en su uso.

Madurez de la Organización: DESMET organiza la madurez de organización

en cuatro niveles principales:

- Capacidad de evaluación limitada severamente.
- Capacidad de evaluación cualitativa.
- Capacidad de evaluación cualitativa.
- Capacidad de evaluación cualitativa y cuantitativa.

Capacidad de evaluación a todo nivel.

Requerimientos Prácticos:

Existe un número de restricciones que pueden influenciar la selección final del método de evaluación que mejor se adapta a las necesidades del proyecto, como lo son (Kitchenham, 1996):

- El tiempo que es necesario para diferentes opciones de evaluación.
- La confidencialidad que el usuario puede tener en los resultados de una evaluación.
- El costo de una evaluación.

Para DESMET la evaluación de los requerimientos prácticos tiene dos razones de gran importancia:

- Cuando se quiere establecer una mayor claridad entre el estudio de un caso y un experimento formal, ambos cuantitativos.
- Para asegurar la factibilidad de la evaluación.

Tal como se dijo anteriormente, los requisitos técnicos ofrecen argumentos suficientes para soportar la selección de uno de los métodos de evaluación en comparación con los restantes; en tal sentido, se aplica la tabla de selección de los

métodos de DESMET como se presenta a continuación.

Tabla D.1. Selección de los Métodos de DESMET. Fuente: (Kitchenham, 1996)

Método de Evaluación	Características Favorables
Experimentos Cuantitativos	• Beneficios claramente cuantificables. • Tiempo de aprendizaje relativamente pequeño. • Personal disponible para tomar parte del experimento. • Deseo hacer contribuciones de contexto independiente al método. • Método relacionado con una pregunta sencilla. • Beneficios directamente observables desde los resultados de las tareas.
Casos de Estudio Cuantitativo	• Beneficios cuantificables en un proyecto sencillo. • Procedimientos de desarrollo estables. • Personal con experiencia métrica. • Tiempo para la evaluación proporcional al tiempo transcurrido del proyecto de tamaño normal. • Beneficios cuantificables previo al retiro del producto.
Resumen Cuantitativo	• Beneficios no cuantificables en un solo proyecto. • Proyectos con experiencia de uso del método. • Existencia de una base de datos del proyecto con información de productividad, calidad, información de métodos.
Selección Cualitativa	• Tiempo corto para el ejercicio de evaluación. • Gran número de métodos a evaluar.
Caso de Estudio Cualitativo	• Beneficios difíciles de cuantificar. • Beneficios observables en un simple proyecto. • Procedimientos de desarrollo estables. • Tiempo para la evaluación proporcional al tiempo transcurrido del proyecto de tamaño normal.
Experimento Cualitativo	• Beneficios difíciles de cuantificar. • Beneficios observables en un proyecto sencillo. • Tiempo de evaluación pequeño. • Población de usuarios del método variada.
Resumen Cualitativo	• Beneficios difíciles de cuantificar. • Población de usuarios del método variada. • Beneficios no observables en un proyecto sencillo. • Proyectos realizados para aprender del método.
Análisis de Efectos Cualitativos	• Requerimiento de unión de métodos. • Interés en evaluación de métodos genéricos. • Disposición de opiniones expertas acerca del método. • Carencia de estabilidad en el proceso de desarrollo.
Benchmarking	• Método no intensivo al nivel humano. • Salidas del método capaz de ser comparadas con un ranking en términos de qué tan buenos son.

Con respecto a los tiempos asociados, se presenta una escala de tiempos relativos para cada método.

Tabla D.2. Tiempos Relativos asociados con los Métodos de Evaluación de DESMET. Fuente: (Kitchenham, 1996)

Riesgo Relativo	Método de Evaluación	Comentarios
Largo (Más de 3 Meses)	Estudio de un Caso (Cuantitativo y Análisis de Características)	Un estudio de un Caso puede durar lo que dura el desarrollo del proyecto.
Medio (Varios Meses)	Análisis de Características. Medición	El tiempo necesario para enviar y recibir un cuestionario.
Corto (Varias Semanas)	Experimento (Cuantitativo y Análisis de Características)	Pueden ser realizados en un corto tiempo o en un período viable.
	Benchmarking	El desarrollo apropiado de las pruebas puede tomar algo de tiempo.
	Análisis de Características. Modo Proyección	Tiempo requerido para identificar las características que constituirán las bases para la recolección del material.
Muy Corto (Varios Días)	Cuantitativo. Medición	Asumiendo la existencia de una base de datos, e tiempo que se requiere para extraer la información y analizarla.
	Análisis de Efectos Cualitativos	Asumiendo la existencia de una base de datos, e tiempo que se requiere para extraer la información.

Con respecto a la confianza de los resultados, se presenta una escala de magnitud de riesgo asociado a cada método de evaluación.

Tabla D.3. Riesgos asociados con los Métodos de Evaluación de DESMET. Fuente: (Kitchenham, 1996)

Riesgo Relativo	Método de Evaluación	Comentarios
Muy Alto	Análisis de Efectos Cualitativos	Está basado en una vista subjetiva de un experto sencillo lo cual tiene un alto riesgo de obtener resultados intermedios.
Alto	Estudio de un Caso Cuantitativo	La evaluación del impacto está basada en una comparación de dos proyectos diferentes.
	Análisis de Características, Estudio de un Caso o Modo de Proyección	La evaluación esta basada e la evaluación de una persona y el criterio de evaluación es subjetivo.
Medio	Estudio de Caso Cuantitativo con políticas de la organización	La evaluación del impacto puede usar información acerca de proyectos normales. Este tiene mayor probabilidad de que el resultado sea válido.
	Análisis de Características, Medición	La evaluación está basada en la experiencia práctica de un número de proyectos.
Bajo	Estudio de un Caso Cuantitativo	Replicación del estudio de caso del proyecto con métodos estadísticos que pueden ser usados.
	Análisis de Características, Experimento	La replicación y la muestra aleatoria del estudio del caso reduce el riesgo de una sugestión sistemática, sin embargo, la evaluación es subjetiva.
	Medición Cuantitativa	Replicación a través de proyectos mediante métodos estadísticos pueden causar impacto.

Muy Bajo	Experimento Cuantitativo	El todo de un método científico soporta la integridad de los experimentos.

Con respecto a los costos asociados, se presenta una escala para cada método.

Tabla D.4. Costos asociados con los Métodos de Evaluación de DESMET. Fuente: (Kitchenham, 1996)

Riesgo Relativo	Método de Evaluación	Comentarios
Alto	Experimento	El costo de un experimento es relativo al número de personas que pueden contribuir en pro de él.
Medio	Estudio de un Caso	Los costos estarán relacionados más que todo con los trabajos adicionales para realizar las evaluaciones.
	Análisis de las Características, Medición	Los costos de introducir, enviar y recibir una medición es algo que requiere su esfuerzo, a lo se une el tiempo y costo derivado del entendimiento de aquella persona que debe rellenar el cuestionario.
	Benchmarking	El costo más grande de un banco de datos es de diseñar, probar y correrlo a él con cada uno de los métodos o herramientas.
	Análisis de las Características. Modo Proyección	Esto puede ser realizado por una persona durante unas semanas. El costo se verá incrementado si es necesario entender una especificación de requerimientos detallados de los requerimientos de los usuarios potenciales de dichos métodos.
Bajo	Medición Cuantitativa	Asumiendo la existencia de una base de datos, el análisis de la información para evaluar el impacto del método envuelve un esfuerzo relativamente pequeño.
Muy Bajo	Análisis de Efectos Cuantitativos	Asumiendo la existencia de una base de datos, la extracción de la información para evaluar el impacto del método envuelve un esfuerzo relativamente pequeño

ANEXO E - Modelo de las 4+1 Vistas

Según Kruchten (2000), en la Arquitectura de un sistema intensivo de software se pueden identificar diversas características para diferentes propósitos, entre los principales se encuentran los siguientes:

- Para dirigir la organización lógica del sistema.
- Para organizar la funcionalidad del sistema.
- Para direccionar la concurrencia.
- Para describir la distribución física del software en la plataforma.

Estos propósitos proporcionan las Vistas Arquitectónicas. Se puede definir una Vista Arquitectónica como "*una descripción simplificada, una abstracción del sistema desde una perspectiva particular, cubriendo aspectos particulares y omitiendo aspectos que no son relevantes para una perspectiva*" (Kruchten, 1995).

En la figura E.1. Modelo de las 4+1 Vistas se muestran las cinco vistas mencionadas junto con los elementos más importantes de UML correspondientes a cada vista.

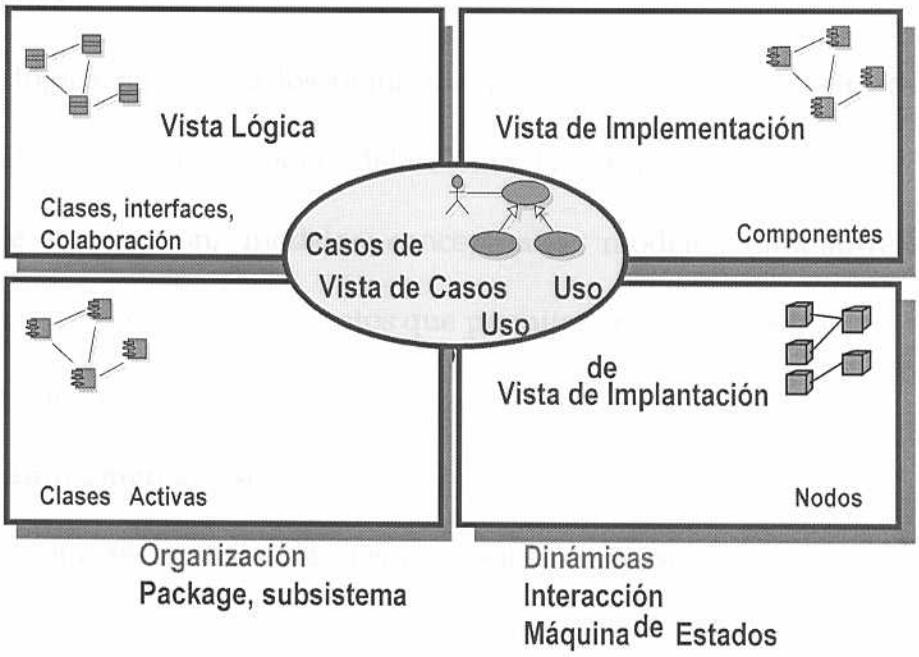

Figura E.1. Modelo de Vistas 4 +1 y elementos de UML.
Fuente: (Kruchten, 2000)

Vista de Casos de Uso

Esta vista es redundante con respecto a las otras porque ésta representa como una especie de compendio de las otras cuatro, de allí que venga el "+1", sin embargo juega dos papeles críticos (Kruchten, 2000):

- Actúa como un manejador para ayudar a los diseñadores a descubrir los elementos arquitecturales durante el diseño de la arquitectura.
- Valida e ilustra el diseño de la arquitectura en papel y también se utiliza como punto de partida de las pruebas de un prototipo arquitectural.

Esta vista contiene los escenarios o casos de uso claves, para cada uno de los

cuales se describen las secuencias de interacción entre objetos y procesos (Kruchten, 2000).

Vista Lógica

La vista lógica representa los requerimientos funcionales, es decir, los servicios que los usuarios finales esperan del sistema. Para representar esta vista se utilizan cartas de navegación, modelos conceptuales, modelos entidad-relación y el diagrama de clases como elementos que permiten mostrar la estructura estática del sistema (Kruchten, 2000).

Vista de Implementación

En esta vista se describe la descomposición del sistema dentro de capas y subsistemas. Este aspecto puede ser parcialmente reflejado en el contenido de la Vista Lógica, pero en esta vista se utiliza otro nivel de detalle (Kruchten, 2000).

Vista de Procesos

Direcciona el aspecto concurrente del sistema en tiempo de corrida, muestra las tareas o procesos así como interacciones. En esta vista se controla todo lo relacionado con procesos concurrentes que se encarguen de realizar alguna tarea (Kruchten, 2000). Ilustra los puntos de concurrencia y paralelismo del sistema.

Los diseñadores describen la vista de procesos en diversos niveles de abstracción, cada uno incluyendo un asunto diferente. A alto nivel, la vista de proceso puede ser vista como un conjunto de redes lógicas de programas de comunicación ("procesos") ejecutándose independientemente, el cual está distribuido a través un conjunto de recursos de hardware, que está conectado por un *"bus"*, una LAN o

una WAN (Kruchten, 2000).

Vista de Implantación

Muestra como varios ejecutables y otros componentes de tiempo de corrida son mapeados a las plataformas principales o nodos de computación. Trata la puesta en marcha, instalación y rendimiento (Kruchten, 2000).

En la Vista de Implantación son tomados en cuenta los requerimientos no funcionales del sistema tales como la fiabilidad, la mantenibilidad, la portabilidad, la usabilidad, etc. (Kruchten, 2000).

una WAN (Kruchten, 2000).

Vista de Implantación

Muestra como varios ejecutables y otros componentes de tiempo de corrida son mapeados a las plataformas principales o nodos de computación. Trata la puesta en marcha, instalación y rendimiento (Kruchten, 2000).

En la Vista de Implantación son tomados en cuenta los requerimientos no funcionales del sistema tales como la fiabilidad, la mantenibilidad, la portabilidad, la usabilidad, etc. (Kruchten, 2000).

ANEXO F - Rapide

Es un lenguaje Orientado a Objeto, concurrente basado en eventos, diseñado específicamente para el prototipado de arquitecturas y sistemas distribuidos. Cada sistema está conformado por muchos componentes ejecutándose de manera independiente bajo diferentes restricciones. Es un ADL que permite la simulación y análisis del comportamiento de los sistemas en los primeros pasos del proceso de desarrollo (Mann et al., 1993).

Está diseñado para definir arquitecturas y simular su comportamiento, usando simulación causal de eventos (Nordstrom, 1998).

El lenguaje RAPIDE completo consiste de cinco sub-lenguajes, ellos son:

- **the Types Language y the Predefined Types** los cuales son un caso especial de tipos de interfaz
- **the Pattern Language**
- **the Executable Language**
- **the Architeture Language, y**
- **the Constraint Language.**

En la figura F.1., se muestra la relación entre estos sub-lenguajes. Donde la relación **Es Sub-lenguaje**, es transitiva. Las producciones de un sub-lenguaje, son un sub-grupo de las producciones de un super-lenguaje. Por ejemplo, **the Constraint Language** es un sub-lenguaje de **the Architeture Language**, y **the Executable Language**, pero si se observa la Figura 19, gracias a la transitividad, **the**

Types Language es un sub-lenguaje de todos.

RAPIDE está basado en criterios de diseño cuya idea fundamental es: proveer constructores de arquitecturas que le permitan a la arquitectura del sistema, ser expresada de forma ejecutable para simulación, antes de que las decisiones de implementación hayan sido tomadas, permite además adoptar un modelo de ejecución que capture el tiempo y el comportamiento tan precisamente como sea posible. Este ADL provee restricciones formales y **mapping**, así como también direcciona tópicos de escalabilidad involucrados en el modelado de arquitecturas para sistemas industriales (Mann et al., 1993).

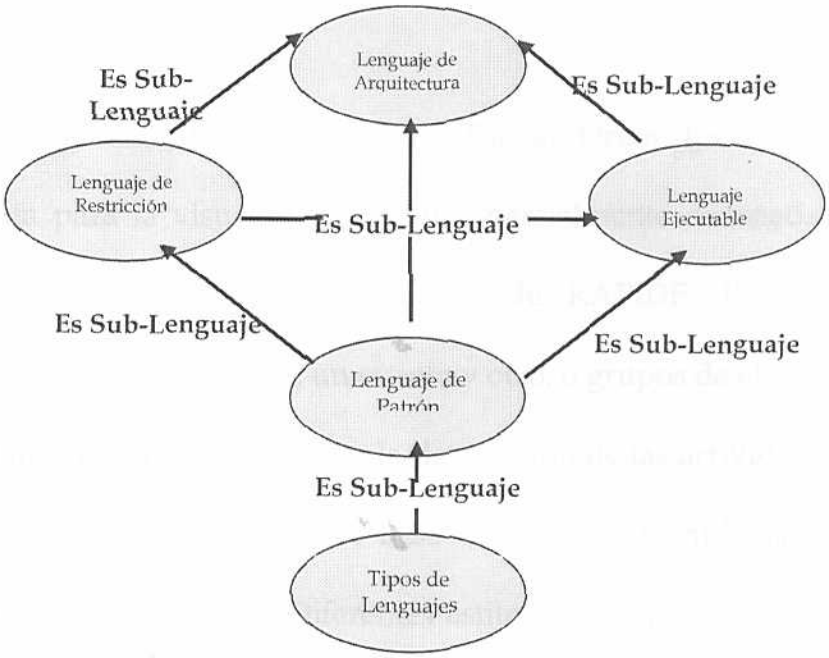

Figura F.1. Sub-lenguajes de RAPIDE y su relación.
Fuente: (Stafford, 1997).

RAPIDE cuenta con cinco herramientas, dentro de las cuales se incluyen un editor de arquitecturas, un compilador, un verificador de restricciones en tiempo de ejecución, un **browser** gráfico, y una herramienta que facilita la animación, todas

estas serán descritas a continuación (Stafford, 1997).

El **Editor de Arquitectura** es denominado **Raparch (The Rapide Architecture Editor)**: esta herramienta permite editar gráficamente arquitecturas; utilizando cajas y líneas, pueden ser dibujados y editados los módulos y conexiones de una arquitectura.

El **Compilador** (Rpdc, **Rapide Compiler**): analiza el código fuente de RAPIDE, reporta los errores de sintaxis y semántica y genera un ejecutable.

El **Verificador de restricciones** en tiempo de ejecución (**RTS, Constraint Checking Runtime System**): maneja la distribución de los recursos durante la ejecución además de comprobar que las ejecuciones (**computation**) producidas no violen las restricciones del modelo.

El **Browser** gráfico denominado Pov (**Partial Order (poset) Viewer**): es una herramienta para la visualización gráfica parcialmente ordenada de los eventos generados por los archivos ejecutables de RAPIDE. Permite visualizar la dependencia entre los eventos, un evento y otro, o grupos de ellos.

Raptor (The Rapide Animator): es la descripción de las actividades de un sistema en un dibujo del mismo. Éste se basa en el paso de mensajes a través de un diagrama de cajas y flechas. Diferentes estilos de animación son apropiados para distintos sistemas. Además la animación es una ayuda para el entendimiento humano; de la misma forma es una herramienta poderosa para prototipos de arquitecturas. En RAPIDE una Arquitectura es un módulo que contiene un conjunto de módulos llamados componentes, reglas de conexión que definen la

manera en la cual se realiza la conexión entre componentes y un grupo de restricciones (Stafford, 1997).

Una manera de ver gráficamente los elementos de una Arquitectura RAPIDE es como se muestra en la figura F.2:

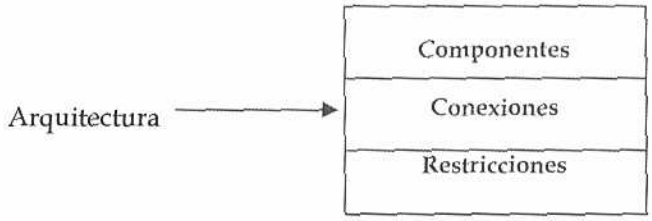

Figura F.2. Elementos de una arquitectura Rapide.
Fuente: (Stafford, 1997).

Componentes: son módulos activos, en el sentido de que reciben y generan eventos, realizan llamadas y ejecución de funciones. Pueden contener módulos pasivos o activos dentro de ellos. Como una regla estilística, pero no una restricción del lenguaje, ni un módulo activo ni una referencia para activar módulos, puede ser pasada como parámetro de una acción o la llamada de una función. Los módulos activos, sólo pueden comunicarse a través de sus interfaces (Stafford, 1997).

Los componentes pueden por ellos mismos, ser arquitecturas. Este provee una capacidad simple para desarrollar una arquitectura estructurada jerárquicamente más que una estructura plana (de un sólo nivel) de comunicación. Cuando una interfaz en una arquitectura, es asociada con un módulo o con otra arquitectura, la arquitectura resultante es llamada "instancia" de la original (Kenny y Park, 1999).

Un componente de una arquitectura es un módulo, por ejemplo, un objeto de un tipo de interfaz. Este puede ser visto de tres maneras: como una interfaz sencilla, como una arquitectura que implementa una interfaz, o un módulo que implementa una interfaz (Stafford, 1997). Esto puede ser visto en la figura F.3.

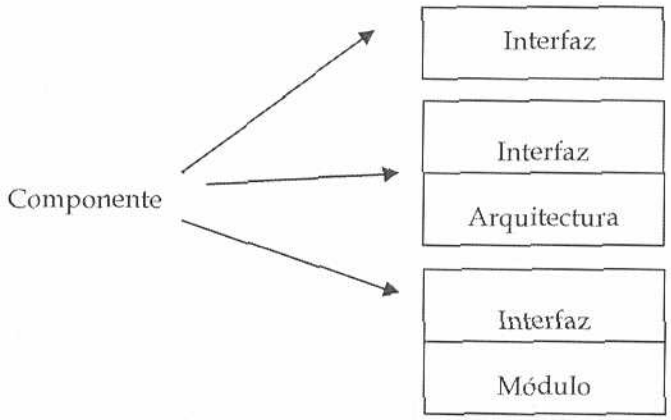

Figura F.3. Tipos de Componentes.
Fuente: (Stafford, 1997).

Interfaces: un tipo interfaz describe la encapsulación de los componentes del sistema. Describe la interfaz de los componentes del sistema con otros componentes. Un tipo interfaz define un tipo de objeto llamado módulo. **Módulos**: los módulos, son definidos por un grupo de procesos (posiblemente vacíos) que verifican eventos y reaccionan a ellos por ejecución arbitraria de eventos, que pueden generar nuevos eventos.

Conexiones: una conexión entre componentes, define equivalencias entre llamadas a funciones de componentes o dependencias entre eventos generados y recibidos por componentes. Una conexión entre interfaces de la arquitectura y sus

componentes, definen equivalencias entre llamadas a funciones con nombre, que forman parte de la interfaz, y llamadas a funciones de componentes, o definen dependencia entre eventos de las acciones de las interfaces y eventos de las acciones de los componentes. Una regla de conexión puede depender sobre las condiciones de ejecución y puede conectar un variado número de componentes.

Restricciones: en RAPIDE las restricciones están dadas por las restricciones realizadas sobre los patrones. El comportamiento de un programa RAPIDE puede ser restringido por las restricciones de los patrones, que están basados en las especificaciones de los patrones. Las restricciones de los patrones denotan los patrones o eventos y estados que son aceptables o no (Kenny y Park, 1999).

Un elemento en común para componentes, conexiones y restricciones es **Map**.

Map: es un constructor que define reglas de **mapping** que transforma la ejecución de una o más arquitecturas (dominio) en una ejecución de otra arquitectura o una interfaz (rango). Una función puede ser el dominio de otra función, así estas pueden estar compuestas transitivamente, es decir, puede definir cómo la ejecución de una arquitectura puede ser interpretada como ejecución de otra. Una función puede contener restricciones que evitan el orden en el cual sus reglas pueden ser iniciadas. Una función es grupo de constructores de lenguaje experimental para relacionar diferentes niveles en un diseño jerárquico (Mann et al., 1993).

Los programas de RAPIDE 1.0 generan eventos e interactúan con ellos, lo cual se convierte en una característica principal de RAPIDE.

Ejecución basada en eventos

Una ejecución es un grupo de eventos con un orden parcial que relaciona eventos en el grupo. Este orden parcial en la ejecución de RAPIDE 1.0 representa dependencia entre eventos y el tiempo en el cual los eventos pasan con respecto a varios relojes. La relación de dependencia es también llamada relación causal ya que modela cuando son causados y cuando ocurren los eventos (Stafford, 1997).

Un **Evento**, es un objeto generado por una llamada a una acción. Una declaración de acción define un tipo de evento asociado, este tipo de evento es el tipo de evento generado por las llamadas a esa acción. Cada acción llamada genera un nuevo evento el cual es distinto de todos los eventos previos (Stafford, 1997).

Por otra parte, los **Patrones de Evento** son expresiones que describen órdenes parciales de eventos. Un patrón actúa como una plantilla que puede adecuarse a un orden parcial de eventos, sí el evento y el orden entre ellos satisfacen la semántica de los patrones (Stafford, 1997). RAPIDE permite especificar una arquitectura de dos formas; una de ellas es con una Arquitectura de Conexión de Interfaces (en inglés, **Interface Connection Architectures** - ICA) y la segunda, es hacerlo con una Arquitectura de Conexión de Objetos (en inglés, **Object Connection Architectures** - OCA). Generalmente se realizan las especificaciones tipo ICA ya que:

- Una ICA, puede definirse antes de que los módulos del sistema hayan sido construidos (Santoro y Park, 1995); siendo esta característica muy útil al momento de seguir los pasos dentro de un método de evaluación ya que, en

este primer paso, sólo deben definirse las interfaces de la arquitectura del sistema.

- Las conexiones entre los módulos, están definidas explícitamente por las conexiones entre las interfaces (Santoro y Park, 1995). La Figura xxxx muestra el *tipo de arquitectura* ICA

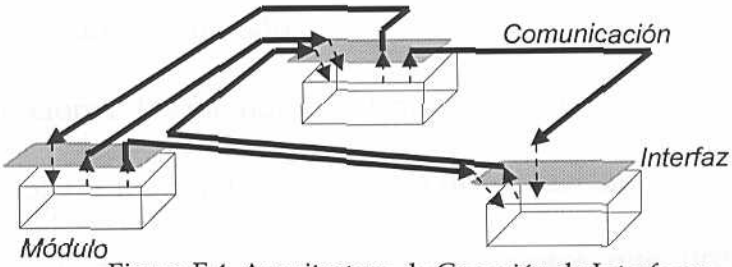

Figura F.4. Arquitectura de Conexión de Interfaces.
Fuente: (Santoro y Park, 1995)

este primer paso, sólo deben definirse las interfaces de la arquitectura del sistema.

- Las conexiones entre los módulos, están definidas explícitamente por las conexiones entre las interfaces (Santoro y Park, 1995). La Figura xxxx muestra el tipo de arquitectura ICA

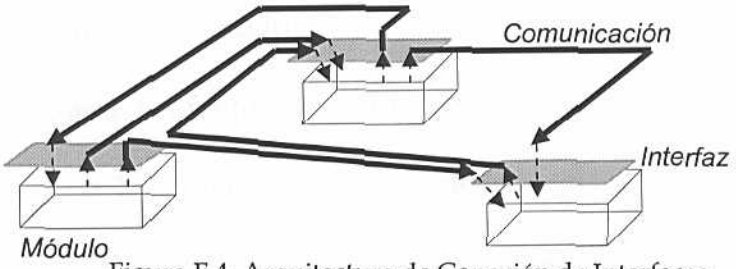

Figura F.4. Arquitectura de Conexión de Interfaces.
Fuente: (Santoro y Park, 1995)

ANEXO G- Análisis de Características, Caso de Estudio.

DESMET define un ejercicio de evaluación de un caso de estudio como la aplicación de un método o herramienta en un proyecto real (Kitcheham, 1996). Kitcheham (1996), propone unas actividades principales para elaborar la evaluación según este método:

- Seleccionar los métodos o herramientas a evaluar.
- Establecer los criterios y la escala de evaluación.
- *Seleccionar uno o más proyectos pilotos que probarán los métodos o* herramientas.
- Probar los métodos o herramientas en cada uno de los proyectos pilotos.
- Ponderar cada una de las cualidades del método o herramienta.
- Analizar los resultados.

Cada actividad involucra una serie de artefactos que son regularmente el resultado de la evaluación, útiles para el análisis final.