



INSTITUTO POLITÉCNICO NACIONAL

Centro de Investigación en Computación

Maestría en Ciencias de la Computación

La Evaluación de los Productos de Software Asistida por el Sistema MECA.

TESIS QUE PARA OBTENER EL GRADO DE MAESTRO EN
CIENCIAS DE LA COMPUTACIÓN PRESENTA:
ALMA DELIA CUEVAS RASGADO.



Dedicatoria

A Dios.

Agradezco infinitamente su presencia constante a través de este largo camino; unas veces doloroso, otras placentero, pero siempre con la plena confianza de que cada evento fuese con la finalidad de acrecentar mi fe y lograr mi crecimiento moral y profesional.

A mi familia.

Mis padres: Cirilo Cuevas (†) porque creyó que podía lograrlo y María Luisa Rasgado Santiago quien siempre me encomendó a San Vicente Ferrer en situaciones difíciles.

Mis hermanos: Alejandro, Enedino, María Concepción, Lucelia, Lilia, María Isabel y María Luisa quienes me han dado su apoyo moral y han sido ejemplo a seguir en mi afán lograr mis propias metas.

Ing. Jose Luís Sosa López

Docente del Instituto Tecnológico de Oaxaca

Este documento es el resultado de la confianza que acrecentó en mí, al proporcionarme la oportunidad de crecer como profesional; cuya motivación, en muchos casos, ayudó a que mi perseverancia y tenacidad no decayeran a lo largo de este desafiante camino.

Al Instituto Tecnológico de Oaxaca.

Agradezco el apoyo incansable al lugar donde he crecido como profesional, cuya confianza que ha depositado en mí, me ha inducido a aprender las nuevas tecnologías informáticas e iniciar una nueva ruta poco incurrida por nosotros los oaxaqueños: en de la Investigación, en el cual me comprometo a trabajar siempre en beneficio de nuestra institución y por ende, de nuestro estado de Oaxaca.

Al Instituto Tecnológico del Istmo.

mi alma mather.

Donde nací como profesional y cuyas carencias me indujeron a crecer académicamente compitiendo con los egresados de las mejores escuelas del país, siempre con la finalidad de mejorar el ejercicio académico y docente en beneficio del mi raza zapoteca, mi región el Istmo de Tehuantepec y por mi estado de Oaxaca.

A Consejo del Sistema Nacional de Educación Tecnológica .

Agradezco todo su apoyo durante mi estancia en el CIC-IPN.

A la Dirección General de Institutos Tecnológicos.

Por su incansable apoyo, que me ha permitido desarrollarme en el ejercicio docente y actualmente en el inicio de mi participación en el área de la investigación.

Al Dr. Bárbaro Jorge Ferro Castro:

Por su apoyo incondicional al iniciar este recorrido, mismo que en principio fue doloroso, pero con su confianza pude continuar mi propio desafío y lograr crecer como profesional.

A mis maestros, miembros del jurado:

Por haberme regalado su valioso tiempo, apoyo y motivación. Agradezco la amistad que me han brindado durante este quehacer académico y espero que prevalezca aún a través de la distancia.

A Luis Antonio Gama Moreno.

Agradezco su apoyo y comprensión como testigo fiel y participante activo con sus consejos que minimizaron los problemas que surgieron en el desarrollo de este trabajo de postgrado.

A mis amigos.

Jorge Manjarréz, Rubén Peredo, Joaquín Medina y David Desiga quienes me ayudaron a resolver mis dudas.

Walter Cuevas, Cutberto García y Jorge Martínez que aportaron un granito de arena en el desarrollo de este trabajo.

Dr. Jesús Figueroa que con su comprensión y sabiduría ayudó a disminuir mis preocupaciones.

Sergio, Javier y Elena que simplemente me invitaron a distraerme cuando lo necesité.

Y a aquellos que me animaron desde la tribuna.

Les agradezco infinitamente su apoyo.

Respetuosamente.

Alma Delia Cuevas Rasgado

Índice de páginas

	Pág.
Capítulo I	
Introducción.....	1
1.1 Planteamiento del problema.....	2
1.2 Justificación.....	2
1.3 Hipótesis.....	3
1.4 Objetivos.....	3
1.4.1 Objetivo general.....	3
1.4.2 Objetivos específicos.....	3
1.5 Conocimiento de los requerimientos.....	3
1.5.1 Presentación general.....	3
1.5.2 Clientes o Usuarios.....	4
1.6 Metas.....	4
1.7 Funciones del sistema.....	4
Capítulo II	
Los modelos cualimétricos.....	6
2.1 El modelo de McCall.....	6
2.2 El modelo de Boehm.....	8
2.3 El estándar IEEE 1061.....	9
2.4 El estándar ISO/IEC 9126.....	10
2.5 El modelo SATC.....	12
2.6 El modelo MECA.....	14
2.7 Comparación de los modelos de calidad.....	15
2.8 La Calidad en Uso.....	19
2.8.1 Definición de la Calidad en Uso.....	19
2.8.2 El modelo para la Calidad en Uso.....	20
2.8.3 Calidad y Satisfacción.....	20
2.8.4 Calidad en Uso de un software.....	21
2.8.5 Software que no tiene Calidad en Uso.....	21
2.8.6 Implementación de la Calidad en Uso en el Sistema MECA.....	21
2.9 Evaluación de la Satisfacción.....	22
2.9.1 Encuesta relativa a la Satisfacción del usuario.....	23
2.9.2 Modelo matemático para la Evaluación de la Calidad en Uso.....	25
Capítulo III	
Análisis y diseño del Sistema MECA.....	27
3.1 Modelado de la arquitectura del sistema MECA.....	27
3.1.1 La vista de los casos de uso.....	28
3.1.1.1 Diagrama de casos de uso.....	28
3.1.1.2 Diagrama de secuencia.....	32
3.1.2 Vista de diseño.....	33
3.1.2.1 Diagrama de colaboración.....	33
3.1.2.2 Diagrama de estados.....	35
3.1.2.3 Diagrama de actividades.....	37
3.1.2.4 Diagrama de clases.....	38
3.2 Esquema de la Base de datos.....	39
3.2.1 Diseño de la Base de datos.....	40
Capítulo IV	
Desarrollo, implementación y prueba del sistema.....	49
4.1 Aplicaciones Web.....	50
4.2 Consideraciones mínimas de hardware y software.....	50

4.3 Arquitectura del Sistema MECA.....	51
4.4 Instalación del Sistema MECA.....	55
4.5 Prueba e Implementación.....	56
4.5.1 Resultados por orden de evaluadores.....	56
4.5.2 Seguimiento de la Calidad en Uso del Sistema MECA.....	62
Capítulo V	
Conclusiones y recomendaciones.....	65
5.1 Trabajos a Futuro.....	67
Bibliografía.....	90

Índice de anexos

Anexo	Pág.
A Cuadro resumen de la medición, clasificación, categorización y conclusiones de la evaluación de la calidad del software.....	69

Índice de apéndices

Apéndice	Pág.
A Diagramas de secuencia.....	71
B Diagramas de colaboración.....	77
C Cálculo de los Puntos de Función.....	79
D COCOMO Básico.....	88

Índice de figuras

Figura	Nombre	Pág.
2.1	Esquema de la Calidad del Software de McCall	7
2.2	Modelo de Calidad de Software de Boehm.....	9
2.3	Estructura de las métricas de Calidad de Software según el estándar IEEE-1061 [19].....	10
2.4	Modelo de Calidad de Software de la Norma ISO/IEC 9126.....	11
2.5	Calidad en el ciclo de vida del producto	20
2.6	Encuesta a la Calidad en Uso.....	24
2.7	Informe de la encuesta a la Calidad en uso.....	25
3.1	Caso de Uso general.....	29
3.2	Diagrama de secuencia de la Evaluación individual.....	33
3.3	Diagrama de colaboración de la Evaluación.....	35
3.4	Diagrama de estados del Sistema MECA.....	36
3.5	Diagrama de actividades del Sistema MECA.....	37
3.6	Diagrama de clases y relaciones del Sistema MECA.....	39
3.7	Arquitectura de la Base de Datos.....	40
3.8	Modelo Lógico Entidad-Relación	41
3.9	Secuencia de ejecución de la Evaluación.....	48
4.1	Diagrama de despliegue del Sistema MECA.....	52
4.2	Interacción cliente / servidor.....	52
4.3	Diagrama de componentes del Administrador.....	53
4.4	Diagrama de componentes del Evaluador Formal.....	54

4.5	Diagrama de componentes del Evaluador Informal.....	55
4.6	Diagrama de interrupción de la Evaluación.....	56
4.7	Métricas complejas para el evaluador Tres	59
4.8	Métricas complejas en la evaluación	60
4.9	Resultado del producto tres	61
4.10	Módulos implementados al sistema MECA para el apoyo del evaluador	61
4.11	Pantalla de ayuda de la primera etapa de la metodología MECA	62
4.12	Calificación de la Calidad en Uso del Sistema MECA.....	62
4.13	Pantalla de ayuda a la encuesta.....	63
4.14	Representación gráfica de la Calidad en Uso del sistema.....	64
AP.A.1	Diagrama de secuencia de alto nivel de abstracción del Registro de evaluadores....	71
AP.A.2	Diagrama de secuencia de bajo nivel de abstracción del Registro de evaluadores...	72
AP.A.3	Diagrama de secuencia del Registro de Proyectos	72
AP.A.4	Diagrama de secuencia de bajo nivel del Registro de Proyectos.....	73
AP.A.5	Diagrama de secuencia de la Asignación de Proyectos	73
AP.A.6	Diagrama de secuencia de bajo nivel de la Asignación de Proyectos.	74
AP.A.7	Diagrama de bajo nivel de la Evaluación individual	76
AP.B.1	Diagrama de colaboración del Registro del Evaluador	77
AP.B.2	Diagrama de colaboración del Registro del Proyecto	77
AP.B.3	Diagrama de colaboración de la Asignación de Proyectos	78
AP.C.1	Computación de los Puntos de Función [36]	85
AP.C.2	Cálculo de los Puntos de Función Ajustados [36]	86
AP.C.3	Media de las Líneas De Código de acuerdo al lenguaje de programación [36]	87
AP.D.1	COCOMO Básico.....	89

Índice de tablas

Tabla	Nombre	Pág.
2.1	Modelo SATC	13
2.2	Tabla comparativa de los modelos de calidad	18
3.1	Tabla de Eventos que desencadena la Evaluación	32
3.2	Descripción de las tablas de la Base de Datos MECA	42
3.3	Atributos de la tabla Código	43
3.4	Atributos de la tabla Comentario	43
3.5	Atributos de la tabla Empresa	43
3.6	Atributos de la tabla Etapa	44
3.7	Atributos de la tabla Evaluador	44
3.8	Atributos de la tabla Evaluador_Proyecto	45
3.9	Atributos de la tabla Métrica	45
3.10	Atributos de la tabla Met_Param	46
3.11	Atributos de la tabla Proyecto	46
3.12	Atributos de la tabla Retorno	46
3.13	Atributos de la tabla Tipo_Eval	47
4.1	Comparación del método manual y automatizado del evaluador Dos.....	58
A.1	Resumen de la medición, clasificación, categorización y conclusiones de la Evaluación de la Calidad del Software.....	69
AP.A.1	Secuencia de eventos del Registro	71
AP.A.2	Sección Evaluador	72
AP.A.3	Sección Proyecto	73
AP.A.4	Asignación de Proyecto	74
AP.C.1	Tabla de entradas de usuario	80
AP.C.2	Tabla de salidas de usuario	81
AP.C.3	Tabla de peticiones de usuario	83
AP.C.4	Tabla de archivos lógicos	84
AP.C.5	Computación de los Puntos de Función del Sistema MECA [36]	85

Resumen

El trabajo que a continuación se presenta es el desarrollo de una herramienta para automatizar un Modelo Cualimétrico para la Evaluación de la Calidad del Software MECA, el cual se basa en la ampliación y desarrollo de los principios expuestos en la Norma ISO/IEC 9126 (International Organization for Standardization, Organización Internacional para la Estandarización / International Electrotechnical Commission, Comisión Internacional Electrotécnica).

Los modelos cualimétricos actuales no cuentan con herramientas que automaticen su uso, y el disponer de un instrumento de este tipo, facilita la aplicación y divulgación del modelo.

La herramienta ha sido elaborada utilizando los principios y lineamientos de la metodología UML (Unified Modeling Language, Lenguaje Unificado de Modelado) y ha sido programado en Java en ambiente de red, utilizando SQL Server para la Base de Datos.

Además, se fundamentó y elaboró un método para incorporarle al modelo la evaluación de la Calidad en Uso, aporte teórico y práctico de este trabajo. Este método aplica características de la Calidad en Uso y en especial estudia la Satisfacción del cliente con respecto al producto.

Abstract

The current work introduces a software tool designed for the Software Quality Evaluation Model (MECA) automatization. This model is based on the extension and improvement of evaluation principles exposed in the ISO 9126 Norm.

Currently used Quality measurement models, lack of tools which automatize its application. Having such type of instruments facilitate model's application and spreading.

The introduced tool is based on principles and lineaments taken from the Unified Modeling Language Methodology; it has been codified with the Java language and it is oriented to networking environments; SQL Server is used as the Data Base Management System.

Furthermore, a new method was designed and implemented in order to incorporate the evaluation of Quality in Use to the MECA model; this represents both a theoretical and practical contribution. This method applies the characteristics of the Quality in Use and specially studies the satisfaction of customers.

Índice de páginas

	Pág.
Capítulo I	
Introducción.....	1
1.1 Planteamiento del problema.....	2
1.2 Justificación.....	2
1.3 Hipótesis.....	3
1.4 Objetivos.....	3
1.4.1 Objetivo general.....	3
1.4.2 Objetivos específicos.....	3
1.5 Conocimiento de los requerimientos.....	3
1.5.1 Presentación general.....	3
1.5.2 Clientes o Usuarios.....	4
1.6 Metas.....	4
1.7 Funciones del sistema.....	4
Capítulo II	
Los modelos cualimétricos.....	6
2.1 El modelo de McCall.....	6
2.2 El modelo de Boehm.....	8
2.3 El estándar IEEE 1061.....	9
2.4 El estándar ISO/IEC 9126.....	10
2.5 El modelo SATC.....	12
2.6 El modelo MECA.....	14
2.7 Comparación de los modelos de calidad.....	15
2.8 La Calidad en Uso.....	19
2.8.1 Definición de la Calidad en Uso.....	19
2.8.2 El modelo para la Calidad en Uso.....	20
2.8.3 Calidad y Satisfacción.....	20
2.8.4 Calidad en Uso de un software.....	21
2.8.5 Software que no tiene Calidad en Uso.....	21
2.8.6 Implementación de la Calidad en Uso en el Sistema MECA.....	21
2.9 Evaluación de la Satisfacción.....	22
2.9.1 Encuesta relativa a la Satisfacción del usuario.....	23
2.9.2 Modelo matemático para la Evaluación de la Calidad en Uso.....	25
Capítulo III	
Análisis y diseño del Sistema MECA.....	27
3.1 Modelado de la arquitectura del sistema MECA.....	27
3.1.1 La vista de los casos de uso.....	28
3.1.1.1 Diagrama de casos de uso.....	28
3.1.1.2 Diagrama de secuencia.....	32
3.1.2 Vista de diseño.....	33
3.1.2.1 Diagrama de colaboración.....	33
3.1.2.2 Diagrama de estados.....	35
3.1.2.3 Diagrama de actividades.....	37
3.1.2.4 Diagrama de clases.....	38
3.2 Esquema de la Base de datos.....	39
3.2.1 Diseño de la Base de datos.....	40
Capítulo IV	
Desarrollo, implementación y prueba del sistema.....	49
4.1 Aplicaciones Web.....	50
4.2 Consideraciones mínimas de hardware y software.....	50

4.3 Arquitectura del Sistema MECA.....	51
4.4 Instalación del Sistema MECA.....	55
4.5 Prueba e Implementación.....	56
4.5.1 Resultados por orden de evaluadores.....	56
4.5.2 Seguimiento de la Calidad en Uso del Sistema MECA.....	62
Capítulo V	
Conclusiones y recomendaciones.....	65
5.1 Trabajos a Futuro.....	67
Bibliografía.....	90

Índice de anexos

Anexo	Pág.
A Cuadro resumen de la medición, clasificación, categorización y conclusiones de la evaluación de la calidad del software.....	69

Índice de apéndices

Apéndice	Pág.
A Diagramas de secuencia.....	71
B Diagramas de colaboración.....	77
C Cálculo de los Puntos de Función.....	79
D COCOMO Básico.....	88

Índice de figuras

Figura	Nombre	Pág.
2.1	Esquema de la Calidad del Software de McCall	7
2.2	Modelo de Calidad de Software de Boehm.....	9
2.3	Estructura de las métricas de Calidad de Software según el estándar IEEE-1061 [19].....	10
2.4	Modelo de Calidad de Software de la Norma ISO/IEC 9126.....	11
2.5	Calidad en el ciclo de vida del producto	20
2.6	Encuesta a la Calidad en Uso.....	24
2.7	Informe de la encuesta a la Calidad en uso.....	25
3.1	Caso de Uso general.....	29
3.2	Diagrama de secuencia de la Evaluación individual.....	33
3.3	Diagrama de colaboración de la Evaluación.....	35
3.4	Diagrama de estados del Sistema MECA.....	36
3.5	Diagrama de actividades del Sistema MECA.....	37
3.6	Diagrama de clases y relaciones del Sistema MECA.....	39
3.7	Arquitectura de la Base de Datos.....	40
3.8	Modelo Lógico Entidad-Relación	41
3.9	Secuencia de ejecución de la Evaluación.....	48
4.1	Diagrama de despliegue del Sistema MECA.....	52
4.2	Interacción cliente / servidor.....	52
4.3	Diagrama de componentes del Administrador.....	53
4.4	Diagrama de componentes del Evaluador Formal.....	54

4.5	Diagrama de componentes del Evaluador Informal.....	55
4.6	Diagrama de interrupción de la Evaluación.....	56
4.7	Métricas complejas para el evaluador Tres	59
4.8	Métricas complejas en la evaluación	60
4.9	Resultado del producto tres	61
4.10	Módulos implementados al sistema MECA para el apoyo del evaluador	61
4.11	Pantalla de ayuda de la primera etapa de la metodología MECA	62
4.12	Calificación de la Calidad en Uso del Sistema MECA.....	62
4.13	Pantalla de ayuda a la encuesta.....	63
4.14	Representación gráfica de la Calidad en Uso del sistema.....	64
AP.A.1	Diagrama de secuencia de alto nivel de abstracción del Registro de evaluadores....	71
AP.A.2	Diagrama de secuencia de bajo nivel de abstracción del Registro de evaluadores...	72
AP.A.3	Diagrama de secuencia del Registro de Proyectos	72
AP.A.4	Diagrama de secuencia de bajo nivel del Registro de Proyectos.....	73
AP.A.5	Diagrama de secuencia de la Asignación de Proyectos	73
AP.A.6	Diagrama de secuencia de bajo nivel de la Asignación de Proyectos.	74
AP.A.7	Diagrama de bajo nivel de la Evaluación individual	76
AP.B.1	Diagrama de colaboración del Registro del Evaluador	77
AP.B.2	Diagrama de colaboración del Registro del Proyecto	77
AP.B.3	Diagrama de colaboración de la Asignación de Proyectos	78
AP.C.1	Computación de los Puntos de Función [36]	85
AP.C.2	Cálculo de los Puntos de Función Ajustados [36]	86
AP.C.3	Media de las Líneas De Código de acuerdo al lenguaje de programación [36]	87
AP.D.1	COCOMO Básico.....	89

Eliminado: 3

Eliminado: 4

Eliminado: 5

Índice de tablas

Tabla	Nombre	Pág.
2.1	Modelo SATC	13
2.2	Tabla comparativa de los modelos de calidad	18
3.1	Tabla de Eventos que desencadena la Evaluación	32
3.2	Descripción de las tablas de la Base de Datos MECA	42
3.3	Atributos de la tabla Código	43
3.4	Atributos de la tabla Comentario	43
3.5	Atributos de la tabla Empresa	43
3.6	Atributos de la tabla Etapa	44
3.7	Atributos de la tabla Evaluador	44
3.8	Atributos de la tabla Evaluador_Proyecto	45
3.9	Atributos de la tabla Métrica	45
3.10	Atributos de la tabla Met_Param	46
3.11	Atributos de la tabla Proyecto	46
3.12	Atributos de la tabla Retorno	46
3.13	Atributos de la tabla Tipo_Eval	47
4.1	Comparación del método manual y automatizado del evaluador Dos.....	58
A.1	Resumen de la medición, clasificación, categorización y conclusiones de la Evaluación de la Calidad del Software.....	69
AP.A.1	Secuencia de eventos del Registro	71
AP.A.2	Sección Evaluador	72
AP.A.3	Sección Proyecto	73
AP.A.4	Asignación de Proyecto	74
AP.C.1	<u>Tabla de entradas de usuario</u>	80
AP.C.2	<u>Tabla de salidas de usuario</u>	81
AP.C.3	<u>Tabla de peticiones de usuario</u>	83
AP.C.4	<u>Tabla de archivos lógicos</u>	84
AP.C.5	Computación de los Puntos de Función del Sistema MECA [36]	85

Eliminado: 2

Glosario

Actividad. Es una ejecución no atómica en curso, dentro de una máquina de estados.

Acción. Es una computación atómica ejecutable que produce un cambio en el estado del modelo o la devolución de un valor.

Administrador. persona que desempeña el papel de evaluador, auditor, contralor de la calidad del software.

Aseguramiento de la Calidad del Software (ACS). Proceso integral del ciclo de vida del software que comprende la administración y la evaluación. En el cual se establecen las políticas, los objetivos, las normas, los procedimientos, las actividades, los planes y los sistemas orientados a asegurar la calidad del software a través de su ciclo de vida.

Atributos. Son los elementos más detallados de evaluación del software y que deben ser medibles con el propósito de que sean útiles, los cuales permiten clasificar y determinar el nivel de calidad alcanzado por el software específico.

Calidad. La palabra calidad designa el conjunto de atributos o propiedades de un objeto que nos permite emitir un juicio de valor acerca de él; en este sentido se habla de la nula, poca, buena o excelente calidad de un objeto.

Calidad del software. Concordancia del producto de software con los requerimientos funcionales y de rendimiento explícitamente establecidos, con las normas de desarrollo y de documentación vigentes y con las características implícitas que se espera de todo software desarrollado profesionalmente.

Características. Son propiedades del software, que permiten describir y evaluar su calidad.

Característica de Calidad. La ISO/IEC 9126 establece que cada característica de calidad está dada por un "conjunto de atributos que demuestran la capacidad" sobre algunos aspectos del software que pueden ser refinados a través de múltiples niveles de subcaracterísticas.

CASE. (Computer Assisted Software Engineering, Ingeniería de Software Asistido por Computadora). Es la automatización del software. Es una filosofía que se orienta a la mejor comprensión de los modelos de empresa, sus actividades y el desarrollo de los sistemas de información.

Caso de uso. Es una descripción de un conjunto de secuencias de acciones que un sistema ejecuta y que produce un resultado de interés, para un actor particular.

Clase. Es una descripción de un conjunto de objetos parecidos.

Componente. Es una parte física y reemplazable de un sistema que se integra de una o más interfaces y proporciona la realización de dicho conjunto.

Cualimétrico. Modelo que establece la definición de requerimientos y la elaboración de los procedimientos y de las herramientas para garantizar que sea alcanzado el nivel de calidad de un producto.

Diagrama. Es una representación básica de un conjunto de elementos, que se dibujan como un grafo conexo de nodos (elementos) y arcos (relaciones).

Diagrama de actividades. Muestra el flujo de actividades del sistema, los objetos que actúan y sobre los que se actúa.

Diagrama de casos de uso. Representa un conjunto de casos de uso, actores y sus relaciones. Se usa para organizar y modelar el comportamiento del sistema.

Diagrama de clases. Muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones.

Diagrama de componentes. Muestra un conjunto de componentes y sus relaciones.

Diagrama de colaboración. Muestra la organización estructural de los objetos que envían y reciben mensajes.

Diagrama de despliegue. Es una simple colección de nodos y arcos cuyas relaciones son de dependencia y asociación.

Diagrama de estados. Máquina de estados formada por estados, transiciones, eventos y actividades, estos diagramas son importantes para modelar el comportamiento de una interfaz, una clase o una colaboración.

Diagrama de objetos. Muestra un conjunto de objetos y sus relaciones.

Diagrama de secuencia. Es un diagrama de interacción que resalta la ordenación temporal de los mensajes; es decir, representa un conjunto de objetos y los mensajes enviados y recibidos por ellos.

Diagramas estructurales. Son los aspectos estáticos de un sistema, aquellos que representan su esqueleto y su andamiaje, ambos relativamente estables.

Estado. Es una condición o situación en la vida de un objeto durante la cual satisface alguna condición, realiza alguna actividad o espera algún *evento*.

Evaluación de la calidad del software. Es el proceso de medición, clasificación y determinación de la calidad de un elemento de evaluación, sobre la base de los atributos seleccionados para determinar si satisface los requisitos implícitos y explícitos preestablecidos por el productor y por el cliente respectivamente.

Evento. Es la especificación de un acontecimiento significativo que ocupa un lugar en el tiempo y en el espacio, puede activar una transición.

IEEE. El instituto de ingenieros en Electricidad y Electrónica (IEEE, por sus siglas en inglés) fue fundado en 1884, con Alexander Graham Bell y Thomas Alva Edison en la lista de fundadores. Actualmente es la más grande sociedad técnica profesional del mundo.

ISO/IEC 9126. (the International Organization for Standardization, Organización Internacional para la Estandarización / International Electrotechnical Commission, Comisión Internacional Electrotécnica) define seis características de calidad del software con la finalidad de abarcar todos los aspectos de la calidad definidos en ISO.

Máquina de estados. Se usa para modelar el comportamiento de objetos reactivos.

MECA. Modelo Cualimétrico para la Evaluación de la Calidad del Software.

Métricas. Son los estándares de medición que se aplican a los atributos del software en cada una de sus etapas para hacer la evaluación y determinar el grado de calidad existente en el software.

Middleware. Es un módulo intermedio que actúa como conductor entre sistemas permitiendo a cualquier usuario comunicarse con varias fuentes de información que se encuentran conectadas por una red.

Modelo. Se crea para obtener una idea más simple de la realidad y comprender mejor el sistema. Con UML se construyen modelos a partir de bloques de construcción básicos tales como clases, interfaces, colaboraciones, componentes, nodos, dependencias, generalizaciones y asociaciones.

Modelo matemático. Colección de símbolos, un conjunto de proposiciones que describe cualesquiera características pertinentes de estos símbolos y un conjunto de expresiones matemáticas que los relacionan.

Norma. Documento establecido por consenso y aprobado por un organismo reconocido, que provee, para el uso común y repetitivo, reglas, directrices, características para actividades dirigido a alcanzar el nivel óptimo de orden en un concepto dado.

Objeto. Es una cosa, generalmente extraída del vocabulario del espacio del problema o del espacio de la solución.

Objeto reactivo. Es aquel para el que la mejor forma de caracterizar su comportamiento es señalar su respuesta a los eventos lanzados desde fuera de su contexto. Tiene un ciclo de vida bien definido, cuyo comportamiento se ve afectado por su pasado.

Servlet. Es una clase en lenguaje Java que puede recibir peticiones (normalmente HTTP) y generar una salida (normalmente HTML, WML o XML).

SQL Server. Es un motor de bases de datos capaz de soportar millones de registros por tabla con una interfaz intuitiva y con herramientas de desarrollo integradas como Visual Studio 6.0 o .NET, además incorpora un modelo de objetos totalmente programable con el que se puede desarrollar cualquier aplicación que manipule componentes de SQL Server, es decir, hacer aplicaciones para crear Bases de Datos, tablas, DTS, backups, etc.

Subcaracterísticas. Son las particularidades que se generan de las características y que describen sus requisitos implícitos y explícitos preestablecidos por el productor y por el cliente respectivamente.

Transición. Es una relación entre dos estados que indica que un objeto que esté en el primer estado realizará ciertas acciones y entrará en el segundo estado cuando ocurra un evento especificado y se satisfagan unas condiciones especificadas.

UML (Unified Modeling Language, Lenguaje de Modelado Unificado) es un lenguaje estándar para escribir planos de software. Puede utilizarse para visualizar, especificar, construir y documentar los elementos de un sistema que involucran una gran cantidad de código.

Usuario formal. Persona que es designada oficial o formalmente por una institución o empresa.

Usuario informal. Persona que no necesita una designación oficial, ni una empresa para evaluar la calidad de un software, podrá ingresar libremente al sistema y evaluar su producto.

Vista. Es una proyección de la organización y estructura de un modelo del sistema, que se centra en un solo aspecto.

Organización del documento

Este trabajo de tesis está estructurado de cinco partes: En el *capítulo I* se presenta un panorama general del origen del modelo de calidad MECA, el problema que resuelve al ser implementado en un producto de software, los objetivos que ha cubierto con esta implementación; así mismo, se dan a conocer los requerimientos de la herramienta, tales como usuarios y funciones.

En el *capítulo II* se presentan las cualidades resumidas de los modelos cualimétricos que han servido como base en el desarrollo de la metodología que implementa el Sistema MECA; se expone un cuadro comparativo de estos modelos y se explican las bondades de MECA. Se propone un modelo para la evaluación de la Calidad en Uso como un concepto imprescindible en el diseño del producto, se presentan sus subcaracterísticas, atributos y se elige a la Satisfacción como subcaracterística principal con la finalidad de conocer el grado de calidad en el desempeño del producto.

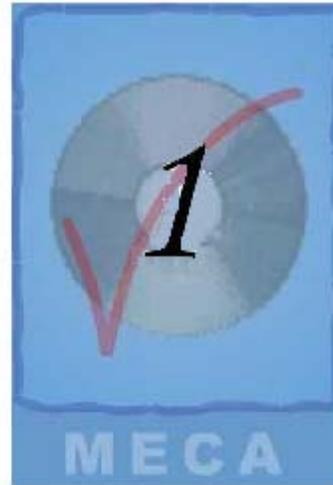
En el *capítulo III* se analizan las dos primeras fases del ciclo de vida del software en el cual se identifican las características del dominio de información, integrada por la contabilización de las pantallas de entradas y salidas, la cantidad de componentes y archivos que interactúan en el proceso y la Base de datos. Se presenta la fórmula para calcular los puntos de función basándose en la complejidad de la información administrada así como en un estudio de factibilidad usando el método COCOMO básico. Además, se define el modelado de la arquitectura cliente/servidor del sistema MECA, integrada por tres vistas: la de los casos de uso, diseño y Base de Datos cada una con sus detalles particulares.

En el *capítulo IV* se muestra el desarrollo de la aplicación, en ella se define y describe el comportamiento de las aplicaciones Web. En la implementación, se contemplan las consideraciones mínimas de software y hardware con las que debe contar el equipo para fungir como servidor o cliente, la arquitectura del Sistema MECA y los prerrequisitos de instalación que dan lugar al funcionamiento correcto del sistema. En la prueba se presenta la evaluación de tres productos de software usando el Sistema MECA.

En el *capítulo V* se presenta la conclusión de la tesis; se contemplan las ventajas que se derivan de la implementación de MECA al ser una herramienta de apoyo en el proceso de auditoría de sistemas y para desarrollar productos de calidad en el mercado; se citan los trabajos planeados a futuro que lo enriquecerán. Se expone de manera general las propuestas que han surgido de esta tesis y que aportarán elementos importantes a su funcionalidad con la finalidad de conocer su visión a futuro y su aportación a la sociedad informática.

Además el trabajo incluye un glosario con una lista de términos que complementan los temas; varios anexos que muestran gráficamente su contenido y una lista de referencias bibliográficas, en la cual se han apoyado algunos de sus conceptos.

Introducción



Actualmente, en el ámbito de la producción de software, la calidad es indispensable, no sólo para conservar una parte del mercado, luchando contra una competencia cada vez más preparada, sino como el resultado del movimiento global dentro del proceso de mejoramiento continuo de los estándares de producción en todos los sectores productivos.

La industria del software es un sector donde el concepto de calidad total ha generado una revolución radical. La producción industrial de software es una actividad relativamente joven. La demanda de software y la complejidad del producto parecen crecer más que las metodologías, el personal capacitado y las herramientas para automatizar la producción.

A pesar de la incorporación de herramientas CASE (Computer Assisted Software Engineering, Ingeniería de Software Asistido por Computadora), la producción de software continúa siendo una actividad con alta participación de recursos humanos, cien por ciento intelectual y en cierto sentido, sin insumos ni materias primas.

Estas circunstancias han producido una prolongada crisis del software, donde los productos se entregan con demoras, los desarrollos exceden lo inicialmente presupuestado y muchas veces no cumplen con los requerimientos originales, extendiéndose tanto a la dificultad de proporcionar mantenimiento, como al cumplimiento de criterios de calidad.

En consecuencia, en el CIC (Centro de Investigación en Computación) del IPN (Instituto Politécnico Nacional) se ha elaborado un Modelo Cualimétrico para la Evaluación de la Calidad del software (MECA), el cual se basa en el estándar ISO/IEC 9126 [23] (International Organization for Standardization, Organización Internacional para la Estandarización / International Electrotechnical Commission, Comisión Internacional Electrotécnica). Sus métricas, están contenidas en una

biblioteca de fórmulas estudiadas en la tesis “La Actualización y Automatización de un Modelo Cualimétrico para la Evaluación de Calidad del Software”. [8].

La automatización del modelo incluye todas las métricas que aparecen en la norma, relacionadas con la elaboración de los productos de software, y al analizar el comportamiento de las características definidas dentro de cada etapa, incluye otras más. Sin embargo, no se abarca con suficiente especificidad la característica denominada Calidad en Uso.

1.1 Planteamiento del problema

El control de la calidad del software es una materia que preocupa tanto a los productores como a los usuarios. Originalmente, la calidad de un programa o sistema se evaluaba de acuerdo al número de defectos por cada mil líneas de código. Hoy en día, el concepto moderno de Calidad del Software, requiere de una congruencia total entre los requerimientos y características del producto, para lograr una plena satisfacción del usuario. Surgen componentes de la calidad, tales como: confiabilidad, flexibilidad, facilidad de uso, integridad, consistencia, etc; es decir, se quieren productos que se puedan transportar, sean fáciles de mantener y/o ampliar, sencillos de entender, de validación accesible, compatibles con otros sistemas, rápidos y efectivos, etc.

La evaluación de la Calidad del Software, en la práctica, requiere de métricas que se aplican a los atributos del software en cada una de sus etapas, para evaluarlas y determinar su grado de calidad. En la actualidad, no se conoce una herramienta que realice esta actividad, por tal motivo, en este trabajo se propone la implementación de un sistema que automatice el Modelo Cualimétrico para la Evaluación de la Calidad del Software (MECA) y que representa un apoyo importante en la solución del problema.

1.2 Justificación

La medición es algo común en el mundo de la ingeniería: se mide el consumo de energía, el peso, las dimensiones físicas, la temperatura, el voltaje, la relación señal-ruido, etc. Por desgracia, la medición es menos común en el mundo de la ingeniería del software, ya que existen problemas al acordar en qué y cómo medir. La única forma racional de mejorar cualquier proceso es medir sus atributos, desarrollar un juego de métricas significativas y utilizarlas para proporcionar indicadores que conduzcan a una estrategia de mejora del producto final.

Por esa razón, se requiere que el diseñador conozca los requerimientos de cada etapa del ciclo de vida del software. Una vez implementados estos requerimientos, se pueden valorar de acuerdo a una metodología de evaluación y determinar el grado de la calidad del producto. Todo software implementado con miras a ser evaluado, debe contar con un ejecutable o documentación; con el propósito de que el auditor observe sus atributos (internos y externos) y emita un resultado.

1.3 Hipótesis

El desarrollo de un sistema que se ocupe de evaluar la calidad del software utilizando métricas de calidad e incluyendo una nueva característica denominada Calidad en Uso; constituye una herramienta importante que podría incidir favorablemente en el proceso del control de la calidad de los sistemas informáticos.

1.4 Objetivos

A continuación se presenta el propósito de este trabajo:

1.4.1 Objetivo general

Desarrollar una aplicación que apoye el proceso de control de la calidad, evaluando los resultados de cada una de las etapas del ciclo de vida del software, mediante la aplicación de un modelo cualimétrico que siga los lineamientos de la ISO/IEC 9126 [23].

1.4.2 Objetivos específicos

Entre los objetivos particulares o específicos se pueden citar los siguientes:

- Complementar las métricas de MECA con las que integran la característica Calidad en Uso.
- Permitir que el control de la calidad de los sistemas informáticos se realice conforme a los estándares internacionales.
- Desarrollar el sistema utilizando el Paradigma Orientado a Objetos en su diseño, así como el lenguaje Java en su programación, obteniendo la ventaja de su ejecución independiente de la plataforma del sistema operativo.

1.5 Conocimiento de los requerimientos

Todo proyecto basa su éxito en una especificación correcta y exhaustiva de sus requerimientos; éstos se refieren a las descripciones de las necesidades de un usuario, lo que se desea hacer [31].

1.5.1 Presentación general

El Sistema MECA se implementa para ser una herramienta confiable, clara y rápida en la determinación de la calidad de los productos de software.

1.5.2 Clientes o Usuarios

Entre los posibles usuarios del Sistema MECA se encuentran: auditores informáticos, técnicos diseñadores de sistemas, profesionales en computación, analistas, jefes de proyecto, administradores de sistemas y usuarios con los conocimientos básicos de cómputo y que necesiten conocer el nivel de calidad de un producto de software.

1.6 Metas

En términos generales, la meta es la automatización de la evaluación de la Calidad del Software y los elementos que se satisfacen durante este proceso son:

- Obtener una inspección clara y precisa en el control de la calidad de los sistemas informáticos.
- Mediante las métricas de la Calidad en Uso se conocerá el grado de satisfacción del producto hacia los clientes.
- El mecanismo de seguridad que será implementado en el Sistema MECA dará como resultado una correcta administración de la evaluación de productos de software.
- La portabilidad del sistema permitirá la fácil implementación de nuevos elementos a MECA.

1.7 Funciones del sistema

Las funciones corresponden a las acciones que ejecutará el sistema:

- *Funciones de negocio.* Son necesarias para que la evaluación se desarrolle de manera correcta y se pueda certificar el proceso:
 - *Registro* : El registro es el almacenamiento de los datos particulares del(os) evaluador(es) y el(los) proyecto(s). Estos datos se deben almacenar en los archivos del sistema para identificar a que proyecto y evaluador corresponden los resultados de la evaluación.
 - *Asignación*: Se asignan los Proyectos a los Evaluadores que realizarán la valoración de los productos de software.Estas funciones existen para facilitar la correcta administración de la evaluación, por lo cual el administrador es quien las realiza.
 - *Evaluación*: Proceso en el cual se presenta el formato de evaluación a utilizar para verificar los requerimientos del producto especificados en la etapa a evaluar, introduciendo estos datos al Sistema MECA.
 - *Funciones de soporte informático*: Son necesarias para el buen funcionamiento del sistema, proveer un mejor servicio y mantener un nivel de seguridad en la información administrada:
 - *Ayuda*: proporciona un glosario de términos, y una guía como apoyo al funcionamiento de la evaluación.
-

-
- *Seguridad:* al controlar el sistema por medio de la clave de acceso se mantiene un grado de confidencialidad en la información de cada usuario.
 - *Claridad en los mensajes:* se mantiene un tipo de expresión gramatical adecuado a un nivel estándar de entendimiento.
-

Los modelos cualimétricos



La definición de calidad según la ISO/IEC 8402 [20] es: “Conjunto de propiedades y características de un producto o servicio que le confieren aptitud para satisfacer necesidades explícitas o implícitas”. La definición de Evaluación de la Calidad del Software [15] “Es el proceso de medición, clasificación y determinación de la calidad de un elemento de evaluación, sobre la base de los atributos seleccionados para determinar si satisface los requisitos implícitos y explícitos preestablecidos por el productor y por el cliente respectivamente”.

Ambos conceptos solicitan un conjunto de propiedades del producto, así mismo, la verificación de los requisitos o necesidades implícitas y explícitas en él. Estos dos elementos interactúan constantemente para dar paso al surgimiento de los modelos de calidad o modelos cualimétricos.

Los modelos cualimétricos establecen una serie de requisitos de medición que deben cumplir las organizaciones o individuos, que quieran acceder a un nivel de calidad de un producto de software determinado. La estructura de estos modelos de calidad se caracteriza por ser arborescente, donde el tronco representa las propiedades de mayor nivel; las ramas son otros niveles, que determinan subpropiedades y éstas generan otras que se van desglosando en puntos específicos del producto. La finalidad es auscultar todos los detalles que componen la totalidad del producto y determinar su calidad.

A continuación, se presentan en orden cronológico, las características de los principales modelos cualimétricos, haciendo una comparación entre ellos.

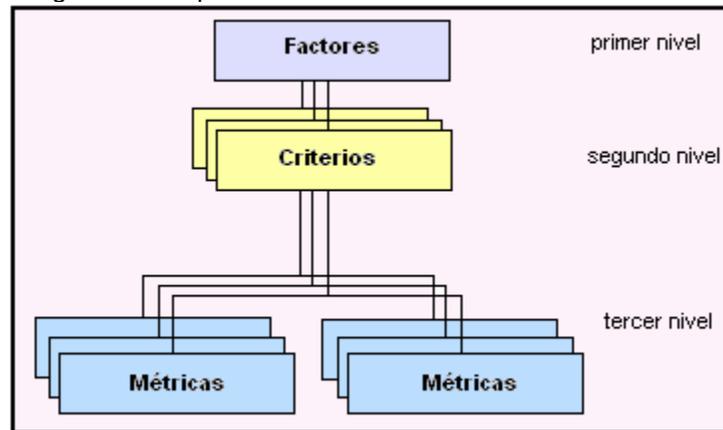
2.1 El modelo de McCall

El modelo más antiguo y que ha servido como guía o patrón para esta actividad es el de McCall [14]. Fue desarrollado entre los años 1976 a 1977 y se conoce también como modelo *GE (General Electric)* ya que fue auspiciado por esta empresa, o como *FCM (Factores/Criterios/Métricas)*, por la forma en que define los niveles de evaluación de las propiedades de calidad.

En la Figura 2.1 se muestra gráficamente la esencia del modelo de McCall. Se fundamenta en la existencia de tres niveles: *primero* se ubican los elementos (factores) que forman la visión general, el propósito del software; *segundo* se encuentran los aspectos técnicos (criterios) y en el *tercero*, los patrones de medición (métricas).

El modelo establece que un *criterio* se puede comunicar con dos o más *factores* (por ejemplo, el criterio *Simplicidad* se encuentra en el factor *Fiabilidad* y en el de *Mantenimiento*), una *métrica* puede afectar el comportamiento de varios *criterios* (por ejemplo, las métricas de *Modularidad* afectan a *Interoperabilidad*, *Transportabilidad*, *Flexibilidad*); es decir, *factores*, *criterios* y *métricas* interactúan constantemente.

Figura 2.1 Esquema de la Calidad del Software de McCall



Los *factores*, se establecen a nivel del cliente (primer nivel) y la administración, estos integran el punto de vista externo del software; es decir, lo que le interesa al usuario. Estas propiedades pueden ser: la *Confiabilidad*, *Mantenibilidad*, y *Usabilidad*.

Los *criterios* se establecen en el nivel de quien diseña el software y se definen como la manera en que deben construirse los componentes del mismo; es decir, aquellos aspectos, tecnologías y funciones que sólo competen al desarrollador del producto. Bajo este concepto se definen la características de *Prueba* y *Eficiencia*.

Las *métricas* se definen a nivel de quienes usan el producto final. Son preguntas cuyas respuestas indican con un valor numérico o alfabético la presencia del atributo evaluado (por ejemplo; para evaluar el criterio *Simplicidad*, se verifica que las funciones requeridas en la prueba sean fáciles de experimentar). McCall propuso dividir la suma de las métricas que cubrían aspectos de calidad entre el número total de ellas, para obtener el valor del criterio. Presenta once características que consideró identificaban la correcta planeación y funcionamiento de los programas de cómputo.

2.2. El modelo de Boehm

El estudio de Boehm [7], lo llevó a identificar un conjunto de características que representaban la forma de medir la calidad de un producto de software.

Un buen producto según Boehm, debía funcionar bien y satisfacer la totalidad de las características inherentes.

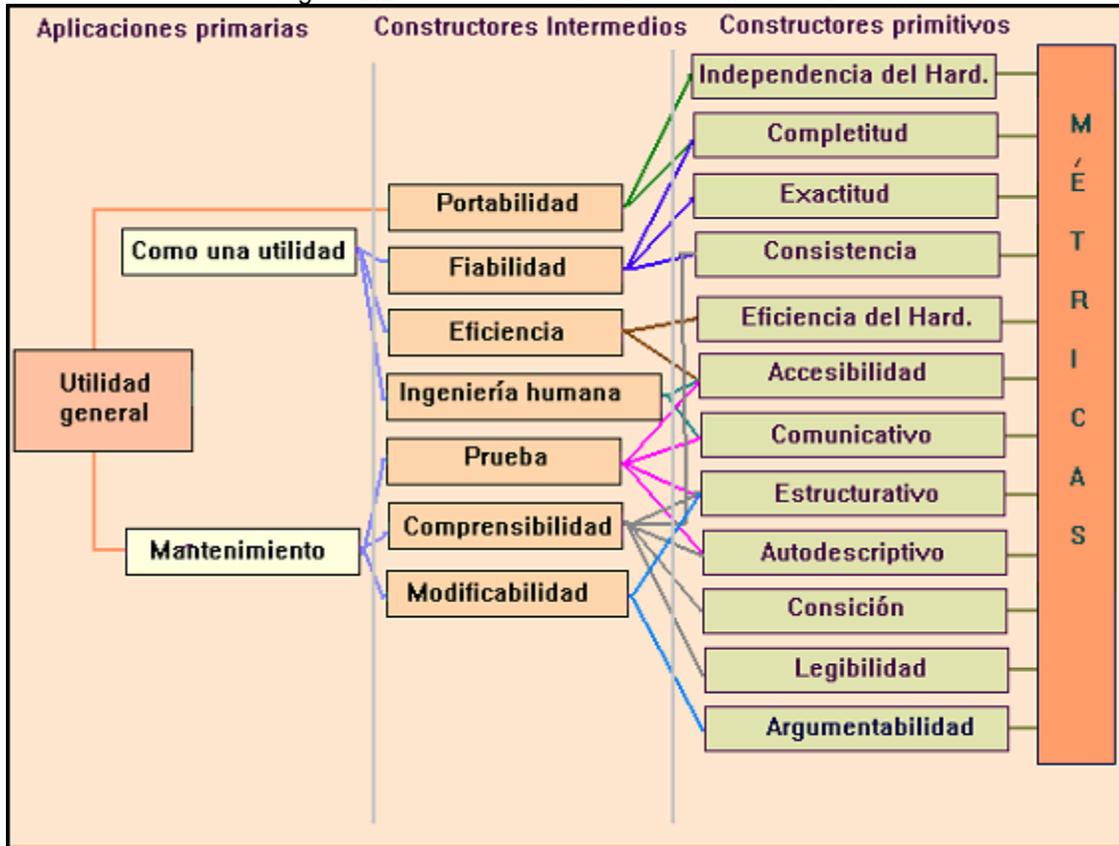
El modelo de Boehm, también conocido como *RLSQF (Rome Laboratory Software Quality Framework, Marco de Trabajo de Calidad del Laboratorio Roma)* ubica al software desde dos perspectivas: Utilidad y Mantenibilidad. Boehm puntualizó que en primer lugar, el producto de software *debe ser útil* es decir, que sirva para un fin u objetivo que sea de provecho; por tal razón, plantea en su modelo la presencia de un usuario final, quien recibe los beneficios de la utilidad y por otro lado, está la perspectiva del usuario de mantenimiento quien adapta el producto de acuerdo a los requerimientos del usuario final (véase la Figura 2.2).

Partiendo de este enfoque se estructuraron tres niveles: En el *primer nivel* se pueden apreciar dos puntos de vista; el primero (del usuario final), donde se encuentran las propiedades del software que pueden ser cuantificables, en este caso es preciso que el programa sea Portable (Portability) a cualquier plataforma de hardware o sistema operativo, Fiable (Reliability), Eficiente (Efficiency), la Ingeniería humana (Human engineering) definida como la capacidad de perfecta comprensión del usuario con respecto al software. El segundo punto de vista (del usuario de mantenimiento), es la habilidad del programador para probar, entender y modificar el software. Se consideran además los criterios de Prueba (Testability), Claridad (Understandability) y Facilidad de Modificación (Modificability), con el propósito de administrar correctamente el proceso y el resultado de la funciones, facilitar el buen desempeño de éstas y finalmente localizar y corregir los errores sin afectar la estructura básica del sistema.

El segundo nivel, conocido como criterios y atributos, se deriva de las propiedades del software. En este nivel se encuentra el Programador primitivo, que realiza cambios o corrige errores detectados en el sistema con el propósito de cubrir todos los requerimientos implícitos y explícitos establecidos por el productor y por el cliente respectivamente.

En el tercer nivel se encuentran las métricas representadas como unidades mínimas de medición. Éstas se requieren para determinar el valor de los criterios del segundo nivel.

Figura 2.2 Modelo de Calidad de Software de Boehm



2.3 El estándar IEEE 1061

El estándar de la IEEE 1061 [19] presenta una metodología para implementar los requerimientos que debe cumplir un producto de software, para determinar su calidad. Se define un conjunto de atributos que sirven para validar métricas durante el proceso de diseño del software y la valoración como producto final, a diferencia de los modelos de McCall y Boehm que no se encuentran definidos en su totalidad.

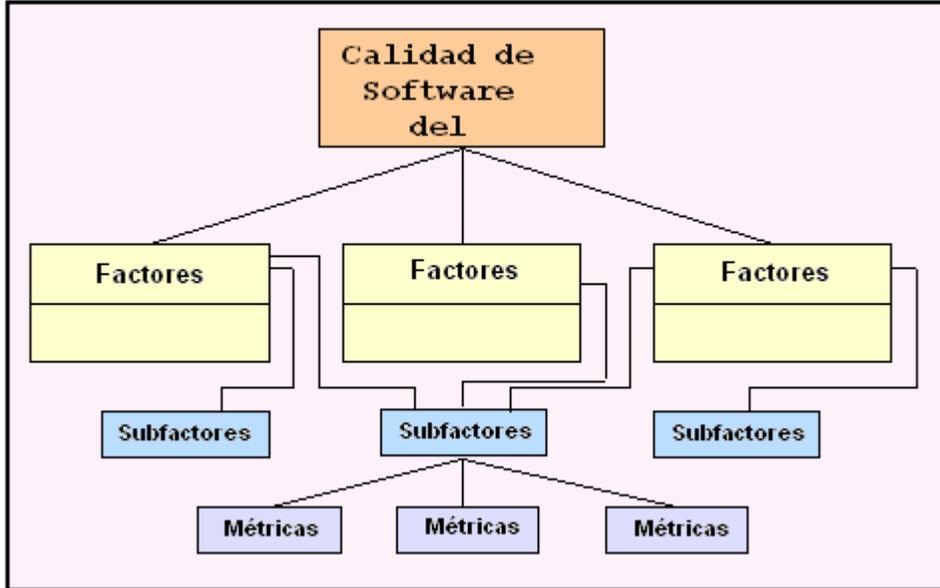
El estándar se concluyó en 1992 y en él no se llega a la especificación de las métricas, pero sí a la forma de aplicarlas.

La Figura 2.3 muestra gráficamente la naturaleza del estándar IEEE 1061. Se dice que su diseño se considera flexible porque permite agregar, modificar y eliminar elementos de la evaluación. Le llama niveles a la posición lógica tipo arbórea en la que se encuentran los elementos que forman el modelo.

En el primer nivel de la jerarquía (*factor*) se encuentran los requerimientos que se deben acordar y definir en el proyecto. Para cada factor hay un valor expresado en números y la suma de estos valores determinará si el factor ha sido alcanzado

o no. En el segundo nivel de la jerarquía (*subfactores*), están los atributos más concretos del software, en donde estos tienen más significado para el personal técnico (analistas, desarrolladores y personal de mantenimiento). En el tercer nivel están las *métricas*, mismas que forman una lista con preguntas específicas derivadas de los subfactores acerca del producto, para verificar el cumplimiento correcto del proceso de elaboración.

Figura 2.3 Estructura de las métricas de Calidad de Software según el estándar IEEE-1061 [19]



2.4 El estándar ISO/IEC 9126

El estándar ISO/IEC 9126 [23] centra su atención en el uso del producto final y define seis características importantes que se deben cumplir para determinar su calidad y que se pueden aplicar a cualquier tipo de software, describiendo la forma de realizar la evaluación.

Se estructura en tres niveles. En el *primero*, están las características que se definen como las propiedades del software, que permiten describir y evaluar su calidad. En el *segundo*, están las subcaracterísticas definidas como las particularidades que se generan de las características y describen los requisitos implícitos y explícitos, preestablecidos por el productor y por el cliente respectivamente. En el *tercer* nivel, están las métricas o atributos que permiten la realización de la medición.

Al combinar las características de la calidad del software desde el punto de vista del usuario, surge el concepto de Calidad en Uso. Este concepto es también aplicable a la perspectiva del usuario de mantenimiento. Sin embargo, existen diferencias entre las capacidades y necesidades de estos usuarios. Por ejemplo, el usuario común

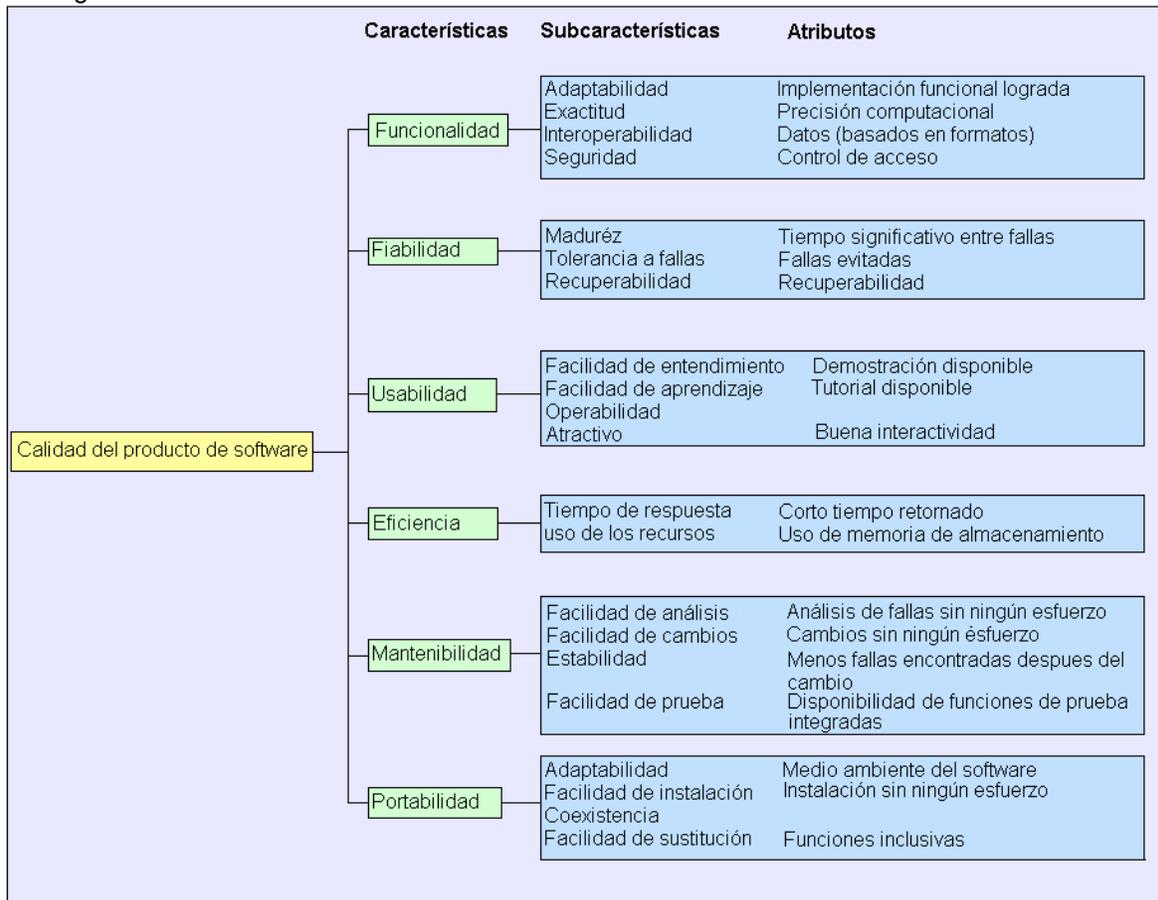
evalúa solo aquellos atributos del software que se usan en su tarea particular y no percibe las del ambiente del hardware y soporte.

La norma describe además las métricas basadas en los atributos internos y externos del producto, éstos se aplican generalmente en los requerimientos de diseño. La norma evalúa las métricas basadas en los atributos internos y externos a través del ciclo de vida del software, con la finalidad de lograr la calidad necesaria y suficiente para el producto.

La *ISO/IEC 9126* propone el modelo que se muestra gráficamente en la Figura 2.4 [23]. Las seis características de la figura se eligieron de acuerdo a los siguientes requerimientos:

- Para que conjuntamente cubran todos los aspectos de la calidad del software que define la ISO.
- Para describir la calidad del producto con un mínimo de superposición.
- Para acercarse tanto como sea posible a la terminología establecida.
- Para formar un conjunto de no más de seis a ocho características, por razones de claridad y manejo.
- Para identificar áreas de atributos de productos y después refinarlas.

Figura 2.4 Modelo de Calidad de Software de la Norma ISO/IEC 9126.



La norma *ISO/IEC 9126* reconoce la Calidad Interna y la Externa.

La Calidad Interna contiene todas las características y detalles del software que se proveen durante la implementación del código, revisión y prueba. La naturaleza de la calidad del producto se representa por las métricas internas. Estas no cambian al menos que se rediseñe el software.

La Calidad Externa comprende aquellas características, como la calidad en la ejecución del software, que se mide normalmente durante la etapa de prueba y con datos simulados. En esta etapa, regularmente la mayoría de las fallas se resuelven. Como es difícil corregir toda la arquitectura del software u otros aspectos de su diseño, su naturaleza, por lo general, se mantiene fija a través de la prueba.

La norma aplica ciertos atributos internos que tienen influencia en el comportamiento externo del software; por ejemplo, se puede medir la *Confiabilidad* (como atributo externo) observando el número de errores en la ejecución del producto y la *Confiabilidad* (como atributo interno), revisando detalladamente las especificaciones técnicas y el código fuente, para conocer el nivel de tolerancia de errores.

La jerarquía no es perfecta, ya que algunos atributos pueden aplicarse en más de una subcaracterística.

2.5 El modelo SATC

El modelo fue creado por la NASA (National Aeronautics and Space Administration, *Agencia Nacional de Aeronáutica y el Espacio*) como un plan estratégico para mejorar el diseño de sus productos; por consiguiente, el modelo SATC (*Software Assurance Technology Center, Centro de Tecnología del Aseguramiento del Software*) puede adaptarse a cualquier software, así como aplicarse en la evaluación del mismo.

El modelo SATC reconoce la existencia de dos criterios:

- el que “trabaja bastante bien” y
- el que esta “disponible cuando se necesita”.

El criterio “trabaja bastante bien”, significa que sus interfaces funcionan, es decir, cumplen los requisitos para los que fueron creados y además se añaden otros como Fiabilidad, Mantenibilidad, Reusabilidad y Corrección.

El criterio “Disponible cuando se necesita”, se refiere a que la ejecución del software es dependiente de la tarea que realiza, de tal manera que los retrasos en la disponibilidad de otro software, podría hacer lento el proceso del primero y por tanto, posponer sus objetivos.

En la Figura 2.1 se muestra la forma en que el modelo SATC, establece su estructura basado en tres niveles: donde en el primer nivel están las *metas*; en el segundo *los atributos*, que se consideran los riesgos del proyecto; obviamente, la mayoría de los proyectos de la NASA requieren un menor grado de riesgo. Los riesgos se clasifican en *métricas*, que constituyen el tercer nivel.

Tabla 2.1 Modelo SATC.

Metas	Atributos	Métricas
Calidad de los requerimientos	Ambigüedad	Número de frases débiles
		Número de frases opcionales
	Completes (Integridad)	Número de TBDs/TBA ^{**}
		* To Be Definite (Para ser definido) ** To Be Added (Para ser añadido)
	Comprensibilidad	Estructura del documento
		Índice legible
Volatilidad	Conteo de cambios y/o Conteo de requerimientos	
	Etapa del ciclo de vida cuando se hacen los cambios	
Trazabilidad	Número de requerimiento de software no identificados en los requerimientos del sistema	
	Número de requerimiento de software no identificados en código y prueba	
Calidad del producto (código)	Estructura/Arquitectura	Complejidad lógica
		Uso de GOTO
		Tamaño
	Mantenibilidad	Correlación de complejidad lógica/tamaño
	Reusabilidad	Correlación de complejidad lógica/tamaño
	Documentación interna	Porcentaje de comentarios
Documentación externa	Legibilidad del índice	
Efectividad de la información	Uso de recursos	Horas invertidas del personal en actividades del ciclo de vida
	Proporción de completes	Tareas terminadas
		Tareas planeadas terminadas
Efectividad de las pruebas	Corrección	Errores y gravedad
		Tiempo de búsqueda de errores
		Tiempo de corrección de errores
		Localización de código de fallas

2.6 El modelo MECA

La evaluación del producto de software, requiere en la práctica de características que complementen y especifiquen las seis que ofrece el estándar ISO/IEC 9126 [23].

Entre los años 1990 y 1993, en el CDI (Centro de Desarrollo Informático) en la Habana, Cuba, surgió un modelo llamado QUALYSOFT [38] considerado como la base de MECA. Este modelo propone la particularización de las seis características anteriores a través del ciclo de vida del software planteado según lo establecido en la norma ISO/IEC12207 [21]. Al evaluar el ciclo de vida, las características se detallan a través de subcaracterísticas, atributos y métricas, estas últimas mediante preguntas planteadas en un formato para cada etapa y que utiliza el evaluador para su valoración.

Este modelo se ha mejorado constantemente, dando lugar al Modelo Cualimétrico para la Evaluación de la Calidad del Software MECA [15] desarrollado en el *CIC-IPN (Centro de Investigación en Computación del Instituto Politécnico Nacional)* en México, Distrito Federal y en el que se ha agregado una característica más: la *Calidad en Uso*, proponiendo una Metodología para el Aseguramiento de la Calidad del producto de software. De entre otros puntos interesantes, esta metodología sugiere la creación de un comité para la evaluación de la calidad de productos software, característica que otros modelos no consideran.

La metodología examina al software a través de las etapas del ciclo de vida, hecho por el cual la hace consistente. Considera que cada etapa tiene como resultado, una faceta diferente del mismo producto; por esta razón, le da un tratamiento independiente a cada una, evaluando el número de elementos del software, como etapas surjan en él.

Otra particularidad del modelo MECA es considerar que la documentación del software forma parte del mismo, por lo que se trata como un producto más y se evalúa de forma separada.

MECA ha resaltado propiedades importantes de los modelos de calidad anteriores, con la finalidad de realizar una evaluación más completa. Al concentrar sus métricas en una *Biblioteca de fórmulas* [8], permite la adaptación de nuevos elementos de calidad sin afectar su integridad.

Las etapas del ciclo de vida planteados y que el modelo MECA implementa son:

- Estudio, análisis o investigación preliminar
 - Análisis y determinación de los requerimientos del software
 - Diseño
 - Prueba de aceptación
 - Mantenimiento y,
 - Evaluación de la Documentación de soporte del software.
-

Desde el punto de vista de diseño, el modelo se organiza mediante la estructura arborecente de tres niveles; al igual que otros, en el primer nivel se encuentran las *características*, en el segundo las *subcaracterísticas* y en el tercero, los *atributos o métricas*. Cada uno de estos conceptos son equivalentes a los de la norma ISO/IEC 9126.

Desde el punto de vista de gestión, el modelo ofrece una estructura basada en papeles, en la cual se plantea la organización de un comité dedicado al Aseguramiento de la Calidad del Software en la empresa. Uno de los papeles que se deriva del comité es *el Administrador*, quien registra a los evaluadores y proyectos de la empresa o institución; el segundo papel es *el Evaluador*, que a su vez, se clasifica en *Evaluador Formal, Informal y Usuario o Cliente Final*.

El *Evaluador Formal* tiene una designación oficial de una empresa o institución; el modelo recomienda la presencia de un único Administrador en el comité; así mismo, un único comité por empresa. El comité está integrado por un *Presidente* y varios *Evaluadores formales*.

El *Evaluador Informal* no tiene una designación oficial de una institución o empresa. Evalúa un producto de software de forma independiente.

El *Usuario o Cliente Final* es la persona que interactúa directamente con el software, hecho por el cual conoce perfectamente su comportamiento. Este evalúa la Calidad en Uso del producto; La Calidad en Uso se evalúa después de la etapa de prueba de aceptación, bajo la hipótesis de que sólo los usuarios podrán determinar, vía su experiencia cotidiana con el producto, los resultados de su satisfacción con el software.

Cada etapa evalúa sólo un producto de software y es independiente de las otras. Estas etapas contienen un formato donde se solicitan los valores de cada métrica para procesarlos y obtener la calificación del atributo, las subcaracterísticas y las características. Mediante estos resultados se obtiene la calificación de la etapa permitiendo ubicar al software como: Muy bueno, Bueno, Regular o Malo (véase Anexo A, Figura A.1).

2.7 Comparación de los modelos de calidad

La diferencia principal entre los modelos radica en la forma en que se detallan los requerimientos de la evaluación; los modelos McCall, Boehm, SATC y la ISO/IEC 9126 indican los elementos a evaluar, mientras que la IEEE 1061 especifica la forma de realizarlo. El MECA abarca ambos requerimientos, tanto de forma como de contenido.

Una perspectiva es el punto de vista a través del cual se ubican o precisan el uso de las propiedades de la calidad del software. Bajo este concepto radica la otra diferencia entre los modelos de calidad estudiados.

El modelo McCall ubica al producto desde la perspectiva de operación, revisión y transición. Cada una de estas propiedades ubican la calidad del software. El modelo Boehm tiene como perspectivas la utilidad y la mantenibilidad. La IEEE-1061 considera la perspectiva del administrador o director del proyecto y el del usuario final. El modelo SATC considera la calidad de los requerimientos, producto (código), efectividad de la implementación y efectividad de las pruebas, tomando como base principal los riesgos del proyecto. La ISO/IEC 9126 y MECA, además de considerar las propiedades de la calidad interna y externa en el producto, añade la perspectiva del usuario de mantenimiento y el usuario final.

MECA, además presenta las siguientes peculiaridades:

- *Considera la documentación del software como parte del producto.* Los modelos y estándares de calidad anteriores no profundizan en la particularización de medidas que permitan evaluar la documentación del software en su totalidad. Considera que la documentación debe realizarse después del Diseño y en paralelo a la etapa de Prueba e Implementación. Al finalizar el desarrollo y prueba del producto, también termina su documentación, considerándose la última etapa del ciclo de vida del modelo.

En la actualidad, se está trabajando sobre la aplicación de la Reusabilidad en las etapas de Estudio, Análisis o Investigación Preliminar, Análisis y Determinación de los Requerimientos del Software, considerando que hay algunos requisitos que se pueden estandarizar y por ende, reusar.

- *Implementa un mayor número de métricas de calidad, al expandir la evaluación a través de las etapas del ciclo de vida del software.* El modelo propone la evaluación a través de las etapas del ciclo de vida del software sugerido en la ISO/IEC12207 [21].
- *Propone la organización de un Comité Técnico para el Aseguramiento de la Calidad del Software* en la empresa o institución. Ninguno de los modelos estudiados plantean la relación interpersonal para efectos de la evaluación del producto de software.
- *Implementa la Reusabilidad y la Calidad en Uso como dos características añadidas a su biblioteca de componentes.* La norma ISO/IEC 9126 propone un modelo para la evaluación de estas características, pero no especifica las métricas de forma particular.

En la Tabla 2.2 se observan los modelos descritos anteriormente, en ella se encuentran marcadas las propiedades de primer nivel; por ejemplo, el modelo de McCall tiene como propiedad principal o de primer nivel la Fiabilidad, Usabilidad, etc. y así con los otros modelos.

Existen propiedades, factores o características que en unos modelos se reconocen como tal, pero en otros como subpropiedades, criterios o subcaracterísticas como es el caso de la Comprensibilidad, que para el modelo Boehm es propiedad de primer nivel, para la IEEE-1061 es subfactor de la Usabilidad, la ISO/IEC 9126 lo define como Facilidad de Aprendizaje (también

propiedad de la Usabilidad) y MECA lo reconoce como subcaracterística de la Usabilidad.

Al igual que los modelos comparados, las casillas marcadas de MECA, indican las características reconocidas como propiedades de primer nivel. Este modelo abarca casi la totalidad de estas propiedades de los modelos estudiados. En algunos casos, las clasifica como subcaracterísticas y en otros como atributos, pero de alguna manera se encuentran definidos en MECA. Por ejemplo, Comprensibilidad como propiedad de primer nivel para Boehm y SATC, en MECA es una subcaracterística de Usabilidad, la Documentación que resulta ser para Boehm otra propiedad de primer nivel, para MECA es una subcaracterística de la Mantenibilidad; así, la Elasticidad la reconoce como Flexibilidad y es un atributo de la Eficiencia, etc.

En otros casos MECA reúne varias propiedades de primer nivel de los otros modelos en una sola característica, como por ejemplo, la Integridad, Ambigüedad, Completitud, Trazabilidad, Proporción de la completitud y la Volatilidad son subcaracterísticas y atributos de la Funcionalidad. Existe solo una propiedad que no incluye el modelo, la Economía, la razón es porque ésta se aplica en el proceso del producto. MECA se centra en la evaluación del software como producto final.

Tabla 2.2 Tabla comparativa de los modelos de calidad

Características Subcaracterísticas	McCall 1977	Boehm 1978	IEEE-1061 1992	ISO/IEC 9126 1991	MECA 1991	SATC 1996
Fiabilidad	✓	✓	✓	✓	✓	
Usabilidad	✓	✓	✓	✓	Usabilidad	
Comprensibilidad		✓	Usabilidad	(Facilidad de aprendizaje) Usabilidad		✓
Eficiencia	✓	✓	✓	Eficiencia	Eficiencia	
Uso de los recursos						✓
Flexibilidad	✓	✓			Portabilidad	
Reusabilidad	✓	✓	Portabilidad	(Adaptabilidad) Portabilidad	Reusabilidad	✓
Generalidad		✓				
Portabilidad	✓	✓	✓	✓	✓	
Claridad		✓			Usabilidad	Comprensibilidad
Documentación		✓			Mantenibilidad	
Elasticidad		✓			Eficiencia	
Economía		✓	(En tiempo y recursos) Eficiencia			
Mantenibilidad	✓	✓	✓	Mantenibilidad	Mantenibilidad	✓
Comprobabilidad (facilidad de prueba)	✓	✓	(Capacidad de prueba) Mantenibilidad			Corrección
Modificabilidad		✓	(Correctibilidad) Mantenibilidad			
Validéz		✓				
Exactitud	✓	✓			Confiabilidad	
Correctitud		(Exactitud) Confiabilidad	(Corrección) Funcionalidad			✓
Interoperabilidad	✓		Funcionalidad	Funcionalidad		
Funcionalidad			✓			
Integridad	✓	✓		(Coexistencia) Portabilidad	Funcionalidad	Complettitud
Ambigüedad						✓
Complettitud		(Complettitud) Confiabilidad	(Complettitud) Funcionalidad			✓
Trazabilidad						✓
Proporción de la Complettitud		(Complettitud) Confiabilidad	(Complettitud) Funcionalidad			✓
Volatilidad						✓
Efectividad de la Implementación		✓	(Capacidad de pruebas) Mantenibilidad	(Facilidad de pruebas) Mantenibilidad	Mantenibilidad	✓
Estructura/ Arquitectura		Comprensibilidad				✓

2.8 La Calidad en Uso

La calidad de un producto se refleja en la aceptación por parte de sus consumidores y se relaciona con el cumplimiento de las especificaciones proporcionadas por el usuario. En el Modelo para la Evaluación de la Calidad de Software de la norma ISO/IEC 9126, se encuentra definida la Calidad en Uso [23], pero no se incluye de manera explícita una forma para evaluarla. Una de las características de la Calidad en Uso que no se ha tratado totalmente es la Satisfacción, la cual, surge con la competencia, entre los productores de software por acaparar al cliente.

2.8.1 Definición de la Calidad en Uso

La Calidad en Uso se define como “El efecto de un producto que al ser empleado por usuarios específicos, logra sus objetivos particulares con Efectividad, Eficiencia, Productividad, Seguridad y Satisfacción en un determinado contexto de uso” [23].

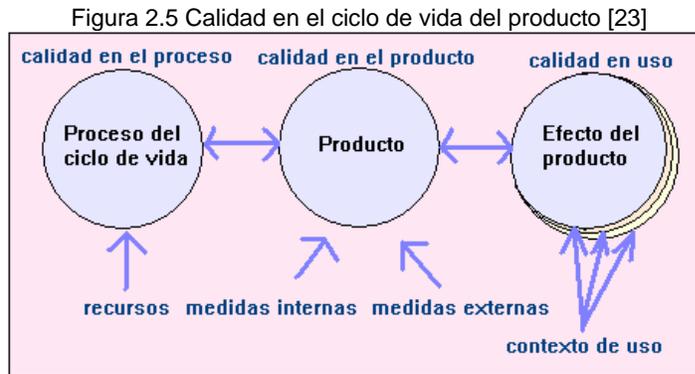
Los usuarios necesitan definir los requerimientos para la Calidad en Uso y bajo esta necesidad, se aplica la calidad interna [25] y externa [24].

Integradas correctamente las medidas internas y externas en el software, provoca un buen efecto en el contexto de uso del cliente, ya que se han vigilado estrictamente los requerimientos de la calidad durante su elaboración, esta característica se refleja en los resultados de la evaluación de la Calidad en Uso.

La meta es lograr la calidad necesaria y suficiente, en cada contexto de uso, ya sea en el de los diseñadores, el personal de mantenimiento o los usuarios finales; especialmente, los usuarios finales son los que determinarán cuando un producto se considera fácil de aprender y usar en términos del tiempo que toman en realizar sus objetivos por lo tanto, es necesario comprender los objetivos del usuario final, los trabajos y tareas que el producto automatiza, modifica o realiza.

Como se muestra en la Figura 2.5, hay tres momentos en los que se debe velar por la calidad durante el ciclo de vida del producto. Primero, *durante el proceso*, en el cual intervienen los recursos y los requerimientos del cliente; luego, en el *producto final*, donde se mezcla la calidad interna y externa. Finalmente, se aprecia *el efecto del producto final* como expresión de la Calidad en Uso.

El cumplimiento de los requerimientos de calidad en el proceso del ciclo de vida del producto redundará en un buen producto y al evaluarlo a través de sus etapas, se reafirma su calidad en el proceso; en caso contrario, se aplica la reingeniería en el producto bajo la etapa que indicó sus deficiencias y se prueba de nuevo. La evaluación de la Calidad en Uso es el efecto del producto en el cliente final.



2.8.2 El modelo para la Calidad en Uso

La ISO/IEC 9126, describe un modelo para evaluar la Calidad en Uso el cual consiste en las siguientes características:

- *Efectividad*. Es la capacidad del producto de lograr metas con *precisión y completitud*, en un contexto de uso especificado.
- *Productividad*. Es la capacidad del producto de emplear eficazmente los *recursos*, en un contexto de uso especificado.
- *Seguridad*. Es la capacidad del software de controlar niveles de riesgo ocasionados por los usuarios, negocios, software y otras propiedades del ambiente en un determinado contexto.
- *Satisfacción*. Es la capacidad del producto de cumplir con las exigencias de los usuarios en un contexto de uso específico.

2.8.3 Calidad y Satisfacción

La satisfacción del usuario es una subcaracterística que no se ha estudiado a profundidad en los modelos de calidad; por consiguiente, es caso de estudio en este trabajo de investigación. La Calidad y la Satisfacción están correlacionadas y su relación, puede verse en los siguientes conceptos:

Satisfacción es :

- El estado psicológico del consumidor que se deriva de la comparación del *rendimiento efectivo* obtenido con el uso de un bien o servicio, así como las expectativas que éste despertó antes de su adquisición [11].
- El resultado de la evaluación cognitiva que hace el cliente ante el servicio que recibe y el sacrificio financiero que realiza [40]. Esta definición relaciona a la satisfacción con la expectativa calidad y precio; otros autores como Kernan y Schmidt [30], añaden las condiciones en las que se produce la compra.

Por consiguiente, puede asegurarse que la medida de la satisfacción es un buen indicio para obtener resultados sobre la calidad del producto. Esto

repercute en una clientela fiel al software, así mismo, garantiza la supervivencia a largo plazo del producto en el mercado.

2.8.4 Calidad en Uso de un software

La Calidad en Uso determinará el grado en que el software está cumpliendo con los requerimientos del cliente y del productor, y el resultado de tenerla redunda en los siguientes aspectos:

- Mejora la productividad y la calidad de las acciones y las decisiones del usuario.
- Reduce el esfuerzo o estrés y permite a los usuarios manejar una variedad amplia de tareas.
- Hace competitivos a los productos en el mercado, que demanda que sean de fácil uso.

2.8.5 Software que no tiene Calidad en Uso

Actualmente, la mayoría de los productos se diseñan con prisa, lo que redunda en un diseño inadecuado. La recopilación de la información acerca de los requerimientos del análisis y diseño no se atiende correctamente, lo que da como resultado un producto con mala Calidad en Uso.

Las consecuencias de una mala Calidad en Uso son:

- Disminuye la salud, bienestar y motivación del usuario, hasta puede incrementar el ausentismo.
- No son explotados en su totalidad. En la medida que el usuario pierde el interés en el uso de las características avanzadas del sistema. En el peor de los casos, estas funciones avanzadas del producto podrían no usarse por parte del cliente final.

2.8.6 Implementación de la Calidad en Uso en el Sistema MECA

La Calidad en Uso está basada en el modo en que los usuarios pueden usar la funcionalidad del producto.

Para la evaluación de la Satisfacción como característica de la Calidad en Uso, se propone una encuesta aplicada al usuario del software, con la cual se proporciona ayuda importante para determinar en qué medida el producto se ajusta a las necesidades y a las expectativas del cliente.

Cada pregunta en la encuesta aplicada a los usuarios finales, solicita una respuesta afirmativa o negativa que corresponde a los atributos de la Calidad en Uso en forma general. Respondidas todas las preguntas se proporciona una calificación para esta característica. Posteriormente, se clasifica al producto bajo los criterios indicados en el Anexo A.

De contar con pocos clientes, se propone aplicar la encuesta a la totalidad de los usuarios; de lo contrario, se determina (a criterio del *Administrador* de la calidad), una muestra de la población de usuarios para que den las respuestas. El *Administrador* avisa a los clientes que participan, vía correo electrónico, teléfono o personalmente, para proporcionarles su clave de acceso al Sistema MECA.

Con el resultado de la encuesta se puede llevar a cabo un seguimiento estadístico y gráfico de cada métrica de la Calidad en Uso, conocer el grado de Satisfacción que las medidas están cubriendo y tomar decisiones para mejorar el producto, si se refleja un resultado no óptimo.

2.9 Evaluación de la Satisfacción

Para medir la Satisfacción se plantean los atributos (métricas) siguientes:

- Apariencia estética.
- Velocidad de respuesta.
- Contenido relevante.
- Funciones adecuadas a la funcionalidad esperada.

Las métricas de *Apariencia Estética* son:

- *Entorno Agradable*. Es preciso proporcionar un ambiente grato, que contribuya al entendimiento de la información por parte del usuario.
- *Comodidad*. En el contexto de uso de un producto, se precisa de una dosis adecuada de confianza, entusiasmo e intencionalidad. La misma tarea que influye en el carácter competitivo y en la confianza de un usuario de forma positiva, puede llevar al aburrimiento y a la frustración en otros.

La comodidad puede surgir porque: el sistema resulta fascinante, constituye un reto, el proceso de aprendizaje resulta satisfactorio, la circunstancia de la novedad resulta atractiva, las posibilidades de mejorar el desarrollo de la tarea son muchas.

Las métricas de la *Velocidad de Respuesta* son:

- *Rapidez*. El tiempo de respuesta es particularmente importante para el software en el que la situación en tiempo real de alguna manera es crítica; por ejemplo cuando hay vidas humanas en riesgo, se esperarían respuestas correctas y rápidas en todos los aspectos del software.
- *Tiempo de Procesamiento*. El sistema ha de caracterizarse por la posibilidad de una comunicación e interacción rápida y efectiva en el muestreo de los resultados de su procesamiento.
- *Respuestas Correctas*. Los resultados se deben basar en términos de estar libre de fallas, de satisfacer los requisitos especificados, así como las necesidades y expectativas del usuario [17].

Las métricas de *Contenido Relevante* son:

- *Claridad y Comprensibilidad.* El software debe ser fácil de entender, independientemente de la experiencia, conocimiento, capacidades lingüísticas o nivel de concentración del usuario. A veces, los mensajes del sistema están hechos en un lenguaje coloquial para proporcionar confianza al usuario y se comete el error de la incomprensibilidad. Se sugiere redactar el mensaje de forma clara, cuanto más claro sea el lenguaje para el usuario menor será la probabilidad de equivocarse en los procesos que ejecute el sistema.

Las métricas de las *Funciones Adecuadas a la Funcionalidad Esperada* son:

- *Utilidad:* Los atributos visuales sean coincidentes al evento o acción que realizan en el sistema, ya que determinados atributos visuales o auditivos, concentran la atención del usuario en la tarea que está desarrollando.
- *Objetividad.* Es preciso tener una definición clara de los objetivos desde el diseño del sistema; entendiendo los requerimientos de los usuarios y contemplando factores como la edad, la experiencia, las limitaciones físicas, las necesidades más especiales, el entorno de trabajo, las influencias sociales y culturales, etc.
- *Seguridad.* La salvaguarda de la información [33] comúnmente trata de la protección del sistema contra el acceso desautorizado o la modificación de información, si está en una fase de almacenamiento, procesamiento o tránsito. También la protección contra la negación de servicios a usuarios no autorizados y la prohibición de servicio a usuarios también no autorizados, incluyendo las medidas necesarias para detectar, documentar y contrarrestar tales amenazas.
- *Administración de los Recursos.* El sistema debe administrar correctamente los recursos de hardware o software, soportar la carga de trabajo, administrar la memoria; ya que en caso contrario el disco duro se llena con tanta información almacenada que el tiempo de respuesta del sistema es cada vez más lento.

2.9.1 Encuesta relativa a la Satisfacción del usuario

En la experiencia de adquirir un producto, el nivel de Satisfacción del cliente depende, en parte, de la comparación de dos clases de expectativas: las de predicción que el cliente espera de un acto de compra y las deseadas, que el cliente desea obtener con esa adquisición [11].

Cuando las expectativas de predicción superan a las deseadas, el cliente estará satisfecho con la adquisición; si coinciden, el cliente supondrá que hizo buena compra y que el producto resultará satisfactorio, pero si las expectativas son menores a las deseadas seguramente el producto provocará insatisfacción en el cliente y su reacción más previsible, será cambiar de proveedor.

La encuesta sobre la satisfacción del cliente que se propone en este trabajo, ha sido adecuadamente diseñada para informar sobre el nivel de Satisfacción

alcanzado por el producto, y permitirá a los clientes, comparar los bienes y servicios sobre una base más amplia y no únicamente atenerse a su precio.

En la Figura 2.6 se aprecian cada una de las respuestas respondidas como “si” y “no”, El resultado de la Calidad en Uso del producto se encuentra ubicado en la parte superior de la figura (primera fila de la pantalla).

Figura 2.6 Encuesta a la Calidad en Uso

Lista de Características, Subcaracterísticas y Atributos	
Calidad en Uso.	100
¿El lenguaje que usa el sistema es claro y comprensible?.	<input checked="" type="radio"/> Si <input type="radio"/> No
¿Es aceptable la respuesta respecto al tiempo de procesamiento?.	<input checked="" type="radio"/> Si <input type="radio"/> No
¿Los iconos que contiene el sistema le revelan la acción que desea realizar?.	<input checked="" type="radio"/> Si <input type="radio"/> No
¿El tamaño, tipo y color de letra, le permiten tener una buena visibilidad en la lectura de los mensajes del sistema?.	<input checked="" type="radio"/> Si <input type="radio"/> No
¿El funcionamiento del sistema cumple con los objetivos por los que fué creado?.	<input checked="" type="radio"/> Si <input type="radio"/> No
¿El sistema satisface los niveles de seguridad requeridos?.	<input checked="" type="radio"/> Si <input type="radio"/> No
¿El resultado del proceso del sistema es correcto?.	<input checked="" type="radio"/> Si <input type="radio"/> No
¿El sistema da respuesta a tiempo a las peticiones del usuario?.	<input checked="" type="radio"/> Si <input type="radio"/> No
¿El sistema administra sus recursos en respuesta a las peticiones del usuario?.	<input checked="" type="radio"/> Si <input type="radio"/> No
¿Existe algún tipo de comodidad al usar el producto?.	<input checked="" type="radio"/> Si <input type="radio"/> No

El *Administrador* registra los datos de los usuarios de la Calidad en Uso en el Sistema MECA, para proporcionarle su acceso a la encuesta. Se sugiere que se precise el tamaño de la muestra de la cantidad de evaluadores de la Calidad en Uso mediante la ayuda de un experto estadístico.

En la Figura 2.7 se presenta el resumen de la evaluación de la Calidad en Uso, para un usuario específico.

Figura 2.7 Informe de la encuesta a la Calidad en uso

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN
Aseguramiento de la Calidad del Software
Informe de Evaluación
ETAPA: Calidad en Uso.

Sistema Subsistema Módulo Programa

Nombre comercial: Sistemas Inteligentes
Versión: 1

EVALUACIÓN

Malo Regular Bien Muy bien Excelente

Puntuación: 100

Fecha de presentación: Fri Feb 14 08:48:38 CST 2003
Comentarios :
Me parece un programa muy excelente

2.9.2 Modelo matemático para la Evaluación de la Calidad en Uso

Para la presentación del informe sobre la calificación de esta característica, se dispone internamente de una fórmula que reconoce dos posibilidades en cada respuesta: “sí” y “no”. El modelo matemático se basa en la probabilidad, dado que el resultado no es determinístico [16], porque se sabe que ocurrirá una respuesta, pero no es posible observar más de dos de ellas.

Cada efecto definido se llama suceso simple, cuyo número es finito (se conoce como el número total de preguntas).

Se usan letras para representar los sucesos simples en el modelo, de manera tal que: a_1 se emplea para denotar el primer suceso, a_2 para el segundo y en general, a_j para el suceso j . Existen n sucesos simples en total:

$$E = \{ a_1, a_2, \dots, a_n \}$$

De hecho, al especificar un conjunto de sucesos E y las probabilidades P , se empieza a construir el modelo matemático.

Para estimar P se considera un conjunto de preguntas relativas a la Calidad en Uso.

$$E = \{ a_1, a_2, a_3, a_4, a_5, \dots, a_n \}$$

Donde a_i representa la pregunta i . De manera general la probabilidad P surge de la siguiente representación:

$$P_j (\{a_1\} \cup \{a_2\} \cup \{a_3\} \cup \{a_4\} \cup \{a_5\} \cup \dots \cup \{a_n\}).$$

Hasta este momento todos las preguntas tienen el mismo peso y son diferentes, es decir:

$$P_j (\{a_1\}) + P_j (\{a_2\}) + P_j (\{a_3\}) + P_j (\{a_4\}) + P_j (\{a_5\}) + \dots + P_j (\{a_{10}\})$$

Dándole un valor de $1/n$ a las respuestas afirmativas y 0 a las respuestas negativas, tenemos que:

$$P(\{a_j\}) = \{1/n \text{ si es afirmativa } \text{ ó } 0 \text{ si no lo es } \}$$

Donde n es el número total de eventos.

Dado $E = \{ a_1, a_2, a_3, \dots, a_n \}$, el resultado de la evaluación estaría en los siguientes rangos:

Si P(E)	Rango	Clasificación
{	0 ... 50	Malo
	51 .. 69	Regular
	70 ... 80	Bien
	81 ... 90	Muy Bien
	91 ... 100	Excelente

Se ha considerado este rango de evaluación, dado el conocimiento empírico aplicado en las calificaciones escolares y otras que se usan de manera cotidiana.

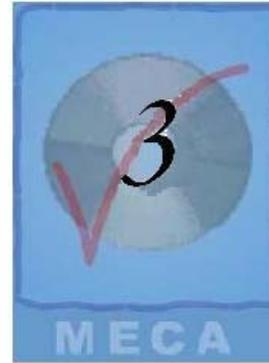
Por ejemplo, para obtener la calificación definitiva en la evaluación se multiplica 0.1×100 que representa el 100% de las preguntas, se tiene que el resultado es 10, ($0.1 \times 100 = 10$) donde la calificación 10 indica que sólo una pregunta se ha contestado afirmativamente; por otro lado, si se tiene el siguiente resultado:

$$1/10 + 1/10 + 1/10 + 1/10 + 0/10 + 0/10 + 1/10 + 1/10 + 1/10 + 1/10 = 0.8$$

Lo cual, se multiplica $0.8 \times 100 = 80$, que representa la calificación final en la encuesta y así sucesivamente.

Es menester añadir que la importancia de la probabilidad no toma su mayor peso en la aplicación de este modelo matemático, sino en el conjunto de respuestas que se obtendrán de la evaluación de todos los usuarios de la Calidad en Uso o en su defecto, la muestra elegida por el *Administrador*, donde se observa el peso de la decisión del cliente final, ya que si existen métricas que determinan el rumbo de una pregunta y ese rumbo no es el óptimo, existe la posibilidad de corregir el problema aplicando la reingeniería del producto.

Análisis y diseño del Sistema MECA



En el Apéndice D se encuentran las características del dominio de información del Sistema MECA con las actividades que se realizan para obtener el grado de complejidad de la herramienta mediante los cálculos de los Puntos de Función.

A partir de los valores obtenidos mediante el cálculo de los Puntos de Función se puede conseguir, a través de algún modelo paramétrico, las cantidades estimadas de indicadores como el esfuerzo, tiempo de desarrollo, la cantidad de personal, la productividad, el costo. Estos servirán para la planeación y el control del proyecto. En el Apéndice E aparece un estudio realizado utilizando el COCOMO [36].

3.1 Modelado de la arquitectura del Sistema MECA

Para explicar detalladamente los requerimientos y diseñar los objetivos del sistema, es útil mostrar una perspectiva gráfica general usando los elementos del Diseño Orientado a Objetos. Para ello se ha utilizado UML (Unified Modeling Language, Lenguaje Unificado de Modelado) creado por Grady Booch, James Rumbaugh e Ivar Jacobson [6]. El UML es un lenguaje estándar para escribir planos de software. Es independiente del proceso, aunque para utilizarlo óptimamente se debe usar en un proceso dirigido por los casos de uso.

En el contexto del software, existen cinco vistas complementarias que son las más importantes para visualizar, especificar, construir y documentar la arquitectura de un sistema:

- casos de uso,
- diseño,
- procesos,
- implementación y
- la vista de despliegue.

Cada una de estas vistas involucra un modelo estructural (modelo de elementos estáticos), así como un modelo de comportamiento (modelo de elementos dinámicos). Estas vistas capturan las decisiones más importantes acerca del Sistema MECA. De manera individual, cada una de estas vistas permite centrar la

Eliminado: Este capítulo se encuentra dividido en dos partes importantes: en la primera se presenta el Análisis y diseño del Sistema MECA, se identifican las características del dominio de información, que está integrada por la contabilidad de las pantallas de entradas y salidas (véase **Anexo D**), la cantidad de componentes y archivos que interactúan en el proceso, y el número de archivos de Base de Datos. Se presenta la fórmula para calcular los puntos de función basándose de la complejidad de la información administrada. ¶
En la segunda parte se encuentra el modelado de la arquitectura del sistema, integrada por tres vistas: los casos de uso, la vista de diseño y la Base de Datos cada una con su detalles particulares, estos detalles se pueden apreciar en el grafo conversacional presentado en el **Anexo C**. ¶

atención en una perspectiva del sistema para poder razonar con claridad sobre las decisiones.

UML define nueve tipos de diagramas que pueden mezclarse y conectarse para componer la parte Estática y Dinámica. Para los sistemas Cliente / Servidor, como lo es *MECA*; la parte Estática la forman:

- Diagrama de clases.
- Diagrama de objetos.
- Diagrama de componentes.
- Diagrama de despliegue.

La parte dinámica o Diagramas de comportamiento son:

- Diagrama de casos de uso.
- Diagrama de secuencia.

Dado que los objetos pueden representar instancias de otros elementos como las colaboraciones, componentes y nodos, también se diseñan para el Sistema *MECA*, los siguientes diagramas:

- Diagrama de colaboración.
- Diagrama de estados.
- Diagrama de actividades.

3.1.1 La vista de los casos de uso

Implica modelar el contexto del sistema o la forma de comportamiento de sus elementos; esto redundando en el diseño de los diagramas de casos de uso.

3.1.1.1 Diagramas de casos de uso

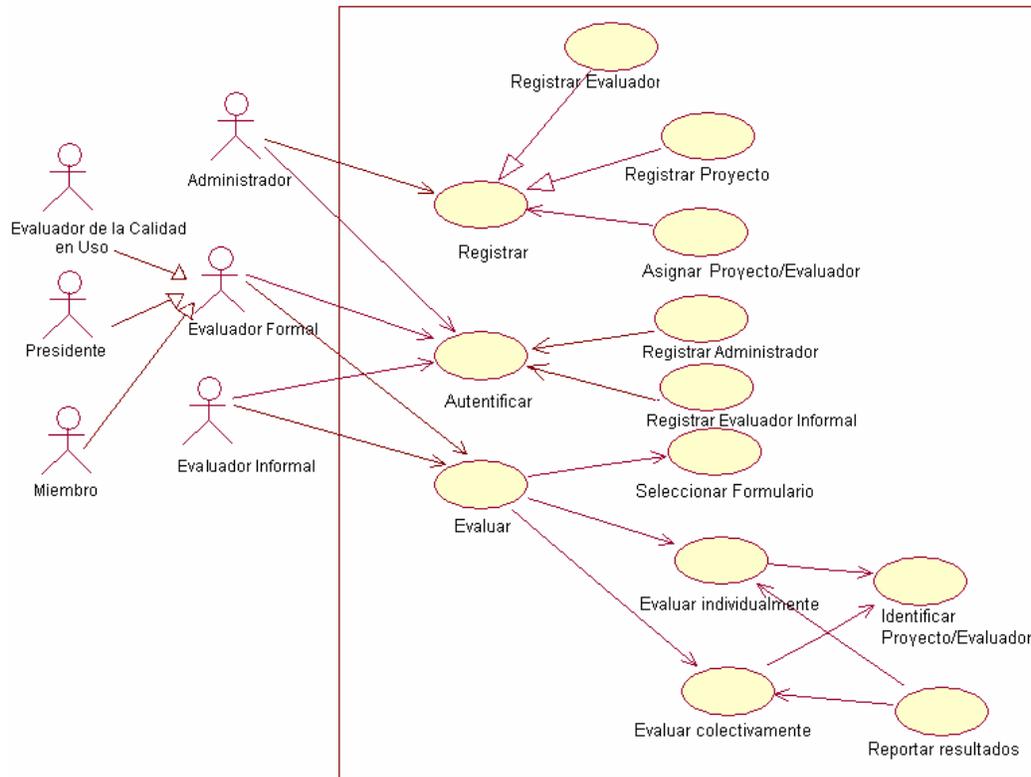
Los casos de uso se utilizan para estructurar los aspectos de comportamiento en un modelo.

Contienen:

- *casos de uso*,
- *actores* y
- *relaciones de dependencia, generalización y asociación*.

En la Figura 3.1 se muestra el diagrama de Casos de Uso General. En el sistema intervienen como actores: el *Administrador*, el *Usuario Informal*, el *Usuario Formal* y el *Usuario de la Calidad en Uso*; se puede ver la relación entre estos actores y los casos de uso (procesos).

Figura 3.1 Caso de Uso general



El Usuario o persona que accede al sistema se clasifica en los siguientes tipos:

- *El Administrador:* persona encargada del aseguramiento de la calidad del software en la empresa o institución productora de software. Sus funciones dentro de *MECA* son:
 - Registrar evaluadores, empresas y proyectos.
 - Proporcionar claves de acceso a los evaluadores (formal) de su empresa o institución.
- *El Evaluador Formal:* persona designada oficial o formalmente por una institución o empresa para evaluar los productos de software elaborados en ella. Accede al sistema con la clave que le proporciona su Administrador. Estos evaluadores pueden ser:
 - *El Evaluador de la Calidad en Uso:* persona que evalúa la Calidad en Uso de un sistema. Puede llamarse también el usuario final del software. Al igual que los otros evaluadores, el Administrador lo registra y proporciona una clave de acceso. La función del evaluador es:
 - Dar respuesta a la encuesta a la Calidad en Uso y emitir su opinión.

-
- *Presidente*: es el evaluador que pide al sistema el promedio de la evaluación del comité; sus funciones son:
 - Buscar el proyecto a evaluar y organizar al comité en esta actividad.
 - Preparar la documentación del proyecto a evaluar.
 - Evaluar el software (todas las actividades del evaluador miembro).
 - Solicitar al sistema el promedio de la evaluación colectiva.
 - *Miembro*: Es el Evaluador que da respuesta al formato de evaluación y conoce el resultado de la etapa evaluada. No podrá acceder al sistema sin su clave. Sus funciones son:
 - Acordar con el presidente del comité sobre el proyecto a evaluar.
 - Buscar el proyecto a evaluar.
 - Preparar la documentación del proyecto.
 - Dar respuesta al formato de evaluación individual.
 - Observar el resultado de la etapa evaluada.
 - *El Evaluador Informal*: Persona que no necesita una designación oficial, ni de una empresa para evaluar la calidad de un software. Ingresa al sistema, evalúa su producto e imprime el resultado de la etapa evaluada. Sus funciones son:
 - Acceder al sistema sin clave.
 - Registrar sus datos y los de su proyecto.
 - Preparar la documentación del software a evaluar.
 - Al registrar sus datos, el sistema le da una clave.
 - Acceder al sistema con su clave.
 - Elegir la etapa a evaluar.
 - Iniciar la evaluación.

En el caso de uso *Evaluar Individualmente*, se realiza el siguiente flujo de eventos:

1. El Usuario ingresa a la etapa elegida para iniciar una evaluación.
2. El sistema le muestra el conjunto de características, subcaracterísticas y atributos que integran la evaluación de la etapa.
3. El Usuario proporciona en forma consecutiva los valores de cada una de las métricas que le presenta el sistema.
4. Al ingresar el *Evaluador* todos los valores de los atributos, el sistema le muestra los resultados de la etapa.
5. El evaluador toma nota de ese resultado y/o lo imprime.

En el caso de uso *Evaluar colectivamente* se recomienda que el comité esté integrado por un Presidente y los miembros (de tres a cuatro integrantes). En la evaluación colectiva solo el presidente podrá imprimir el promedio de la evaluación. Se realiza el siguiente flujo de eventos:

1. El Usuario ingresa a la etapa elegida para iniciar una evaluación.
-

2. El sistema le muestra el conjunto de características, subcaracterísticas y atributos que integran la evaluación de la etapa.
3. El Usuario proporciona en forma consecutiva los valores de cada una de las métricas que le presenta el sistema.
4. Al ingresar el Usuario todos los valores de los atributos, el sistema le muestra los resultados de la etapa.
5. El sistema identifica al Presidente del comité y le ofrece la posibilidad de observar e imprimir el promedio de la evaluación.
6. El Usuario observa el promedio y/o lo imprime por impresora.

El caso de uso *Autenticar* se desarrolla el siguiente flujo de eventos:

1. Un usuario desea ingresar al sistema.
2. De ser un Evaluador Formal o de la Calidad en Uso, Ingresa su clave de acceso.
3. Si el Evaluador no tiene clave de acceso, teclea *Intro* luego el sistema lo identifica como un Evaluador Informal, por lo que solo le pedirá los datos del Evaluador y del proyecto; lo registra y le da una clave.
4. De ser un Administrador ingresará con la clave correspondiente.

El caso de uso *Registrar* tiene el propósito de Ingresar al sistema los datos de los proyectos y evaluadores.

El caso de uso *Registrar Evaluador* se realiza cuando el Administrador cuenta con los datos de un Evaluador a registrar en el sistema. Posteriormente se realiza el siguiente flujo de eventos:

1. El Administrador prepara los datos de un Evaluador.
2. Proporciona los datos personales del Evaluador en el orden en que el sistema se lo solicite.
3. El sistema guarda en sus archivos los datos introducidos y le sugiere imprimirlos por escrito para comprobar la corrección de estos.

El caso de uso *Registrar Proyecto* se realiza cuando el Administrador debe tener los datos de un Proyecto a registrar en el sistema. Posteriormente se realiza el siguiente flujo de eventos:

1. El *Administrador* prepara los datos del proyecto.
2. Proporciona los datos del *Evaluador* en el orden en que el sistema se lo solicite.
3. El sistema guarda en sus archivos los datos introducidos y le sugiere imprimirlos para comprobar su corrección.

El caso de uso *Asignar Evaluador/Proyecto* se realiza cuando el *Administrador* ha registrado los datos de un *Evaluador* y *Proyecto* en el sistema. Posteriormente se realiza el siguiente flujo de eventos:

1. El *Administrador* elige al *Evaluador* y *Proyecto* de la lista mostrada por el sistema.
 2. El sistema guarda en sus archivos los datos elegidos.
-

3.1.1.2 Diagrama de secuencia

El diagrama de secuencia es un diagrama de interacción que muestra el ordenamiento temporal de los mensajes. Sirve para describir el flujo de eventos de un caso de uso, identificando los objetos que participan en el caso de uso y asignando partes del comportamiento del caso de uso a los objetos, en forma de servicios [4].

Un diagrama de interacción contiene:

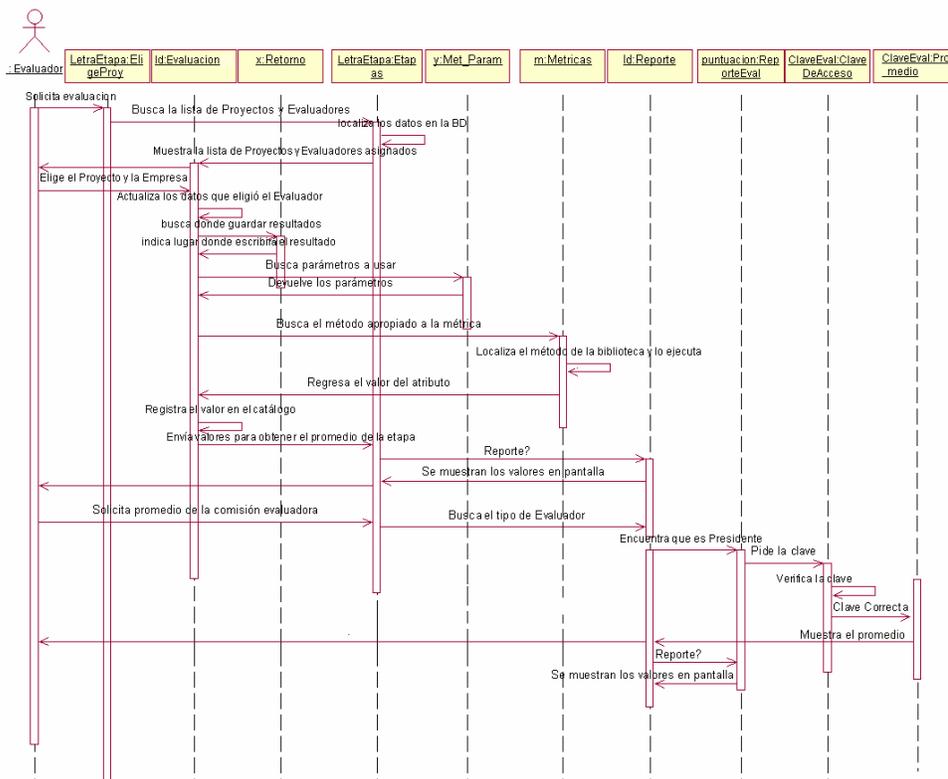
- Objetos.
- Enlaces.
- Mensajes.

En la Tabla 3.1 se presenta el diagrama de secuencia de alto nivel de la evaluación. Los diagramas de registro y asignación de proyectos a evaluadores se encuentran en el Apéndice A. En la Tabla 3.6 se encuentran los eventos [31] que desencadena el diagrama de secuencias.

Tabla 3.1 Eventos que desencadena la Evaluación.

Acción del actor	Acción del sistema
1.- El diagrama comienza cuando el Evaluador tiene un proyecto a evaluar.	2.- Busca la lista de proyectos y evaluadores asignados por el Administrador.
	3.- Presenta la lista
4.- Elige de la lista el proyecto a evaluar	5.- Registra la elección
	6- Presenta el formato de evaluación
7.- Introduce los parámetros que pide el sistema	8.- Identifica el atributo que recibirá la operación de los parámetros dados por el evaluador
	9.- Toma los parámetros e identifica su valor
	10.- Busca en la biblioteca de métodos la métrica que corresponde al atributo a evaluar
	11.- Ejecuta la métrica correspondiente
	12.- Almacena en el atributo el valor que devuelve la métrica
	13.- Si hay mas parámetros Regresa al paso 7 sino ve al paso 14
14.- Solicita un reporte de evaluación	15.- Localiza la clase que realiza el reporte.
	16.- Muestra los resultados de la evaluación en pantalla
17.- Observa los resultados	
18.- Solicita el promedio de la comisión evaluadora	19.- Localiza el tipo de Evaluador que pide el promedio
	20.- Si es Presidente del proyecto ve al paso 21 sino al paso 29
	21.- Pide la clave del Presidente
	22.- Autentifica la clave
	23.- Si es la clave correcta ve al paso 24 sino al paso 29
	24.- Muestra el promedio del comité
25.- Observa el promedio	
26.- Solicita el informe final de evaluación	27.- Muestra el informe final en pantalla
28.- Obtiene la información y pasa a 30	29.- Muestra un mensaje de error y pasa a 26
30.- Se marcha	

Figura 3.2 Diagrama de secuencia de la Evaluación individual.



3.1.2 Vista de diseño

Esta vista comprende las clases, interfaces y colaboraciones que interactúan en la solución del problema. Se forma con los diagramas de colaboración, estados, actividades, clases y objetos.

3.1.2.1 Diagrama de colaboración

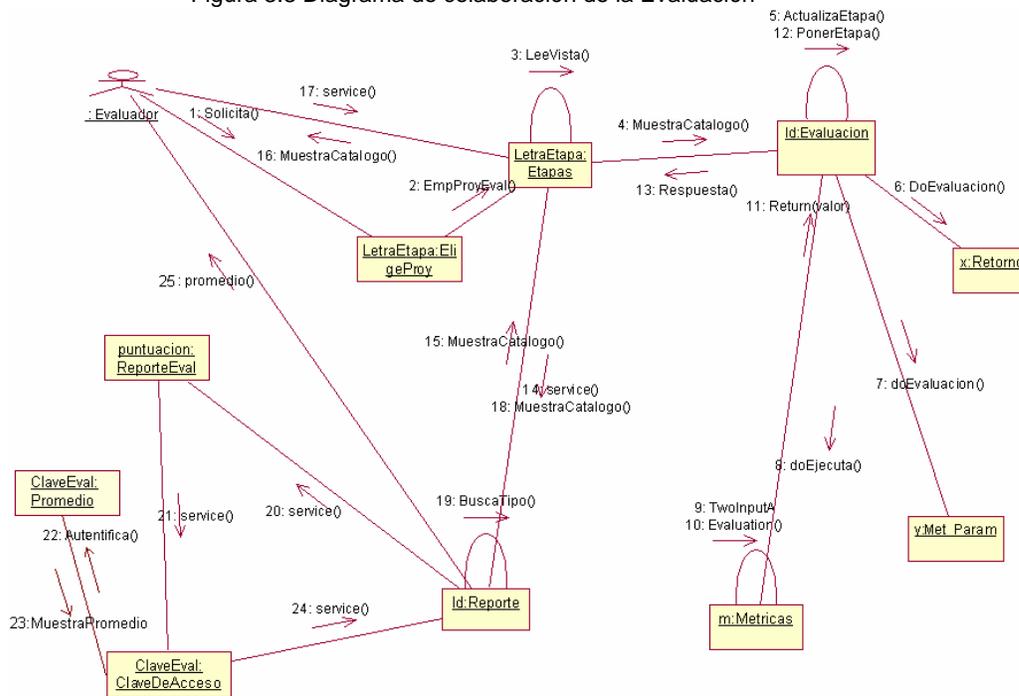
En el diagrama de colaboración destaca la organización de los objetos que participan en cada interacción.

En la Figura 3.3 se muestra el diagrama de colaboración de la Evaluación. Se observa la interacción de las clases que desarrollan la secuencia descrita anteriormente.

1. El Evaluador pide el servicio, mediante la clase *solicita*, esta clase se comunica con la clase *EligeProy*.

-
2. La clase *EligeProy* toma parámetros como es la clave del evaluador, la clave de la empresa y del proyecto, a su vez registra la etapa a evaluar comunicándose con la clase *Etapa*.
 3. La clase *Etapa* recibe estos parámetros y muestra el catálogo de la etapa elegida.
 4. El evaluador ingresa los datos de las métricas, lo cual llama a la clase *Evaluación*.
 5. La clase *Evaluacion* pide a *Retorno* la clave del atributo al cual pertenece la métrica.
 6. La clase *Retorno* busca la clave del atributo y la envía a *Evaluacion*.
 7. *Evaluacion* pide a la clase *Met_Param* los parámetros que se usarán.
 8. La clase *Met_Param* envía los parámetros a *Evaluacion* para solicitar posteriormente a la clase *Metricas* el método correspondiente. La clase *Metricas* contiene los siguientes métodos:
 9. *TwoInputA*, que se ejecuta para determinar el valor del atributo.
 10. *Evaluation()*, ubica este valor en un rango de 0 a 3.
 11. El método de la clase *Metricas* devuelve a *doEvaluacion* el resultado de su ejecución.
 12. La clase *Evaluacion* escribe este valor en el lugar del atributo correspondiente.
 13. Muestra los resultados a la clase *Etapas*.
 14. *Etapas* envía los resultados a *Reporte*.
 15. La clase *Reporte* muestra el reporte con el promedio de la etapa elegida y se comunica con *Etapas*.
 16. La clase *Etapas* recibe la petición de impresión.
 17. Se registra la petición.
 18. *Etapas* envía a *Reporte* el catalogo con los resultados.
 19. La clase *Reporte* busca el tipo del Evaluador que solicita el promedio.
 20. Pide a la clase *ReporteEval* el informe de la evaluación.
 21. La clase *Reporte* pide la autenticación de quien pide el promedio a la clase *ClaveDeAcceso*.
 22. Esta clase verifica la clave correcta y pide el promedio a la clase *Promedio*.
 23. Esta clase muestra el promedio
 24. Devuelve el resultado de la autenticación a *Reporte*.
 25. Muestra al Evaluador el promedio del comité.
-

Figura 3.3 Diagrama de colaboración de la Evaluación



Los diagramas de registro y asignación se encuentran en el Anexo G.

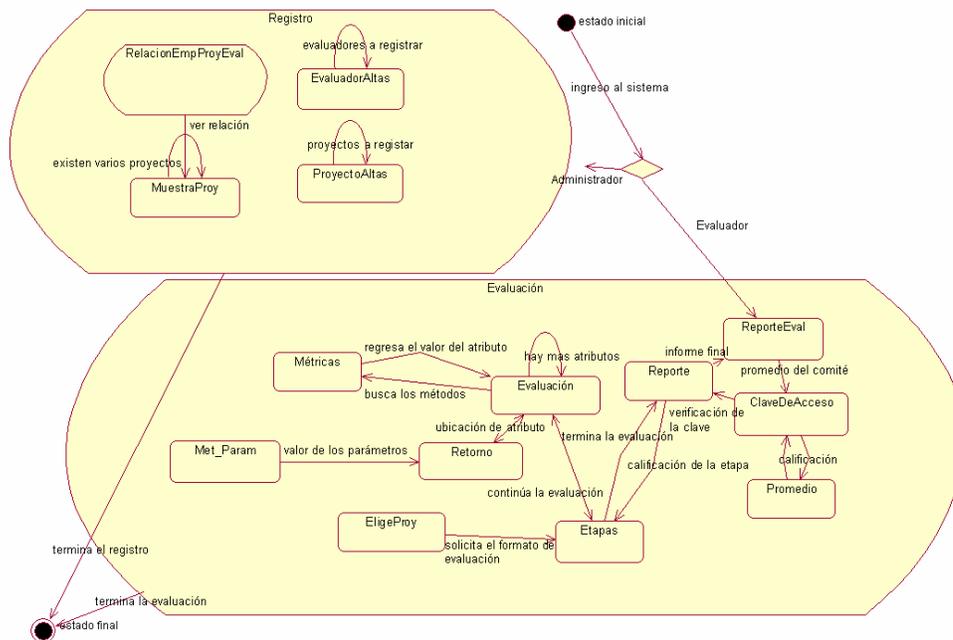
3.1.2.2 Diagrama de estados

El diagrama de estados se usa para modelar los aspectos dinámicos del sistema. Sirve para medir el comportamiento de un objeto y especifica la secuencia de estados por los que pasa a lo largo de su vida en respuesta a eventos, junto con sus respuestas a esos eventos.

El diagrama de estados se puede asociar a las clases, casos de uso o sistemas completos para visualizar, especificar, construir y documentar la dinámica de un objeto individual.

En la Figura 3.4 se muestra el diagrama de estado del Sistema MECA. Las acciones comienzan con un estado inicial, en el que existe una transición representada por una flecha; el rombo representa una condición; en este caso se pregunta por el tipo de usuario que ha desencadenado el evento. Los estados son representados por un rectángulo.

Figura 3.4 Diagrama de estados del Sistema MECA



Un estado espera un evento. Para el evento *Administrador* se activa el *Registro*. Este estado contiene el subestado *EvaluadorAltas* que puede ser el primero en ejecutarse en *Registro*.

El arco indica una ejecución iterada (por ejemplo, se registran dos o más evaluadores). Lo mismo sucede con el estado *ProyectoAltas*.

MuestraProy y *RelacionEmpProyEval* son subestados de *Registro* y se iteran tantas veces como el número de evaluadores y proyectos tenga el comité. El estado *RelacionEmpProyEval* es dependiente de *MuestraProy* porque muestra la relación de evaluadores en base a los datos asignados en *MuestraProy*.

El evento *Evaluador*, activa al estado *Evaluación*. Este estado pone en marcha al subestado *EligeProy*.

EligeProy identifica el proyecto de la empresa a evaluar. Al iniciarse la acción de solicitud del formato de evaluación se pasa al subestado *Etapas*.

Etapas muestra el catálogo de la etapa y pasa al estado *Evaluación* que realiza los cálculos de la evaluación; proceso en la cual, intervienen *Retorno*, *Met_Param* y *Métricas*. Estos subestados devuelven a *Evaluación* el resultado de la etapa.

Etapas activa a *Reporte* para la impresión del resultado. Este estado pasa a *ReporteEval* que muestra el informe final.

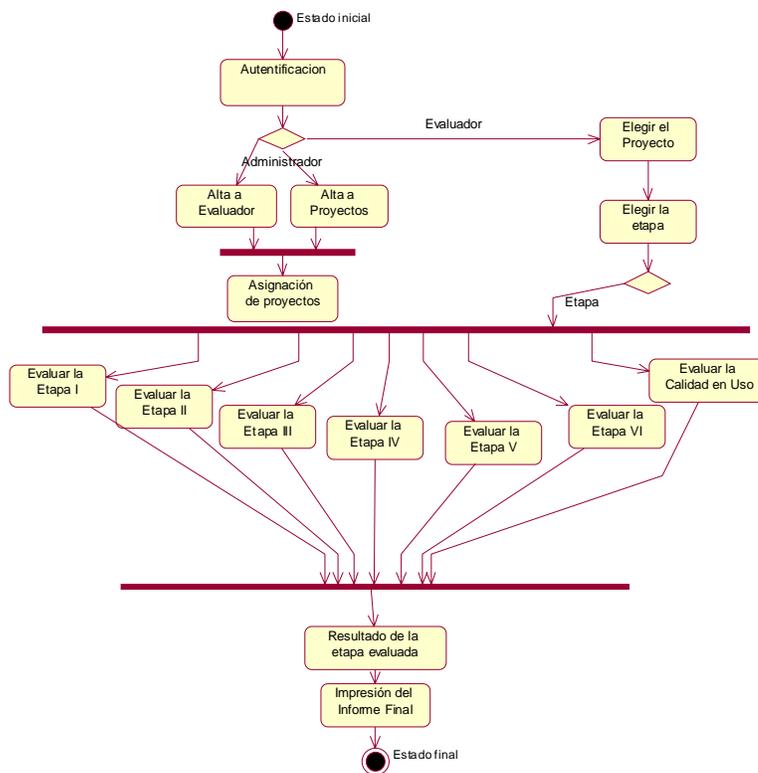
Promedio recibe la petición del promedio, por lo que su estado cambia a *ClaveDeAcceso*. Este evento autentifica la clave del Evaluador; en base al resultado pasa al estado *Promedio* o regresa a *Reporte*.

3.1.2.3 Diagrama de actividades

Un diagrama de actividades es fundamentalmente un diagrama de flujo que muestra el flujo de control entre las actividades. Se utilizan para visualizar, especificar, construir y documentar la dinámica de un conjunto de objetos.

En la Figura 3.5 se encuentran las actividades que se realizan en el Sistema MECA. En ella no se especifican todas las acciones porque una actividad puede tener varios eventos. Pero se muestra las siete etapas a evaluar, estas constituyen las actividades de la evaluación. Aquí se usa la misma simbología que el diagrama de estado, excepto la barra horizontal que representa una unión o división concurrente según sea el caso. Las acciones incluyen llamadas a otras operaciones, envío de señales, creación o destrucción de objetos o como en este caso, simples cálculos como la evaluación de una expresión.

Figura 3.5 Diagrama de actividades del Sistema MECA



3.1.2.4 Diagrama de clases

Los diagramas de clases, se conocen como diagramas estructurales, son los más utilizados en el diseño de los Sistemas Orientados a Objetos, éstos muestran un conjunto de clases, interfaces y colaboraciones, así como sus relaciones.

En la Figura 3.9 se muestra el diagrama de clases del Sistema MECA. La clase *BorrarEleccion* tiene como función modificar y/o borrar las asignaciones erróneas que obtiene de *MuestraProy* por lo que se encuentran asociadas.

La lista de los evaluadores y proyectos que ofrece *MuestraProy* los obtiene de *EvaluadorAltas* y *ProyectoAltas* por lo cual es dependiente.

La clase *EvaluadorAltas* almacena en la Base de Datos MECA los datos del *evaluador*, lo mismo que *ProyectoAltas* con respecto a los proyectos, esto justifica la relación de asociación entre ellas.

La clase *EligeProy* es dependiente de *MuestraProy* porque accede la lista que esta clase le proporciona.

Evaluacion realiza la evaluación, y en ese proceso es dependiente de *Etapas* porque requiere de la identificación de la etapa elegida.

Etapas depende de *EligeProy* del cual recibe las claves de evaluador y proyecto.

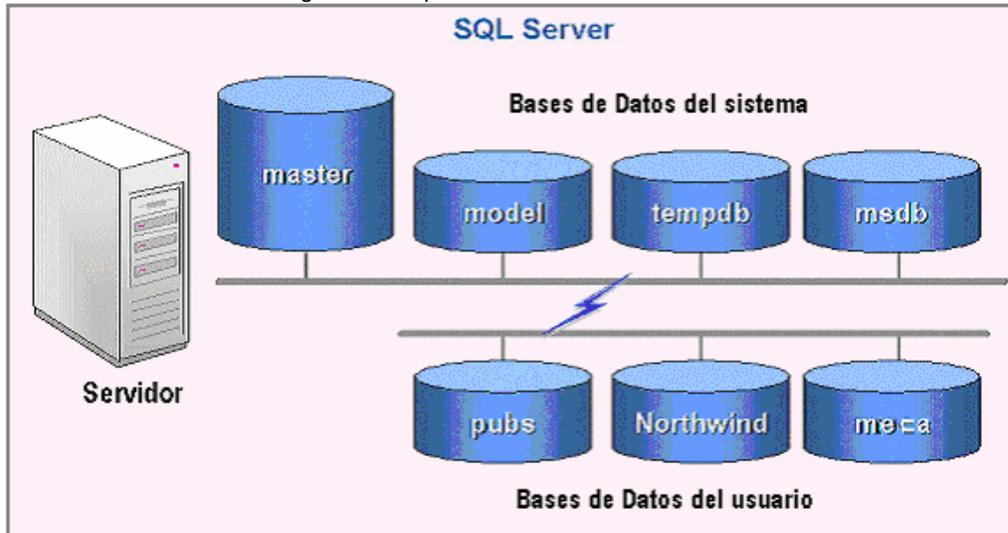
La clase *Evaluación* está asociada y es dependiente de la clase *Métricas* porque ésta tiene los métodos, con las fórmulas, que se aplicarán en la evaluación de los atributos, subcaracterísticas y características de la etapa.

La clase *Met_Param* hereda de *Retorno* porque al procesar el resultado de la fórmula aplicada al atributo, se comunica con *Retorno* que tiene conocimiento del lugar exacto donde será guardado este atributo de la Base de Datos MECA.

La clase *Evaluacion* también es dependiente y asociada a las clases *Retorno* y *Met_Param* que obtiene y ubica los valores de la evaluación en la Base de Datos MECA.

La clase *Etapas* recibe la petición de impresión por lo que está asociada a la clase *Reporte* que depende del formato del informe de evaluación de la clase *ReporteEval*.

Figura 3.7 Arquitectura de la Base de Datos



Durante la operación de la Base de Datos (consultas, actualizaciones, eliminaciones e inserciones) es útil el lenguaje SQL (Structured Query Language, Lenguaje de Consultas Estructurado).

Se ha utilizado el manejador de Base de Datos de SQL Server en la implementación del Sistema MECA porque es capaz de soportar millones de registros por tabla (2.2 terabytes en una sola Base de Datos), y ha admitido hasta 30,000 usuarios simultáneos [12]. Otra de las características del manejador es que permite el procesamiento de consultas anidadas y procedimientos almacenados utilizados en algunas búsquedas de información del Sistema MECA. SQL Server proporciona la escalabilidad y confiabilidad necesarias, y se ejecuta en hardware accesible y manejable. Aunque el tiempo promedio de búsqueda de información depende de la velocidad del procesador y la memoria; el manejador cuenta con un complejo y mejorado sistema de transacciones capaz de realizar más de un millón de transacciones en dos minutos [12].

3.2.1 Diseño de la Base de Datos

El Modelo Entidad-Relación parte de considerar la existencia de entidades, que representan objetos, personas, etc, sobre las que se quiere almacenar información. Las entidades con las mismas características forman un tipo de entidad. A las características necesarias para describir completamente cada tipo de entidad se les denomina *atributo*.

Posteriormente, las entidades y sus atributos se representan físicamente a través de tablas (transformación en un *Modelo Relacional*) en las que los datos se almacenan en dos dimensiones. Las filas de la tabla contienen los atributos, y las

columnas el conjunto de atributos del mismo tipo de cada entidad. El grado de la tabla corresponderá al número de columnas.

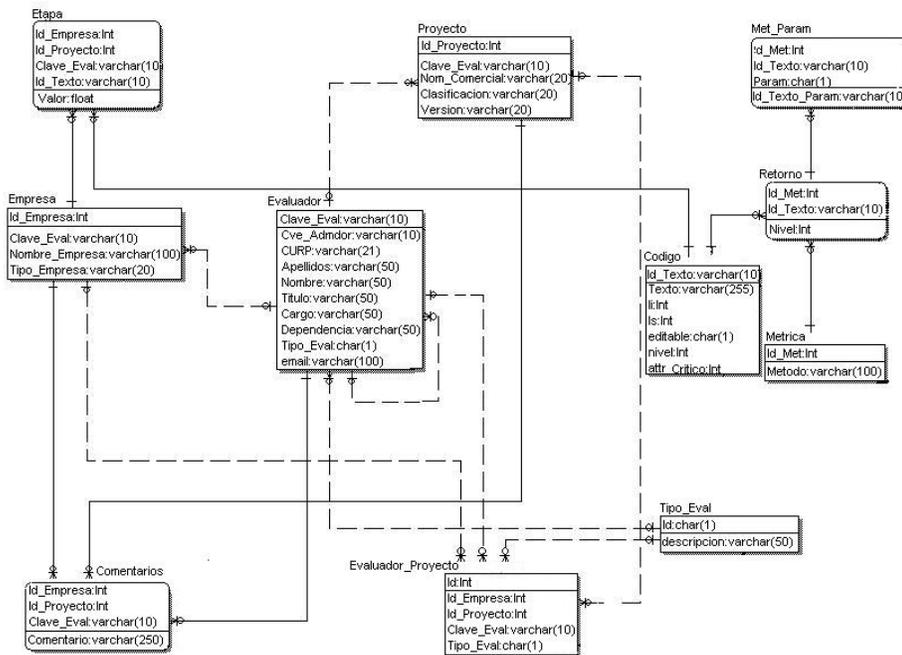
A continuación, se muestran el diagrama conceptual. Lo realmente interesante de una Base de Datos y lo que determina en gran medida su funcionalidad es su esquema conceptual.

El Modelo que se presenta en la Figura 3.18 muestra el diagrama Lógico Entidad-Relación diseñado con la herramienta CASE ER-Win [41]; la Base de Datos se encuentra normalizada ya que se ha procurado la correcta distribución de la información entre las tablas, se ha definido una *llave primaria* en todas ellas y en algunos casos *llaves foráneas*.

Para una mejor interacción de la información; se ha asegurado agregar *integridad referencial* para reforzar la relación. Esto evita que los campos vinculados se modifiquen incorrectamente y que se dejen registros “*huérfanos*”. El motor de base de datos de SQL Server 2000 admite *integridad referencial*, que permite disponer de actualizaciones y eliminaciones en cascada.

La normalización de la Base de Datos consiste en seleccionar la información en componentes con la finalidad de evitar la repetición innecesaria de datos. Normalizar la información es un proceso complejo que bien logrado brinda beneficios importantes al desempeño de consultas de datos y al aprovechamiento del espacio disponible en disco.

Figura 3.18 Modelo Lógico Entidad-Relación



La Tabla 3.2 muestra el diccionario de datos que conforma el modelo Entidad-Relación.

Tabla 3.2 Descripción de las tablas de la Base de Datos MECA

Entidad	Descripción
Codigo	Almacena el nombre de las etapas, características, subcaracterísticas, atributos y métricas del modelo MECA.
Comentarios	Contiene las opiniones de los evaluadores de la Calidad en Uso.
Empresa	Guarda los datos particulares de la empresa cuyo software está siendo evaluado.
Etapas	Almacena los datos de la evaluación, tales como la clave de la empresa, proyecto, evaluador, así como la clave de la característica, subcaracterística y atributo, y la calificación de estos.
Evaluador	Contiene los datos particulares del evaluador y el administrador.
Evaluador_Proyecto	Es una entidad conector entre las entidades Evaluador y Proyecto, la cual establece la relación de que un evaluador puede tener varios proyectos asignados y un proyecto podrá ser valorado por varios evaluadores.
Métrica	Guarda la clave y el nombre del método que será aplicado en la métrica para obtener el valor del atributo.
Met_Param	Almacena la clave y el valor de las métricas evaluadas.
Proyecto	Contiene los datos particulares del proyecto evaluado.
Retorno	Guarda la clave de los métodos, la de las características, subcaracterísticas, atributos y el nivel de ubicación de estos en el formato de la evaluación.
Tipo_Eval	Contiene el catálogo de los tipos de evaluadores.

Una vez realizado el modelo lógico, se procede a la creación de las tablas, llaves, índices, vistas e integridad referencial que integran el modelo físico de la Base de Datos del Sistema MECA. Éste modelo describe la manera en que serán creadas las tablas en la arquitectura de un manejador de Bases de Datos como lo es el SQLServer 2000 utilizado en MECA.

A continuación se muestran las características de cada tabla, sus atributos, tipos de datos, etc.

Codigo.

Cada uno de los nombres de los elementos de evaluación se encuentran en esta tabla.

Codigo
Id_Texto: varchar(10)
Texto: varchar(255)
li: int
ls: int
editable: char(1)
nivel: int
attr_Critico: int

Tabla 3.3 Atributos de la tabla Codigo

Atributo	Tipo	Longitud	Tipo Campo	Descripción
Id_Texto	varchar	10	Llave primaria	Clave de los elementos de la evaluación.
Texto	varchar	255		Nombre de los elementos de la evaluación.
Li	int			Niveles de los elementos de la evaluación.
Ls	int			Niveles de los elementos de la evaluación.
Editable	char	1		Indicación de un campo que espera un valor.
nivel	int			Nivel de los elementos de la evaluación
Attr_Critico	int			Agrupación de los atributos comunes

Comentarios.

Las opiniones escritas por los evaluadores de la Calidad en Uso se encuentran en esta tabla.

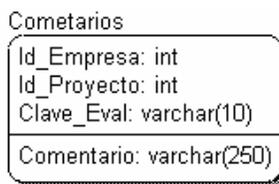


Tabla 3.4 Atributos de la tabla Comentario

Atributo	Tipo	Longitud	Tipo Campo	Descripción
Id_Empresa	int		Llave primaria/foránea	Clave de la empresa.
Id_Proyecto	int		Llave primaria/foránea	Clave del proyecto.
Clave_Eval	varchar	10	Llave primaria/foránea	Clave del evaluador.
Comentario	varchar	250		Texto acerca de la opinión del evaluador de la Calidad en Uso.

Empresa.

Los datos particulares de la empresa evaluada se encuentran almacenados en esta tabla.

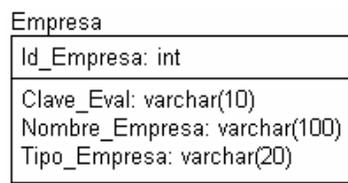


Tabla 3.5 Atributos de la tabla Empresa

Atributo	Tipo	Longitud	Tipo Campo	Descripción
Id_Empresa	int		Llave primaria	Clave de la empresa
Clave_Eval	varchar	10	Llave foránea	Clave del Evaluador
Nombre_Empresa	varchar	100		Nombre de la empresa
Tipo_Empresa	varchar	20		Tipo de empresa (giro organizacional)

Etapa.

Lugar donde se almacenan los datos de las evaluaciones. Al elegir el proyecto de la empresa, el sistema MECA localiza en esta tabla si existe una evaluación previa, de ser así muestra el formato en pantalla, de lo contrario registra la nueva evaluación.



Tabla 3.6 Atributos de la tabla Etapa

Atributo	Tipo	Longitud	Tipo Campo	Descripción
Id_Empresa	Int		Llave primaria/foránea	Clave de la empresa
Id_Proyecto	Int		Llave primaria	Clave del proyecto
Clave_Eval	varchar	10	Llave primaria	Clave del evaluador
Id_Texto	varchar	10	Llave primaria/foránea	Clave del elemento de evaluación
Valor	float			Calificación del elemento evaluado

Evaluador.

Tabla que almacena los datos particulares del evaluador.

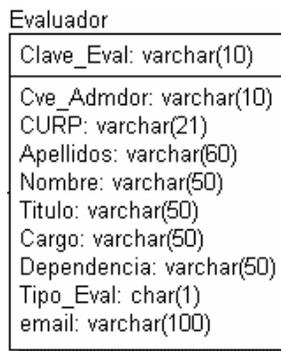


Tabla 3.7 Atributos de la tabla Evaluador

Atributo	Tipo	Longitud	Tipo Campo	Descripción
Clave_Eval	Varchar	10	Llave primaria	Clave del evaluador.
Cve_Admdor	Varchar	10	Llave foránea	Clave del administrador, para identificar a los evaluadores que ha registrado y los informales.
CURP	Varchar	21		Código Único de Registro de Población
Apellidos	Varchar	60		Apellidos del evaluador o administrador
Nombre	Varchar	50		Nombre del evaluador o administrador
Título	Varchar	50		Título o grado académico
Cargo	Varchar	50		Puesto que ocupa en la empresa o institución evaluada.
Dependencia	Varchar	50		Nombre de la empresa o institución

				donde labora el evaluador.
Tipo_Eval	Char	1		Tipo de evaluador, los administradores los identifica en la autenticación de la clave.
email	Varchar	100		Correo electrónico del evaluador

Evaluador_Proyecto.

Entidad que actúa como conector entre Evaluador y Proyecto, su propósito es identificar rápidamente el tipo de evaluador durante la asignación de proyectos a evaluadores de manera tal que no existan dos presidentes para el mismo proyecto o se nombre presidente al evaluador de la Calidad en Uso.

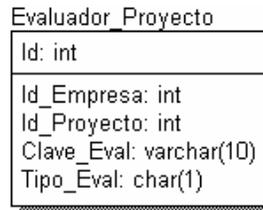


Tabla 3.8 Atributos de la tabla Evaluador_Proyecto

Atributo	Tipo	Longitud	Tipo Campo	Descripción
Id	Int	10	Llave primaria	Clave de la vista o entidad conector
Id_Empresa	Int	10	Llave foránea	Clave de la empresa
Id_Proyecto	Int	21	Llave foránea	Clave del proyecto
Clave_Eval	Varchar	60	Llave foránea	Clave del evaluador
Tipo_Eval	Char	50	Llave foránea	Tipo del evaluador

Métrica.

Entidad que almacena la clave y el nombre de los métodos aplicados a los elementos de evaluación.

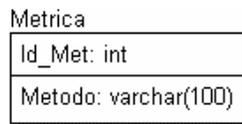


Tabla 3.9 Atributos de la tabla Metrica

Atributo	Tipo	Longitud	Tipo Campo	Descripción
Id_Met	Int		Llave primaria	Clave para identificar los métodos de evaluación
Metodo	Varchar	100		Nombre del método aplicado

Met_Param.

Entidad que aporta los componentes correctos en la aplicación de las fórmulas durante la evaluación. Almacena la clave del método a aplicar, la clave del elemento que antecede al evaluado, el nombre de la variable como parámetro y la clave del elemento evaluado.

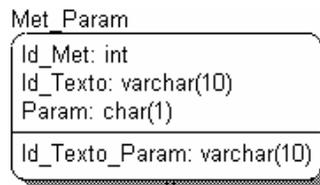


Tabla 3.10 Atributos de la tabla Met_Param

Atributo	Tipo	Longitud	Tipo Campo	Descripción
Id_Met	Int		Llave primaria/foránea	Clave de la métrica o método que contiene la fórmula de la métrica.
Id_Texto	varchar	10	Llave primaria/foránea	Clave de elemento antecesor del actualmente evaluado.
Param	char	1	Llave primaria	Nombre de la variable que funje como parámetro
Id_Texto_Param	varchar	10		Clave del elemento evaluado

Proyecto.

Entidad que almacena los datos de los proyectos a ser evaluados.

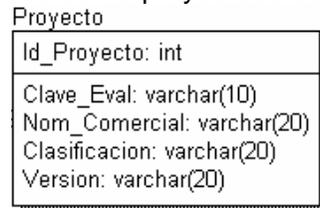


Tabla 3.11 Atributos de la tabla Proyecto

Atributo	Tipo	Longitud	Tipo Campo	Descripción
Id_Proyecto	int		Llave primaria	Clave del proyecto
Clave_Eval	varchar	10	Llave foránea	Clave del evaluador
Nom_Comercial	varchar	20		Nombre del proyecto que está siendo evaluado.
Clasificacion	varchar	20		Tipo de software.
Versión	varchar	20		Número de la versión del software.

Retorno.

Entidad que apoya al manejo correcto de los componentes de la evaluación.

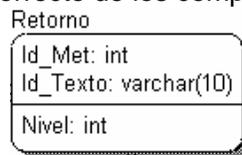


Tabla 3.12 Atributos de la tabla Retorno

Atributo	Tipo	Longitud	Tipo Campo	Descripción
Id_Met	int		Llave primaria/foránea	Clave de la métrica o método a usar en la evaluación.
Id_Texto	varchar	10	Llave primaria/foránea	Clave del elemento que está siendo evaluado.
Nivel	int			Nivel del elemento evaluado en el árbol jerárquico.

Tipo_Eval.

Entidad que contiene el catálogo de usuarios que utilizarán el sistema MECA.

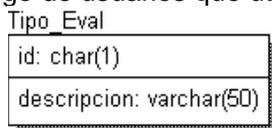


Tabla 3.13 Atributos de la tabla Tipo_Eval

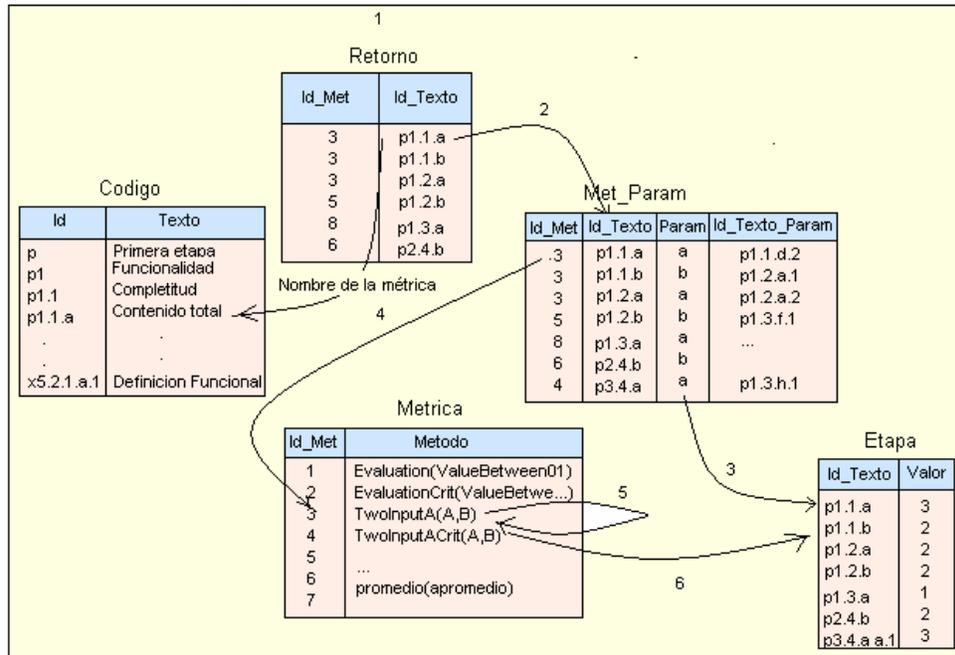
Atributo	Tipo	Longitud	Tipo Campo	Descripción
id	char	1	Llave primaria	Clave del tipo de usuario
descripcion	varchar	50		Nombre del tipo de usuario

En la Figura 3.9 se encuentra la comunicación entre las tablas de la Base de Datos MECA, la secuencia es:

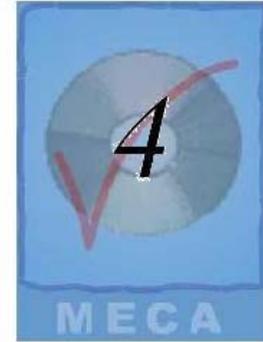
1. La tabla *Retorno* toma la primera clave del campo *Id_Texto* del formato de evaluación actual y lo localiza en la tabla *Met_Param*.
2. Esta clave en la entidad *Met_Param* permite obtener el parámetro a usar en la aplicación de la fórmula de evaluación.
3. Con la clave del paso 1 se localiza el valor del parámetro en la tabla *Etapa*.
4. En base al mismo paso 1, se obtiene de la tabla *Met_Param* la clave del método que contiene la métrica a aplicar en la evaluación (el nombre del método se encuentra en la tabla *Metrica*).
5. Se aplica el método correspondiente, en la tabla *Metrica* este proceso regresa un valor.
6. El valor que devuelve el método aplicado se guarda en la tabla *Etapa* en la posición del Atributo evaluado. Regresa al paso 1 hasta que todos los atributos, subcaracterísticas, características y etapa cuenten con una calificación.

En caso de necesitarse la descripción de la clave durante el proceso, se obtiene del paso 1 con la tabla *Codigo*.

Figura 3.9 Secuencia de ejecución de la Evaluación



Desarrollo, implementación y prueba del sistema



El Sistema MECA se ha desarrollado bajo los requerimientos de programación de la API (Application Programming Interface, Programación de Interface de Aplicaciones) de Java. Bajo estos requerimientos se implementan las clases en *Servlet* [37]. Una de las razones de la elección del lenguaje es que resulta ser independiente de la plataforma en que se encuentra implementado; es decir, el archivo objeto que se genera al compilar el código fuente tiene un formato independiente de la arquitectura de la máquina en que se ejecuta, de manera que cualquier máquina que tenga el sistema de ejecución (la máquina virtual de Java) puede ejecutar ese código objeto (esta característica hace transportable al Sistema MECA). Actualmente, existen máquinas virtuales en cada tipo de sistemas operativos.

Además, la API de Java y una gran cantidad de información acerca del lenguaje están disponibles en Internet de manera gratuita.

Durante el desarrollo de la herramienta se ha utilizado la versión 1.4.1 de *j2sdk* y como servidor de aplicaciones la versión 1.4 de Jakarta Tomcat. Este servidor es uno de los proyectos de código gratuito, de la *Apache Software Foundation*. Estos proyectos son definidos como *Aplicaciones Web* basados en Java creados para ejecutar *Servlets* y páginas JSP (Java Server Pages, Servidor de Páginas en Java). Este servidor de aplicaciones tiene la implementación oficial de referencia de las especificaciones *Servlet 2.3* y *JSP 1.2*.

En cuanto a la disponibilidad del equipo de cómputo, la instalación de la herramienta se ha realizado en una computadora personal con un sistema operativo Windows 2000 Profesional, con procesador Pentium III a 750 MHz y 320 Mb. de Memoria RAM, disco duro de 60 Gb. Se recomienda que el equipo de cómputo usado como servidor tenga mejores características de hardware, ya que la velocidad de atención a las evaluaciones está en función del procesador, el tamaño de memoria RAM y el ancho de banda disponible en la red; es decir, para que haya un buen servicio se requiere contar con la actualización de estos elementos.

Los sistemas operativos que facilitan la implementación del Sistema MECA como servidor de evaluaciones son: Windows 2000 Server, Advanced Server, XP Professional y NT 4.0. El sistema operativo utilizado en la instalación permite que

hasta 10 computadoras compartan simultáneamente una conexión única a Internet. Esta conexión puede ser vía acceso telefónico o por banda ancha; característica que se puede aprovechar en la evaluación simultánea de software de pequeñas empresas diseñadoras que cuentan con una sola conexión a Internet.

4.1 Aplicaciones Web

El Sistema MECA es una *Aplicación Web*. Estas aplicaciones fueron introducidas en la versión 2.2 de la especificación Servlet. De acuerdo con esta especificación, una *Aplicación Web* es una colección de Servlet, páginas JSP, clases Java, documentos estáticos: HTML (HyperText Markup Language, Lenguaje de Marcado para Hipertextos), XHTML (Extensible HyperText Markup Language, Lenguaje Extensible de Marcado para Hipertextos), imágenes, etc. y otros recursos, que pueden ser empaquetados y ejecutados en distintos servidores de diferentes proveedores. La herramienta está integrada por un conjunto de Servlet, archivos con extensión HTML y SWF (editado en Flash de Macromedia).

Una de las características principales de una *Aplicación Web* es su relación con el ServletContext. Esta relación está controlada por el contenedor de *Servlet*, que asocia un único ServletContext para cada aplicación, garantizando que estas no se van a colisionar al almacenar objetos en el ServletContext.

El contenedor que reside una *Aplicación Web* es la estructura de directorios en donde están colocados todos los archivos necesarios para la ejecución de la *Aplicación*. Por tanto, *el primer paso* para instalar una *Aplicación Web* consiste en crear la estructura de directorios en donde han de colocarse sus componentes. En el Sistema MECA, el directorio donde se instala la *Aplicación Web* es: */Meca/WEB-INF/classes*. Este contiene todos los Servlets y cualquier otra utilidad complementaria, que se necesite para la ejecución del sistema.

4.2 Consideraciones mínimas de hardware y software

Las condiciones mínimas que se deben presentar en el hardware y el software para la ejecución del Sistema MECA es la siguiente:

Referente al servidor: Reconociendo como servidor a la máquina que proporcionará el funcionamiento de la evaluación a través de Internet, sus elementos son:

- Procesador Pentium III 450 MHz.
- Memoria RAM 256 Mb.
- Disco duro de 6.5 Gb.
- Tarjeta red Ethernet 10/100.

Referente al cliente: Reconociendo como cliente a una máquina que solicita el servicio de evaluación al servidor a través de Internet, sus elementos son:

- Procesador Pentium.
- Memoria RAM 64 Mb.
- Tarjeta red Ethernet 10/100.

A continuación se especifica el nombre y versión de las aplicaciones con las cuales se realizaron las pruebas de evaluación de la herramienta.

Referente al servidor:

- Jakarta Tomcat 1.4.1 como servidor de aplicaciones WEB.
- Internet Explorer 4.0. o posterior.
- SQL Server 2000.
- Windows 2000 Professional.
- J2sdk 1.4.1 en la cual se han programado los Servlets, que se comunican con la Base de datos vía el puente JDBC.

Referente al cliente:

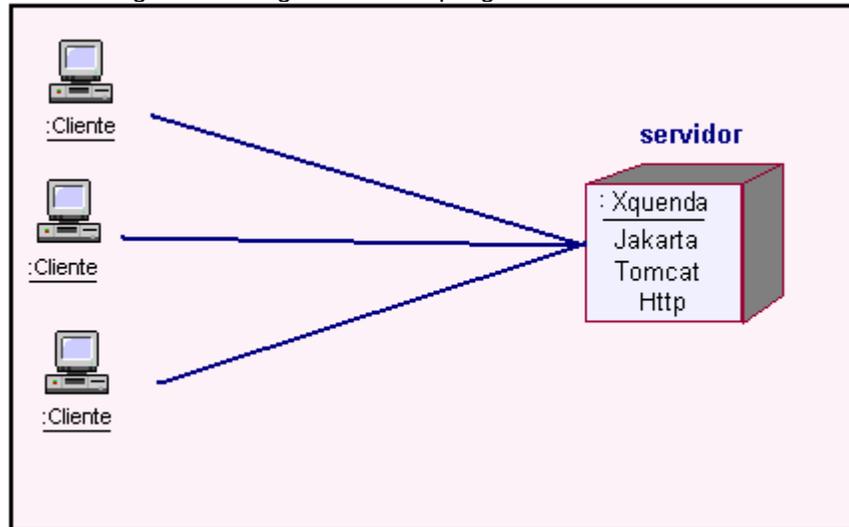
- Internet Explorer 4.0 o posterior.
- Windows 98.

4.3 Arquitectura del Sistema MECA

El sistema se ha desarrollado bajo la arquitectura cliente / servidor de manera que se pueden realizar evaluaciones alternas en un solo punto o en cualquier localización geográfica. Con esta ventaja, los elementos del comité evaluador de una empresa o institución pueden estar separados geográficamente, pero trabajar en una misma actividad: el de la evaluación de sus productos. En este caso, el sistema proporciona el promedio de las evaluaciones al *Presidente*. Con el resultado, se recomienda realizar una reunión física o virtual entre sus miembros para determinar y emitir un dictamen acerca de la calificación final del producto.

Como se observa en la Figura 4.1, el Sistema MECA; implementado bajo la plataforma de cliente / servidor, tiene dos elementos importantes. *Primero*: la computadora desde la cual se accede a la red (*computadora local*), que funciona gracias a un *programa cliente* y *segundo*: ubicado en el otro extremo, se encuentra la computadora a la cual se accede (*computadora remota*), instalada en el CIC-IPN y que facilita lo necesario mediante un *programa servidor*. El programa cliente gestiona la comunicación con el servidor y ofrece las herramientas necesarias para poder trabajar con él. Este programa servidor se encarga de transmitir la información en la forma más adecuada para el (los) usuario (s), ya que un servidor admite múltiples accesos simultáneos. Estos elementos son representados mediante un diagrama de despliegue que muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos [6]. Se pueden utilizar para modelar la vista de despliegue estática del sistema.

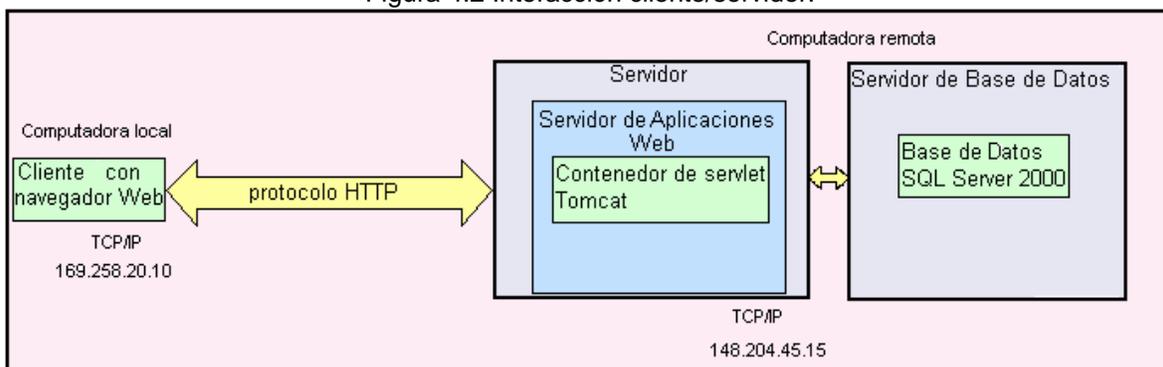
Figura 4.1 Diagrama de despliegue del Sistema MECA.



La Figura 4.2 muestra la interacción del programa cliente / servidor, en el cual, el servidor de aplicaciones tiene un servidor o contenedor Web integrado. Este servidor de aplicaciones se comunica con la Base de Datos durante su procesamiento. Los Servlets incrementan la funcionalidad de la Aplicación Web. Se cargan de forma dinámica por el entorno de ejecución Java del servidor cuando se necesitan. Cuando se recibe una petición del cliente, el contenedor de Web inicia el Servlet requerido. El Servlet procesa la petición del cliente y envía la respuesta de vuelta al contenedor, que es dirigida al cliente.

La interacción cliente / servidor basada en Web usa el protocolo HTTP. EL protocolo HTTP es un protocolo sin estados basado en un modelo de petición y respuesta con un número pequeño y finito de métodos de petición.

Figura 4.2 Interacción cliente/servidor.



Tanto el cliente como el servidor deben utilizar el mismo protocolo de comunicaciones. Este protocolo por excelencia en Internet es *TCP/IP*, que controla la transmisión de archivos electrónicos, dividiendo la información en porciones apropiadamente numeradas para que puedan volver a unirse en su totalidad, o si

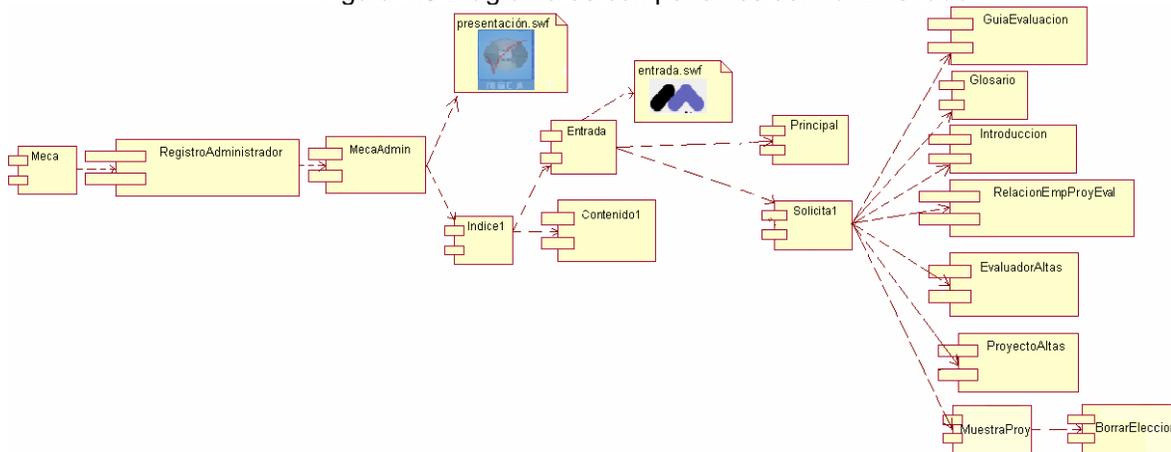
hubiera algún error en la transmisión, retransmitiéndolo. Esta labor de recomposición la realiza el propio *protocolo TCP*.

Por su parte, el *protocolo IP* asigna la dirección (IP) apropiada en Internet a cada paquete de información electrónica que fluye por la Red. Esta dirección identifica e individualiza a todos y cada uno de los ordenadores conectados a Internet, que reciben el nombre de *anfitrión* u *host*. El número IP se compone de cuatro grupos sucesivos de cifras, con un valor comprendido entre 0 y 255, que están separados por puntos. Por ejemplo la dirección IP del servidor donde se encuentra implementado el Sistema MECA es: 148.204.45.15.

En el ámbito del *Modelado Arquitectónico* se encuentran los diagramas de componentes. Estos son un bloque de construcción importante cuando se modelan los aspectos físicos de un sistema.

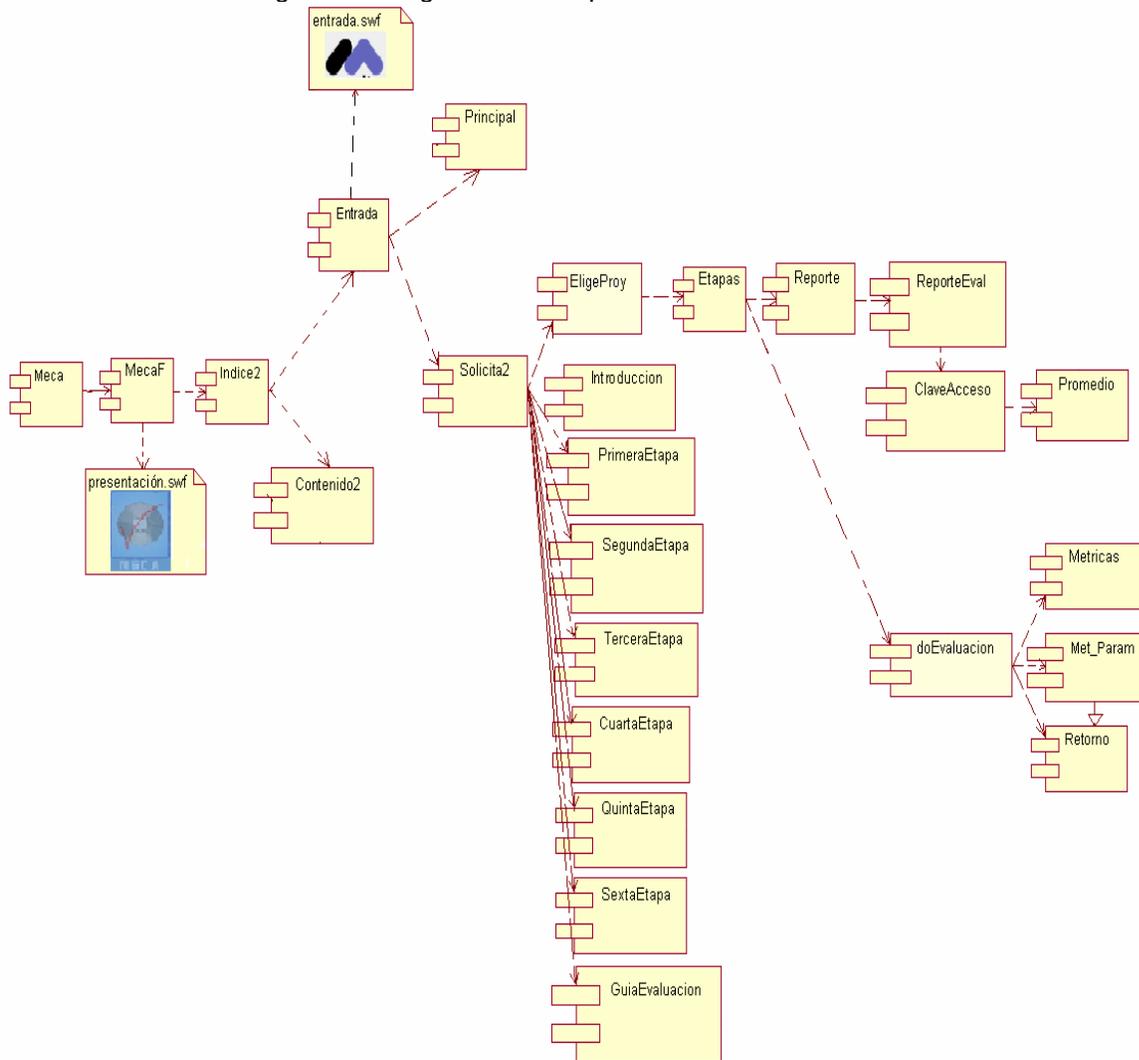
El conjunto de componentes para el caso *Administrador* se muestra en la Figura 4.3 en el cual se observa una distribución interesante a partir de la autenticación de la clave de acceso (método contenido en la clase *Meca*) que proporciona el camino que ha de seguir el usuario en base a su papel. Si se trata de una persona que ingresa por primera vez al sistema y es un *Administrador*, la clase *RegistroAdministrador* proporciona el formato con los datos que se deben cubrir para el registro en los archivos del sistema. La clase *MecaAdmin* (MECA Administrador) contiene la interfaz del *Administrador*, por lo tanto, proporciona las clases diseñadas para este efecto.

Figura 4.3 Diagrama de componentes del *Administrador*.



Los evaluadores formales (de la Calidad en Uso, Presidente o Miembro del comité evaluador) deben contar con una clave de acceso; de lo contrario será necesario solicitarlo a su Administrador; en la Figura 4.4 se observa la clase *MecaF* (MECA Formal) que muestra la interfaz de este papel desencadenando la secuencia de los componentes que realizan la evaluación de tipo Formal.

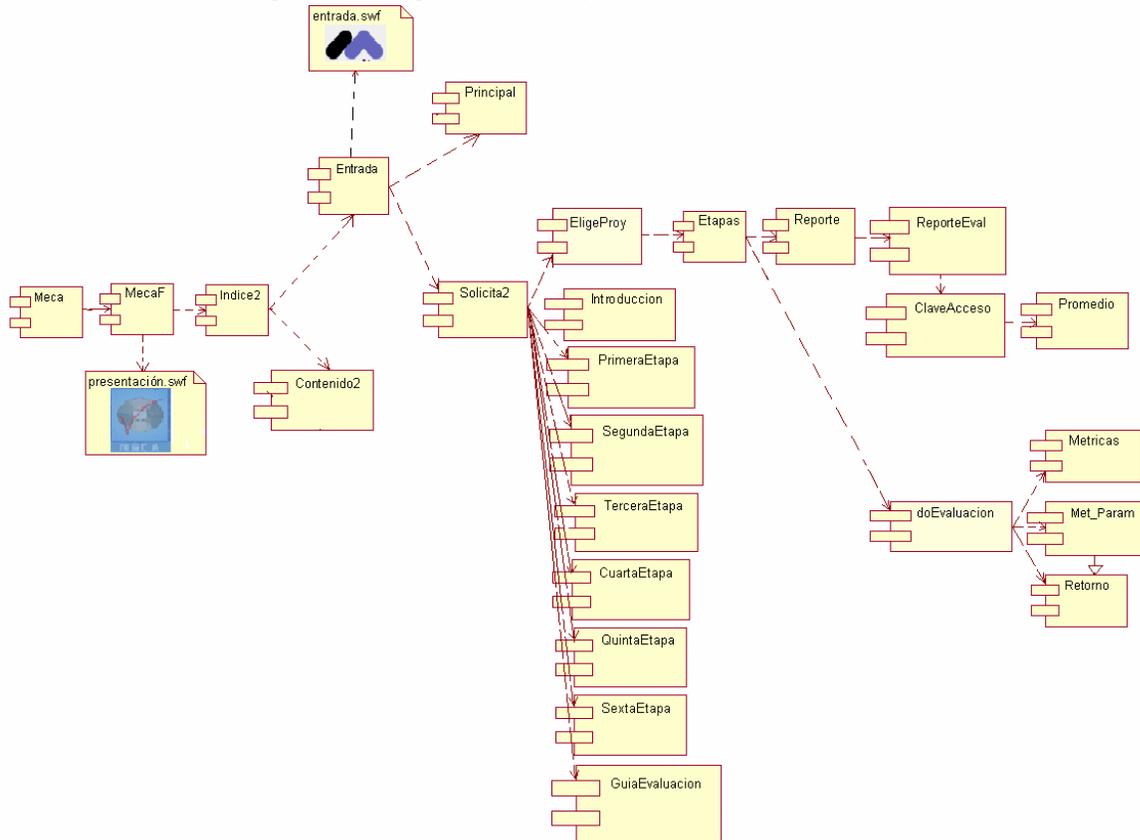
Figura 4.4 Diagrama de componentes del *Evaluador Formal*.



Cuando la persona que ingresa por primera vez es un *Evaluador Informal*, (véase la Figura 4.5) la clase que muestra la interfaz correspondiente es *MecaI* (MECA Informal); misma que dirige la secuencia hacia la evaluación de tipo informal, registrando los datos del Evaluador mediante el componente *RegistroInformal*. El Sistema MECA está implementado bajo componentes *Servlets* que se comunican con la Base de Datos y los archivos con extensión HTML. Estos últimos funcionan como interfaz entre el usuario y el sistema. En este diagrama un componente es un rectángulo con dos pestañas. En su mayoría, los Servlets del Sistema MECA son componentes que realizan la interacción entre la interfaz del usuario y la Base de Datos MECA. La línea discontinua representa la asociación entre los componentes y las interfaces HTML. El icono en los archivos JPG o SWF representa los archivos de gráfico y animación que se encuentran enlazados a la interfaz con extensión HTML. Sirve para darle una mejor presentación al sistema. Los componentes se encuentran asociados entre sí pero también pueden funcionar de manera independiente (mediante el paso de parámetros por valor). La modalidad de ejecución en forma independiente no es recomendable cuando

se busca la seguridad de la información; por lo que se ha establecido el manejo de sesiones en cada uno de estos componentes.

Figura 4.5 Diagrama de componentes del *Evaluador Informal*.



4.4 Instalación del Sistema MECA

Los prerequisites para la instalación del Sistema MECA son los siguientes:

- Instalación de JDK 1.4.
- Instalación de Tomcat 4.1.24.
- Configuración de las variables de ambiente.
- Instalación de SQL Server 2000.
- Restauración de la Base de Datos MECA.
- Creación del manejador ODBC con el nombre: odbc_meca.

Una vez que se hayan debidamente realizado los pasos anteriores, se debe localizar el archivo ejecutable para instalar automáticamente el Sistema (véase el Manual de Usuario del Sistema MECA). Además en el disco compacto que contiene los archivos de la herramienta se encuentra un archivo de desinstalación.

4.5 Prueba e Implementación

A continuación, se presenta el resultado de la evaluación de tres productos diseñados en el CIC-IPN. Estos productos se han evaluado manualmente por un grupo de cuatro expertos del CIC-IPN, por lo que se propuso la valoración usando la herramienta MECA con la finalidad de realizar la comparación de ambos procesos.

4.5.1 Resultados por orden de evaluadores

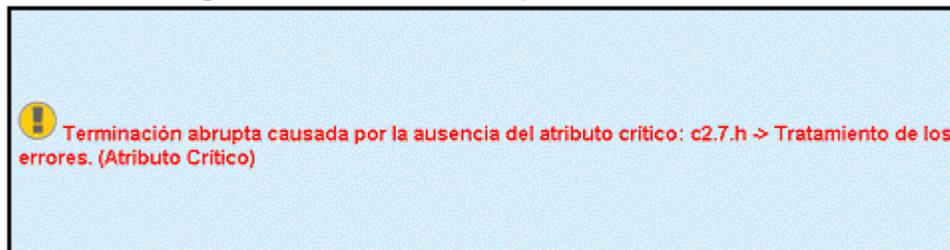
Evaluador uno:

Se presenta la evaluación del primer producto en su etapa *Prueba de Aceptación*. En este proceso el evaluador realizó el papel de Administrador.

Debido a que el producto en la evaluación marcó un error de lectura en la ejecución de uno de sus principales procesos, no se concluyó una de las pruebas principales; por lo tanto, se finalizó este proceso de evaluación.

Durante la prueba se registró en la herramienta la falta de un atributo crítico [15]. En la Figura 4.6 se presenta una de las pantallas de error debido a esta causa.

Figura 4.6 Pantalla de interrupción de la Evaluación.



Dado que una etapa cuenta con atributos; en este nivel existen algunos que son críticos porque su presencia tiene una influencia determinante en la calidad del producto. Los atributos críticos son las propiedades básicas del producto, lo que “debe” estar presente en él para su aceptación. Por lo tanto, la ausencia o incumplimiento de este elemento ocasiona una pérdida de la calidad mínima o básica requerida por el software en su totalidad.

Durante la evaluación, el Sistema MECA detectó la ausencia de un atributo crítico asociado a la subcaracterística: *Tolerancia de errores o fallos* (uno de los parámetros para medir este atributo recibió un valor cero por parte del evaluador). Por esta razón, la característica *Confiabilidad* ha sido también evaluada con cero, ya que el atributo afectado forma parte de esta característica.

La evaluación manual ha pasado por alto la ausencia de este atributo crítico, por lo que lo ha clasificado como un software con *buena* calidad, de *primera* categoría, la

conclusión de *aceptado* y recomienda hacer *pequeñas modificaciones* al producto para mejorar su calidad.

Sugerencia del evaluador uno.

Que el Sistema MECA permita el grabado automático de la evaluación al tiempo indicado por el evaluador.

Repercusión de la sugerencia.

No podrá implementarse el grabado a ciertos minutos de forma automática. No se conoce forma alguna de aplicar esta función en las páginas Web ya que el proceso implica la petición de un servicio del cliente al servidor. Este servicio sólo se puede efectuar, realizando la petición de forma manual ya sea mediante un botón o componente que solicite la petición.

CONCLUSIÓN: la versión del método automatizado ha resultado ser intolerante ante la ausencia del atributo crítico, a diferencia del método manual que ha permitido continuar este proceso logrando un resultado de aceptación al software.

Se modificó el Sistema MECA para permitir la continuación del proceso de evaluación almacenando los errores detectados, afín de presentarlos en el reporte, para que el evaluador comprenda la razón de la calificación final del producto. Se sugiere en tal caso, verificar estos resultados y cubrir los atributos afectados mediante la aplicación de reingeniería en el producto de software y volver a realizar la evaluación.

Evaluador dos.

Se ha evaluado el primer producto en su etapa de *Prueba de aceptación*, la herramienta de evaluación casi ha coincidido con el método manual.

En la Tabla 4.1 se observa la precisión del Sistema MECA en cada calificación de las características y subcaracterísticas de la etapa. El evaluador ha escrito en su propuesta lo incompleto del producto, esta propuesta ha coincidido con el Sistema MECA que ha aceptado el producto recomendando hacer las modificaciones pertinentes para mejorar su calidad.

Tabla 4.1 Comparación del método manual y automatizado del evaluador dos.

Propiedades	Métodos	
	Manual	Automatizado
Funcionalidad	2	2.2
Compleitud	1	3
Consistencia	2	2.2
Corrección	3	3
Usabilidad	2	1.5
Comprensibilidad	2	1.5
Operabilidad	3	1.5
Mantenibilidad	3	3
Claridad	3	3
Concisión	3	3
Estructuración	3	3
Simplicidad	3	3
Reusabilidad	2	3
Modularidad	2	3
Calificación final	2.25	2.4
Clasificación	Bueno	Bueno
Categoría	Primera	Primera
Criterio	Pequeñas modificaciones	Pequeñas modificaciones
Conclusión	Aceptado	Aceptado con modificaciones
Propuesta	Aunque tiene buena calidad no está completa, algo indispensable es lo referente a la configuración o requerimientos del sistema, códigos de error.	

Sugerencia del Evaluador dos.

Que el Sistema MECA permita realizar una mezcla de evaluación cuantitativa y cualitativa. Llámese cuantitativa a la cantidad de elementos que cubren con la información solicitada en el atributo y que están presentes en él; la evaluación cualitativa se refiere al porcentaje de calidad de la información de cada requisito, función o documentación presente en el producto.

Repercusión de la sugerencia: El proceso de la evaluación de la calidad del producto de software implica el conocimiento de las normas básicas de la evaluación en la auditoría de sistemas informáticos, de la metodología del Sistema MECA, de la metodología básica de la ingeniería del software, y del conocimiento del producto objeto de evaluación. La consideración de la sugerencia del evaluador dos de ajustar algunas de las métricas de la herramienta de manera tal que el sistema considerase la evaluación cualitativa haría engorroso el proceso de evaluación; ya que si se estuviese evaluando n elementos, el evaluador tendría que indicar a la herramienta el nivel de cumplimiento de la calidad de cada uno de estos elementos hasta completar la cantidad n definida inicialmente.

Las métricas del Sistema MECA están diseñadas considerando el criterio y la experiencia del evaluador durante el proceso, hecho por el cual se solicita solo el número de elementos que cubre con los requisitos solicitados en la evaluación, dejando a consideración de éste su grado del cumplimiento correcto.

CONCLUSIÓN: El sistema manual se rige por la evaluación cualitativa basándose en el rango de valores de cero a tres para los atributos normales y de cero o tres para los atributos críticos.

El Sistema MECA usa la evaluación cuantitativa pero implícitamente aplica la cualitativa mediante el uso de las fórmulas definidas en la norma ISO/IEC 9126 y las adicionadas a la metodología. La finalidad es minimizar el número de datos solicitados en la evaluación haciendo el proceso más sencillo pero no simple al considerar el criterio del evaluador en la determinación del número de los elementos que cubren los requisitos de calidad.

Evaluador tres.

Durante este proceso se evaluó el segundo producto; en el cual, el Sistema MECA indicó la ausencia de un atributo crítico. Reanudando el proceso, se obtuvo una calificación final de 1.2, el producto se clasificó como *Regular*, categoría de *Segunda*. El sistema sugirió realizar *Grandes modificaciones* para mejorar la calidad del producto, por lo tanto, el software fue rechazado. El resultado fue el mismo en la evaluación manual.

Sugerencia del Evaluador tres: El Sistema MECA presentó atributos cuyos parámetros fueron difíciles de determinar por el evaluador durante el breve período de valoración. Los parámetros aparecen en la Figura 4.7.

Figura 4.7 Métricas complejas para el evaluador Tres.

Tiempo para intentar la operación (Tiempo basado en el primer intento).	0
Tiempo para terminar una tarea específica al primer intento.	<input type="text" value="0"/>
Tiempo de inicio de la operación para tareas al primer intento.	<input type="text" value="0"/>
Tiempo de operación de la prueba.	<input type="text" value="0"/>
Tiempo para intentar la operación (tiempo).	0
Tiempo para terminar una tarea específica.	<input type="text" value="0"/>
Tiempo del inicio de operación para la tarea.	<input type="text" value="0"/>
Tiempo de operación de la prueba.	<input type="text" value="0"/>
Tiempo para intentar la operación (tiempo basado en reaprendizaje).	0
Tiempo para terminar una tarea después de una larga interrupción.	<input type="text" value="0"/>

Otro ejemplo de atributos con mayor complejidad se muestra en la Figura 4.8.

Figura 4.8 Métricas complejas en la evaluación.

Tiempo de un ciclo calculado o simulado confirmado en la revisión.	<input type="text" value="0"/>
Tiempo de un ciclo diseñado o estimado.	<input type="text" value="0"/>
Tiempo de un ciclo.	0
Tiempo medio actual.	<input type="text" value="0"/>
Error de fluctuación del tiempo.	<input type="text" value="0"/>
Tiempo requerido diseñado en la especificación	<input type="text" value="0"/>
Rango de error de fluctuación permisible requerido en la especificación.	<input type="text" value="0"/>
Tiempo de respuesta.	0
Tiempo de respuesta calculado o simulado confirmado en la revisión.	<input type="text" value="0"/>
Tiempo de respuesta diseñado o estimado.	<input type="text" value="0"/>
Rápido tiempo de respuesta.	0

Otra sugerencia del Evaluador tres es acerca de la implementación de un mecanismo de ayuda que permita explicar detalladamente la forma de evaluar estos atributos difíciles para él durante la evaluación.

Repercusión de la sugerencia: La metodología que implementa el Sistema MECA está diseñada para ser utilizada por usuarios con el tiempo suficiente en la búsqueda minuciosa del comportamiento de los atributos del software. Por ejemplo, el comportamiento de las funciones de programación, tiempo de procesamiento, tiempo de las tareas, distribución de la memoria, etc.; así mismo, se deben considerar entre los conocimientos, las normas básicas de la auditoría informática.

CONCLUSIÓN: La valoración realizada por el tercer Evaluador fue un caso en la cual, la ausencia del tiempo suficiente para cubrir una buena evaluación repercutió en los resultados del producto. Aún cuando algunos atributos específicos no se hayan presentado en el método manual, este resultado fue el mismo. Para una mejor comprensión de los elementos evaluados, se implementó en la herramienta un archivo de ayuda que permite conocer todo sobre MECA. Por ejemplo, la metodología, el concepto de cada característica de calidad, subcaracterística y atributo, los objetivos de cada atributo, la fórmula aplicada en la evaluación y la forma de evaluar el atributo. Esta adición a la herramienta la hace más consistente en la evaluación.

Evaluador cuatro:

Se evaluó el tercer producto en su etapa *Documentación de soporte* obteniéndose una calificación similar al método manual. En la Figura 4.9 se muestran estos resultados.

Figura 4.9 Resultado del producto tres.

Tabla de Resumen				
RESUMEN DE CALIDAD	NIVEL DE CLASIFICACIÓN	CATEGORIA DE CALIDAD	CRITERIO DE EVALUACIÓN	CONCLUSIÓN
Funcionalidad. : 1.3	3.- [] Muy bueno	MB [] Superior	[] Sin modificaciones	[] Aceptado sin modificaciones
Usabilidad. : 0.8	2.- [X] Bueno	B [X] Primera	[X] Pequeñas modificaciones	[X] Aceptado con modificaciones
Mantenibilidad. : 2.4	1.- [] Regular	R [] Segunda	[] Grandes modificaciones	[] Rechazado, necesita grandes modificaciones
Reusabilidad. : 3	0.- [] Malo	M [] Tercera	[] Nueva elaboración	[] Rechazado, necesita nueva elaboración
Valor Total : 1.9				

Sugerencias del Evaluador:

Que el Sistema MECA implemente en su ayuda una liga a las normas de la ISO/IEC, IEEE y del Modelo para el Aseguramiento de la Calidad de las cuales ha basado su metodología.

Repercusión de la sugerencia: Implementando la liga a las normas y metodología MECA hace más consistente el proceso porque el evaluador realiza consultas acerca del concepto de los atributos que está evaluando.

En la ayuda se incluye el objetivo y la forma de llevar a cabo la prueba del atributo.

CONCLUSIÓN: La evaluación automatizada del tercer producto coincidió en sus resultados con respecto al método manual. La sugerencia del evaluador cuatro se implementó en la herramienta, presentándose un módulo llamado *Guía de la Evaluación* y se implementó una ayuda para cada una de las etapas del ciclo de vida del producto. La finalidad es apoyar al evaluador durante este proceso, véase la Figura 4.10.

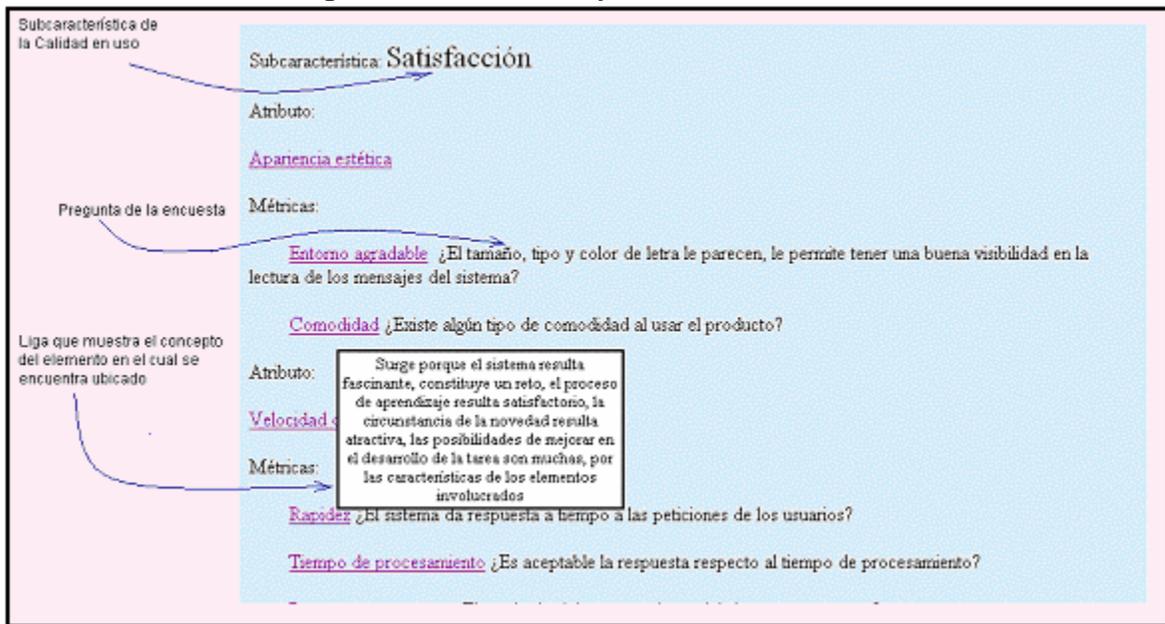
Figura 4.10 Módulos implementados al Sistema MECA para el apoyo del evaluador



En la Figura 4.11 se presenta la primera etapa de la metodología MECA, con información que puede considerarse necesaria durante la evaluación.

Una vez analizados los resultados de cada una de las preguntas de la encuesta a la Calidad en Uso; se observa que la primera métrica tiene cierta dificultad en algunos usuarios. La pregunta es: ¿El lenguaje que usa el sistema es claro y comprensible?. Este problema se ha generado por la falta de los módulos de ayuda en la primera versión de la herramienta. Con la implementación del módulo *Guía para la Encuesta*, presentado en la Figura 4.13 se ha logrado una mejor comprensión de las preguntas porque cuenta con la definición de los elementos de evaluación afín de conocer concretamente la esencia de la métrica evaluada.

Figura 4.13 Pantalla de ayuda a la encuesta.



Siguiendo con el análisis de la Figura 4.12 las métricas dos, tres, cuatro, cinco, seis y siete se han desempeñado correctamente. La métrica ocho cuya pregunta es: ¿El sistema da respuesta a las peticiones del usuario?; cuenta con un cincuenta por ciento de usuarios satisfechos. Esto podría deberse al índice de tráfico que ocurre en la red a determinadas horas del día. Normalmente, el tiempo de respuesta a las peticiones del usuario oscila entre uno a dos segundos.

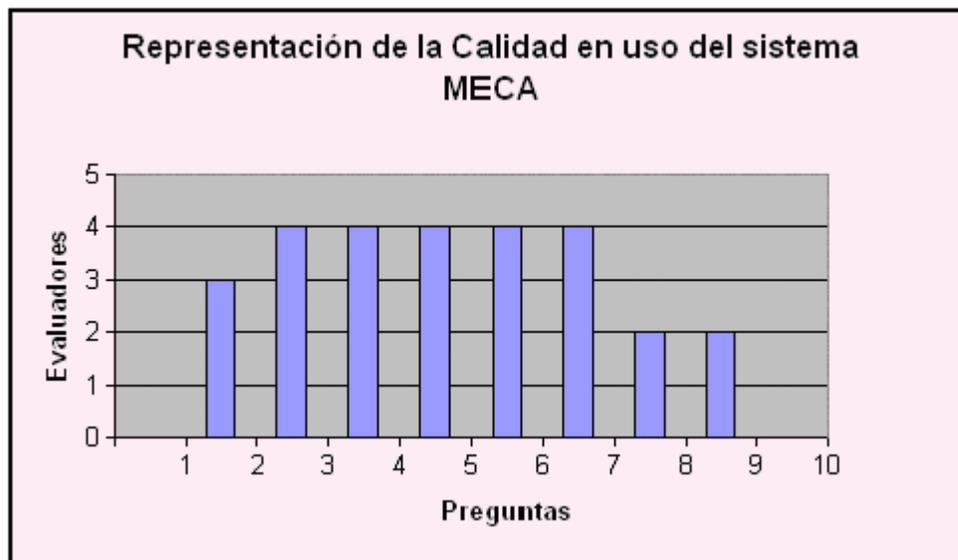
La métrica nueve con la pregunta: ¿El sistema administra sus recursos en respuesta a las peticiones del usuario?, también cuenta con la mitad de usuarios satisfechos. Esta métrica se ha reforzado en la herramienta MECA con el uso de Servlets en su totalidad. En un servidor con tamaño de memoria RAM de 256 Mb o mayor, la administración de la memoria en la distribución de los servicios a los clientes resulta ser aceptable. Esta calidad mejora con la escalabilidad a un procesador mayor que el utilizado en la implementación de la herramienta.

La métrica diez cuya pregunta es: ¿Existe algún tipo de comodidad al usar el producto? se ve afectada ante la insatisfacción del usuario. El problema del desempeño de esta métrica se ha consultado con los evaluadores y resulta ser el

efecto que ha producido la falta de la ayuda en la primera versión, porque se argumenta la incomprensión de la métrica en ese sentido. Con la implementación del módulo de ayuda a la encuesta se ha minimizado el problema de esta métrica.

La Figura 4.14 muestra una perspectiva diferente de los resultados de la evaluación de la Calidad en Uso del Sistema MECA en la que se presenta una población de uso de cuatro personas y las calificaciones que cada una le ha proporcionado. Esta gráfica también suele ser recomendable para el diseñador del producto. En ella se presenta una visión más clara del comportamiento del producto en los clientes finales.

Figura 4.14 Representación gráfica de la Calidad en Uso del sistema.



Conclusiones y recomendaciones



Se ha desarrollado una aplicación que apoya al proceso de control de la calidad, evaluando el nivel de los resultados de las etapas del ciclo de vida del software, aplicando un modelo cualimétrico que sigue los lineamientos de la ISO/IEC 9126, por lo que se ha cumplido en su totalidad el objetivo general de este trabajo.

Los objetivos específicos también han sido cubiertos, éstos a continuación se enumeran:

- Se han añadido a las métricas de MECA, las subcaracterísticas y atributos de la Calidad en Uso como aportación de este trabajo al modelo cualimétrico.
- Se ha implementado las características de calidad del MECA, por lo que ha surgido un control de la calidad de los sistemas informáticos de acuerdo a los estándares internacionales ISO.
- Al implementarse las características de Calidad, se han probado y verificado los resultados de las etapas del ciclo de vida del software.
- El Sistema MECA se ha instrumentado de acuerdo al paradigma orientado a Objetos y se ha programado usando el lenguaje Java obteniéndose las ventajas de éste.

Aunado a los objetivos generales y específicos cumplidos, se precisan otras características del Sistema MECA:

Sirve de apoyo al proceso de auditoría informática y de control de calidad ya que permite la evaluación y determinación del nivel de la calidad de los productos en todas las etapas del ciclo de vida del software.

Trata al software como producto terminado. Al evaluar la calidad del software a través de las etapas del ciclo de vida, permite desglosar sus propiedades. El desglose es para conocer detalladamente la presencia de sus atributos y verificar su comportamiento para determinar el grado en que las propiedades del producto cumplen con las especificaciones de calidad internacional.

La instrumentación del Sistema bajo el paradigma Orientado a Objetos y la programación Java utilizados lo hace más consistente, funcional y portable a los cambios de la plataforma de hardware y software. Así mismo; el sistema se presta al enriquecimiento de nuevas características al modelo sin que este mantenimiento afecte el funcionamiento de sus componentes actuales.

El Sistema MECA sigue un modelo más completo que los presentados en el capítulo II. La cantidad de usuarios se considera amplia, ya que contempla en su panorama de uso a los auditores informáticos, técnicos diseñadores de sistemas, profesionales en computación, analistas y jefes de proyectos, administradores de sistemas y usuarios con los conocimientos básicos de cómputo que busquen conocer el nivel de la calidad de un producto de software.

Casi todos los productos elaborados para el uso del cliente habían tenido indicadores para medir la calidad del mismo, esta medición había sido difícil de implementar mediante un software por ser intangible. La presencia de la herramienta proporciona el mejoramiento del proceso de evaluación de los productos de software. Debido al surgimiento de las normas de calidad ISO se han mejorado los acuerdos en qué y cómo medir este producto.

Con la medición de la Calidad en Uso mediante la aplicación de la encuesta a los usuarios, el Sistema MECA permite conocer el grado de satisfacción del cliente con respecto al producto. Se ha propuesto el análisis de los resultados de las preguntas a través de la población de usuarios encuestados con la finalidad de conocer las métricas que satisfacen al cliente y corregir las que fallan en este aspecto, lo cual redundará en un mejoramiento de la calidad del producto. El informe de la Calidad en Uso representa los resultados de los procesos utilizados para prevenir los errores durante el desarrollo del producto y las prácticas, usadas para detectar los errores que se traducen en software más competitivos, demandados y confiables; en fin, productos desarrollados con calidad. El costo de la calidad no consiste en hacer buenos productos sino el precio que se paga por no hacerlo correctamente.

La herramienta instalada en un servidor de aplicaciones y consultada a través de Internet, espera muchas visitas, hecho por el cual, la creación de Servlets es un buen requerimiento para incrementar su funcionalidad como Aplicación Web.

La herramienta de evaluación cuenta con un control de seguridad basado en claves de acceso. En caso de no haberse finalizado un proceso de evaluación, se recomienda, guardar el avance. El sistema asigna una calificación considerando solo los datos presentes en el avance (generalmente, será un valor menor al esperado como calificación final). La información estará presente en los archivos por tiempo indefinido; hasta que el evaluador requiera reanudar el proceso. Es necesario conservar la clave de acceso, el nombre del producto y la etapa de evaluación para reanudar el proceso. De otra manera, será necesario iniciar nuevamente la evaluación.

En todos los componentes del sistema ha sido diseñado un método de protección de las páginas web a través de la programación Java mediante el uso del control de sesiones, de manera tal que no se permite ejecutar una página sin antes autenticarse. Esta ventaja cubre la posibilidad de mantener la integridad del sistema ante intentos de ataques virtuales.

La población de usuarios que se espera es cubierto por la capacidad del Manejador de Base de Datos, ya que el SQL Server 2000 puede soportar una gran cantidad de registros por tabla.

El sistema es fácil de mantener y actualizar. Es decir; Se pueden añadir subcaracterísticas a una característica presente en el sistema o atributos a una subcaracterística existente. Para estas tareas se requiere el conocimiento de los niveles de calidad en el modelo, el conocimiento de SQL básico, del procesador de consultas del SQL Server 2000 y el lenguaje Java básico.

El presente trabajo ha demostrado que, una vez más, la cultura del orden es primordial al diseñar un producto u ofrecer un servicio de software.

Al hacer uso del Sistema MECA se recomienda tener a disposición del personal los requerimientos de la documentación, software y sus derivados (según sea la etapa a evaluar) y verificar detalladamente el contenido de estos durante la evaluación.

5.1 Trabajos a Futuro

Aportar al Sistema MECA la característica: *Reusabilidad* en la etapa del *Estudio, análisis e investigación preliminar, Análisis detallado y determinación de los Requerimientos*. El catálogo de elementos de evaluación del sistema conserva la estructura y la definición de la tesis: *Actualización y Automatización de un Modelo Cualimétrico para la Evaluación de Calidad del Software* [8]. A cambio, el sistema deja una Base de Datos cuya estructura se presta para el análisis de los resultados de la evaluación.

Con la Base de Datos del Sistema MECA, realizar un estudio y análisis mediante el uso de procedimientos estadísticos y tecnología OLAP para interpretar y clasificar el impacto de la evaluación en el producto y transitivamente en el cliente.

Aportar la asignación de pesos diferentes a las ocho características de calidad del sistema (Funcionalidad, Confiabilidad, Usabilidad, Portabilidad, Eficiencia, Mantenibilidad, Reusabilidad y Calidad en Uso), actualmente tienen el mismo peso en la evaluación.

Permitir que el administrador adapte el modelo a las características propias del producto evaluado, exceptuando la eliminación de características, así como la eliminación de todas las subcaracterísticas de una característica, porque

modificaría en gran escala la estructura del modelo. Tampoco se permitirá la eliminación de atributos críticos, ya que estos forman parte de los requerimientos básicos del producto en la etapa, además el resultado de estas modificaciones haría que el modelo no se ajustara a los requerimientos que plantea la norma ISO/IEC 9126 [23].

Anexo

Conclusiones de la evaluación

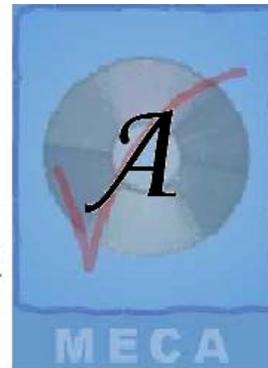


Tabla A.1 Resumen de la medición, clasificación, categorización y conclusiones de la Evaluación de la Calidad del Software

RANGO DE VALORES	NIVEL DE CLASIFICACIÓN	CATEGORÍA DE CALIDAD	CRITERIO DE EVALUACIÓN	CONCLUSIÓN
2,5 - 3	MB - MUY BUENO	SUPERIOR	SIN MODIFICACIONES	ACEPTADO
1,5 - 2,4	B - BUENO	PRIMERA	PEQUEÑAS MODIFICACIONES	ACEPTADO
0,5 - 1,4	R - REGULAR	SEGUNDA	GRANDES MODIFICACIONES	RECHAZADO
0 - 0,4	M - MALO	TERCERA	NUEVA ELABORACIÓN	RECHAZADO

CALIDAD SUPERIOR: Es la máxima categoría de calidad que puede alcanzar un software evaluado.

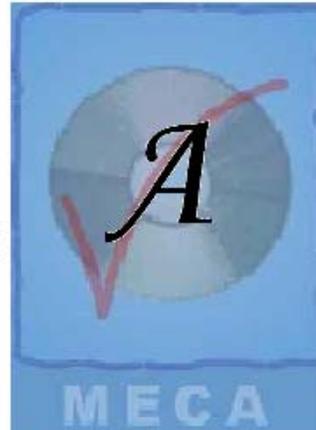
PRIMERA CALIDAD: Es la categoría alta de calidad otorgada al software que ha alcanzado un nivel muy bueno en su evaluación. Quiere decir que el producto es aceptado, aun cuando existen algunos defectos en él que pueden corregirse.

SEGUNDA CALIDAD: Es la categoría baja otorgada al software que ha alcanzado un nivel regular en su evaluación. Esta calidad es inferior a la esperada en la satisfacción de los requerimientos de los clientes. Quiere decir que el software es rechazado para que sea mejorado.

TERCERA CALIDAD: Es la calidad muy baja otorgada al software que ha alcanzado un nivel malo en su evaluación. Quiere decir que el software es rechazado, solicitando una nueva elaboración.

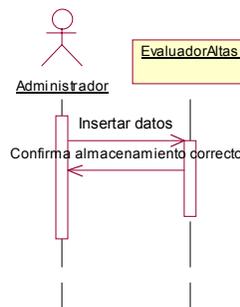
Apéndice

Diagramas de secuencia



La Figura AP.A.1 muestra el diagrama de secuencia a un alto nivel de abstracción en el que el Administrador ingresa los datos de los evaluadores, se activa una secuencia de eventos [31] que se muestran en la Tabla AP.A.1

Figura AP.A.1 Diagrama de secuencia de alto nivel de abstracción del Registro de evaluadores.



Se muestra el caso de uso *Registrar* en cual la columna izquierda contiene el comportamiento del Actor y en la derecha el del Sistema MECA.

Tabla AP.A.1 Secuencia de eventos del Registro

Acción del actor	Acción del sistema
1.- Este caso de uso comienza cuando el Actor (Administrador) llega al sistema.	2.- Pide la clave de acceso.
3.- El Administrador ingresa la clave de acceso a través del teclado.	4.- Clave Incorrecta entonces ve a paso 8 sino ve a paso 5.
	5.- Ofrece las opciones del menú para el Administrador.
6.- Selecciona la opción registro: <ul style="list-style-type: none"> a) Si se registra el Proyecto, véase la Sección Proyecto. b) Si se registra el Evaluador, véase la Sección Evaluador c) Si se elige la opción Asignación de proyectos, véase la Sección Asignación d) Si se elige la opción Relación de Proyectos y Evaluadores, véase la 	7.- Devuelve la clave del evaluador registrado o en el caso de la opción <i>Asignación de proyectos</i> , actualiza los datos del sistema relacionando al evaluador el proyecto correspondiente. ve a paso 13

Sección Relación	
	8.- Muestra el formato de registro del Administrador
9.- Escribe los datos que pide el formato y los envía.	10.- Registra los datos recibidos y proporciona una clave de acceso al Administrador.
11.- Toma nota de la clave proporcionada	12.- ve a paso 5
13.- Anota la clave de registro y se marcha.	

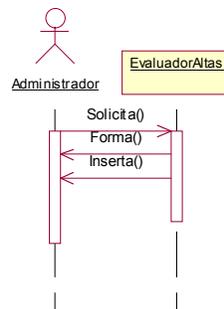
Las secciones independientes que se muestran a continuación constituyen los submódulos del Registro. La Tabla AP.A.2 Sección Evaluador muestra la secuencia de eventos para la sección Evaluador.

Tabla AP.A.2 Sección Evaluador

Acción del actor	Acción del sistema
1.- El Administrador introduce los datos del Evaluador que se le solicite.	2.- Verifica que los datos estén completos.
	3.- Guarda los datos en los archivos del sistema

La Figura AP.A.2 muestra el diagrama de secuencia a un bajo nivel de abstracción, en el que el Administrador solicita el ingreso de los datos de un evaluador, la clase EvaluadorAltas ejecuta el método *forma()* que contiene todos los datos requeridos del evaluador y hace referencia al método *inserta()* que almacena estos datos en la tabla Evaluador.

Figura AP.A.2 Diagrama de secuencia de bajo nivel de abstracción del Registro de Evaluadores.



La Figura AP.A.3 muestra el diagrama de secuencia del Registro de proyectos, en el cual el Administrador solicita el ingreso de los datos del Proyecto, en la Tabla AP.A.3 se encuentra el flujo de eventos del Registro del proyecto.

Figura AP.A.3 Diagrama de secuencia del Registro de Proyectos

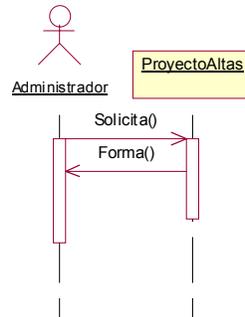


Tabla AP.A.3 Sección Proyecto

Acción del actor	Acción del sistema
1.- El Administrador introduce los datos del Proyecto que se le solicite.	2.- Verifica que los datos estén completos.
	3.- Guarda los datos en los archivos del sistema.

La Figura AP.A.4 muestra los métodos de la clase *ProyectoAltas*, particularmente el método *forma()* muestra el formulario con los datos que Administrador deberá cubrir para que el sistema administre correctamente sus datos en la evaluación, el método *Inserta()* de esta misma clase efectúa la inserción en la tabla Proyecto.

Figura AP.A.4 Diagrama de secuencia de bajo nivel del Registro de Proyectos.



La Figura AP.A.5 muestra la secuencia que sigue el sistema ante la asignación de proyectos a evaluadores, la Tabla AP.A.4 muestra el flujo de los eventos del diagrama.

Figura AP.A.5 Diagrama de secuencia de la Asignación de Proyectos.

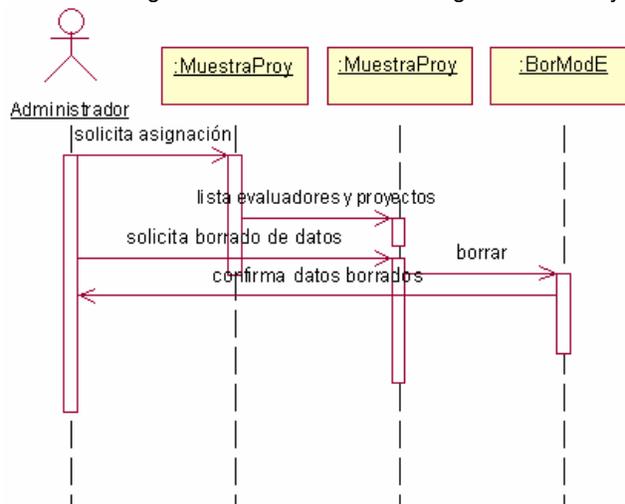
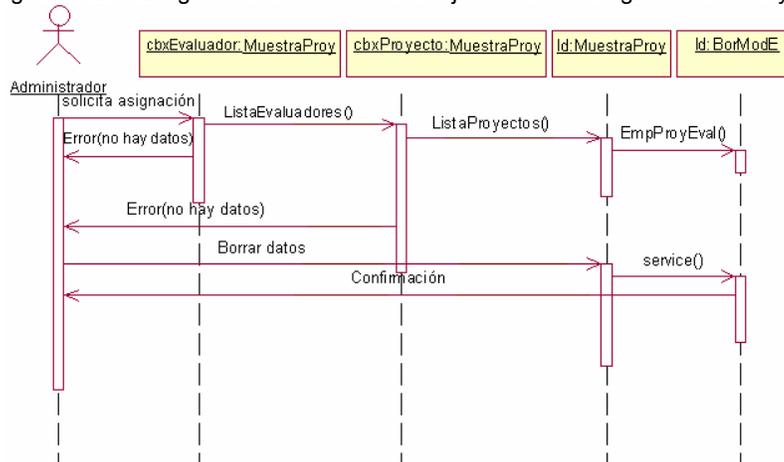


Tabla AP.A.4 Asignación de Proyecto

Acción del actor	Acción del sistema
1.- El Administrador solicita la Asignación de proyectos.	2.- Muestra la lista de Proyectos y Evaluadores de la Empresa.
3.- Elige el Proyecto, el evaluador y el papel que desempeñará el Evaluador en el Proyecto.	4.- Verifica que los datos estén correctamente asignados (no admite la asignación de dos Presidentes en la evaluación del mismo Proyecto y la asignación de Presidente a un Evaluador de la Calidad en Uso).
	5.- Guarda los datos en los archivos del sistema.
6.- Verifica que los datos hayan sido correctamente asignados y pasa a 9 sino solicita el borrado de datos incorrectos.	7.- Borra los datos elegidos de la Base de Datos MECA
8.- Pasa a 4	
9.- Se marcha	

La Figura AP.A.6 muestra la ejecución de la clase *MuestraProy* cuyos métodos hacen posible la asignación de evaluadores y Proyectos, *cbxEvaluador* hace posible el despliegue de la lista de los Evaluadores registrados en el sistema, *cbxProyecto* muestra la lista de Proyectos a evaluar, el Administrador elegirá un elemento de cada lista mostrada, el sistema a su vez registrará esta selección y mostrará mediante *EmpProyEval()* una lista con los elementos elegidos. En caso de ocurrir una equivocación durante la asignación, el Administrador podrá borrar la selección de la lista.

Figura AP.A.6 Diagrama de secuencia de bajo nivel de la Asignación de Proyectos.



En la Figura AP.A.7 muestra el diagrama de secuencia de bajo nivel de la evaluación. En el que un usuario pide el servicio mediante la clase *solicita*. Esta clase presenta la interfaz al Evaluador solicitante; así mismo, envía parámetros (la clave del Evaluador, la clave de la empresa y la clave del Administrador) a la clase *EligeProy*.

La clase *EligeProy* muestra la lista de los Proyectos asignados al Evaluador; quien elige su proyecto. Después de esta elección, se ejecuta la clase *Etapas*.

La clase *Etapas* contiene los métodos que muestran en pantalla el formato de la etapa elegida por el Evaluador. Inicia la inserción de datos al formato; al almacenar estos datos, se ejecuta la clase *Evaluación*.

La clase *Evaluación* almacena el avance, iniciando la asignación de valores a los atributos, subcaracterísticas, características y etapa. Este proceso activa la clase *Retorno* para conocer la clave del atributo.

Evaluación solicita a la clase *Met_Param* los parámetros del atributo; con la finalidad de obtener el valor de este.

Con los parámetros, *Evaluacion* pide a la clase *Metricas* el Método o fórmula correspondiente al atributo, almacena el resultado en su ubicación de la Base de Datos; esta operación se repite para todos los atributos de la subcaracterística, característica y etapa.

La clase *Evaluacion* devuelve a *Etapa* el promedio de la etapa. Esta clase, a su vez, muestra al Evaluador la posibilidad de imprimir el reporte.

La clase *Reporte* muestra el informe de la evaluación, también muestra el promedio del comité, (solo a los usuarios reconocidos como Presidente del comité evaluador); en tal caso, *Reporte* se comunica con la clase *ClaveDeAcceso*.

ClaveDeAcceso autentifica al usuario y se comunica con la clase *Promedio*. *Promedio* muestra el promedio correspondiente a la etapa.

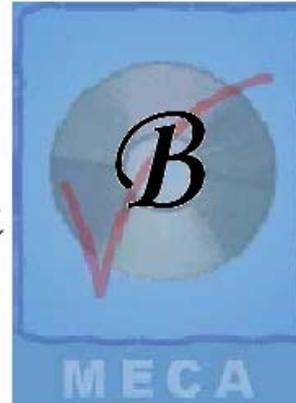
El otro informe de evaluación, lo presenta la clase *ReporteEval*, este es el reporte final.

El diagrama de secuencias muestra la parte de la evaluación en la que se desempeña el método *TwoInputA* de la clase *Métricas*; este método es usado frecuentemente en la evaluación para obtener el valor del atributo.

Una vez obtenido el valor del atributo, el método *TwoInputA* referencia al método *Evaluation* para darle un valor al atributo en el rango de 0 a 3. En esta fase se solicitan y aplican los métodos correspondientes.

Apéndice

Diagramas de colaboración



Como se muestra en la Figura AP.B.1 El diagrama de colaboración se construye a partir del objeto *MaxIdEvaluador* que participa en la colaboración como nodo del grafo, este objeto realiza el método *Forma()* que muestra el formulario, *Inserta()* almacena en la tabla *Evaluador* los datos involucrados e *IsNull()* como validación de los datos.

Figura AP.B.1 Diagrama de colaboración del Registro del Evaluador



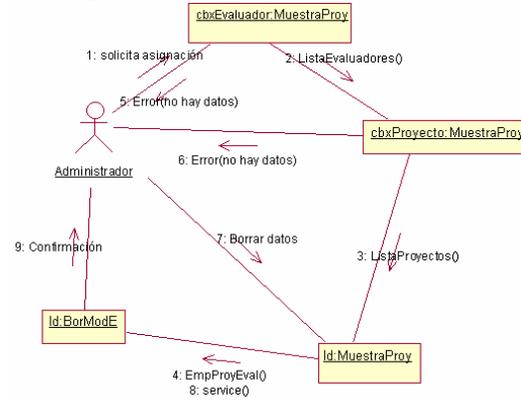
La Figura AP.B.2 Muestra la interacción entre los objetos y métodos para el registro del proyecto, tiene muchas similitudes en la lógica del registro de Evaluadores.

Figura AP.B.2 Diagrama de colaboración del Registro del Proyecto



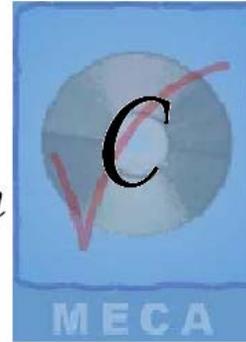
La Figura AP.B.3 muestra los objetos que intervienen en la asignación de proyectos a evaluadores, el Administrador solicita la asignación, el sistema muestra mediante el atributo *cbxEvaluador* y *cxpProyecto* de la clase *MuestraProy* las listas de los Evaluadores y Proyectos, el sistema muestra mediante el método *EmpProyEval()* los elementos elegidos por el Administrador, de haberse elegido erróneamente, la clase *BorModE* conectada a esta clase efectúa el borrado de los datos.

Figura AP.B.3 Diagrama de colaboración de la Asignación de Proyectos



Apéndice

Cálculo de los Puntos de Función



C.1 Características del dominio de información

Al analizar el dominio de la información [36] del Sistema MECA se puede partir de tres puntos de vista sobre los datos y el control:

- **Contenido de la información y sus relaciones:** se analizan los atributos que conforman cada etapa de la evaluación del Sistema MECA y cómo se interrelacionan.
- **Flujo de la información y sus relaciones:** se refiere a la forma en que cambian los datos introducidos por el usuario, relacionándose posteriormente con la biblioteca de métricas de MECA, para producir respuestas tales como el nivel de la calidad en la etapa evaluada.
- **Estructura de la información:** requiere del análisis del tipo de estructura de datos que se aplicará en el diseño del Sistema MECA, tal es el caso del uso de tablas que administran el modelo de evaluación; o bien, típicamente en la Administración de la Base de Datos, que contiene el resultado de la evaluación.

A continuación, se describe las características del dominio de información del Sistema MECA con las actividades que se realizan para obtener el grado de complejidad de la herramienta; estas actividades consisten en medir el número de líneas de código del software, las orientadas a la función que fueron propuestas por primera vez por Albretch [36]. El método para hallar los Puntos de Función consiste en contabilizar los elementos del software tales como las entradas, las salidas, archivos lógicos e interfaces externas y aplicarles la fórmula de los Puntos de Función. Esta fórmula contabiliza los elementos del software con los valores que determinan su complejidad, en él intervienen constantes definidas [36].

Eliminado: métricas

Eliminado: orientadas al tamaño,

Eliminado: que se miden a través d

Eliminado: y

Eliminado: las métricas

Eliminado: a través de los Puntos de Función

Eliminado: ,

Eliminado: el

Eliminado: a

Eliminado: métricas

C.2 Número de entradas de usuario

Una de las medidas del proyecto de software para calcular los Puntos de Función consiste en sumar el número de entradas del usuario. Se debe contar cada dato único de usuario o entrada de control que se introduce en los límites de la aplicación y actualiza un archivo lógico interno, conjunto de datos, tabla o dato

independiente. Esto incluye archivos de entrada y transacciones recibidas de otras aplicaciones (el Sistema MECA no cuenta con este tipo de entrada).

Una entrada se considera única si:

- Tiene un formato diferente
- Tiene el mismo formato que otra entrada pero requiere una lógica diferente de procesamiento, o se modifica un archivo interno lógico diferente.
- Las funciones de actualización (insertar, modificar, eliminar) requiere diferente lógica de procesamiento; por lo tanto, se cuentan por separado.

Las entradas se clasifican en simples, medias y complejas [36]. Las entradas del Sistema MECA, se consideran simples porque no implican el uso de más de dos archivo lógicos y mayor de cuatro datos referenciados [28].

En la Tabla AP.C.1 se presenta el total de las entradas simples de usuarios que proporcionan datos para el funcionamiento de la herramienta de evaluación.

Tabla AP.C.1 [Tabla de entradas de usuario](#)

Entradas de usuario		
Administrador	Clave de acceso	1
	Registro	1
	Alta a Evaluadores y Proyectos	1
	Asignación de Proyectos	1
	Eliminación por asignación errónea	1
Evaluador	Atributos comunes	1
	Formato de evaluación	1
	Promedio del comité	1
Evaluador de la Calidad en Uso	Formato de la encuesta	1
	Comentarios	1
Total		10

C.3 Número de salidas de usuario

Se debe contar cada dato único o salida de control generado proceduralmente. Esto incluye informes y mensajes a otras aplicaciones y usuarios.

Una salida se considera única si:

- Tiene formato diferente.
- Tiene el mismo formato que otra salida pero requiere diferente lógica de procesamiento.

Además de las pantallas y los listados (papel o pantalla), también pueden ser salidas [28]:

- Archivos de transacción enviado a otra aplicación.
- Facturas.
- Cheques.
- Transacciones automáticas.
- Mensajes al usuario.
- Gráficos.
- Archivos back-up, etc.

No se deben contar como salidas:

- Cabeceras de columna, títulos, número de página.
- Mensajes individuales (información, confirmación o respuestas a consultas de error).
- Salida en igual formato y lógica que ya se han contado para otro soporte.

Al igual que en las entradas, las salidas de usuario se clasifican en simples, medias y complejas [36]. Las salidas del Sistema MECA se consideran simples y medias. Simples porque no implican la ejecución de procedimientos para la obtención de la información y medias porque implican hasta catorce elementos de datos referenciados y hasta cinco tablas interactuando durante su procesamiento.

En la Tabla AP.C.2 se observa la totalidad de salidas de usuario que muestran información como resultado del funcionamiento de la herramienta de evaluación.

Tabla AP.C.2 [Tabla de salidas de usuario](#)

Salidas de usuario			
Administrador	Simple	Registro de Evaluadores y Proyectos (Papel)	1
		Asignación de Proyectos a Evaluadores (Papel)	1
		Relación de Proyectos y Evaluadores (Papel)	1
Evaluador	Media	Calificación de la etapa	1
		Promedio	1
		Resumen de la evaluación	1
		Informe de la evaluación	1
Evaluador de la Calidad en Uso	Media	Proporción de la clave de acceso	1
		Calificación de la Calidad en Uso	1
Total			9

C.4 Número de peticiones de usuario

Se debe contar cada combinación única de entrada y salida en la que la entrada “en línea” definida por el usuario genera una salida inmediata también en “línea”. Las peticiones o consultas se pueden proporcionar hacia o desde otra aplicación; por ejemplo, responder a otra aplicación que pregunta por el precio de un producto se contaría como una consulta [28].

Una consulta se considera única si:

- Tiene un formato diferente de otras, ya sea en su entrada o salida.
- Tiene el mismo formato, tanto de entrada como de salida, pero requiere diferente lógica de procesamiento.

Una consulta directa en la Base de Datos es aquella que:

- Utiliza claves simples para recuperar datos específicos.
- Requiere respuesta inmediata.
- No realiza funciones de actualización (aunque se pueden efectuar cálculos).

Las consultas pueden aparecer en:

- Consulta de usuario sin actualización de archivo u otra entidad lógica.
- Archivo de transacción que sale del límite de la aplicación si está accesible al usuario “en línea”.
- Pantalla de selección de menú (todas las pantallas de menú cuentan como una consulta).
- Mensaje de información o pantalla de ayuda.

Las consultas del Sistema MECA se consideran de complejidad simple, media y compleja. Simple porque interviene solo un archivo lógico y las referencias son solo mensajes (menú y pantallas de ayuda), media porque implica el uso de hasta tres entidades y hasta diez datos referenciados (Relación de Proyectos y Evaluadores) y compleja, porque intervienen hasta ocho entidades y con un máximo de cuarenta y cinco datos referenciados (Reporte de la evaluación).

En la Tabla AP.C.3 se totalizan las peticiones o consultas de usuario que muestran información como resultado de una solicitud o entrada de datos en el Sistema MECA.

Tabla AP.C.3 Tabla de peticiones de usuario

Peticiones de usuario			
Administrador	Simple	Guía para administrar	1
		Registrar	1
	Media	Bienvenida (interfaz del administrador)	1
		Relación de proyectos y evaluadores	1
Evaluador	Simple	Guía para la evaluación	1
		Evaluar	1
	Media	Bienvenida (interfaz del evaluador)	1
		Ayuda de etapas	1
		Promedio del comité	1
	Compleja	Reporte	1
Evaluador de la Calidad en Uso	Simple	Glosario	1
		Evaluar	1
	Media	Clave de acceso	1
		Bienvenida (interfaz del eval. de la Cal. en Uso)	1
		Guía para la encuesta	1
		Elegir proyecto	1
		Calificación de la Calidad en Uso	1
		Informe de Evaluación	1
Total			18

C.5 Número de archivos lógicos

Se debe contar cada grupo lógico de datos de usuario o de información de control mantenidos dentro de los límites de la aplicación [28].

El Análisis de los Puntos de Función distingue entre dos tipos de archivos: Archivos con transacciones temporales y los Archivos con registros lógicos de datos permanentes.

Solo los almacenamientos de datos permanentes se ven como archivos lógicos. Cuando se mantienen dentro de la aplicación se clasifican como "archivos internos lógicos". Si se comparten entre aplicaciones se clasifican como interfaces y como archivos lógicos. El Sistema MECA cuenta con archivos internos lógicos. Los archivos con transacciones temporales los administra el manejador de la Base de Datos, por lo que únicamente se cuentan con archivos permanentes.

Se pueden encontrar archivos en:

- Base de Datos : uno por cada vista lógica o camino de acceso
- Archivos maestros: uno por cada grupo de claves
- Tablas mantenidas por los usuarios: métrica, código, evaluador, etc.
- Archivos de procesamiento batch
- Archivos de multimedia

El Sistema MECA proporciona Archivos clasificados como simples, ya que cuenta con hasta cuatro registros lógicos (clave compuesta formada de cuatro claves primarias) y hasta cuatro datos referenciados (claves foráneas).

La Tabla AP.C.4 muestra el número de archivos del Sistema, se contabilizan los archivos multimedios y gráficos.

Tabla AP.C.4 **Tabla de archivos lógicos**

Archivos lógicos	
Número de archivos lógicos	45

C.6 Número de interfaces externas

Se debe contar cada archivo lógico de otro grupo de datos (o información de control) que se envía fuera de los límites de la aplicación, se comparte o es recibido desde otra aplicación.

En el Sistema MECA no existen interfaces externas.

C.7 Fórmula para calcular los Puntos de Función con relación a la complejidad para cada dominio de información

En base al análisis de los valores de las actividades anteriores, se considera que el Sistema MECA es de complejidad *Media*. A continuación se calculan los Puntos de Función del Sistema MECA, mediante la siguiente fórmula:

$$PF = \text{Cuenta Total} \times [0.65 + 0.01 \times \sum F_i] \quad (1)$$

Donde: Cuenta Total es la suma de todas las entradas del Sistema MECA (véase Tabla AP.C.5).

Los valores constantes de la ecuación y los factores de peso que se aplican a las cuentas se determinan empíricamente.

Eliminado: 1

-----Salto de página-----

Figura 2.1 Tabla de métricas de Puntos de Función 1

Entradas de usuario
Registro
Estudio, Análisis e Invest
Análisis detallado y Dete
Diseño general y detallad
Prueba de aceptación
Operación y mantenimier
Documentación de sopor
Calidad en Uso
Total de entradas de usu
Salidas de usuario
Número de salidas de us
Peticiones de usuario
Número de peticiones de
Archivos lógicos
Número de archivos lógic
Interfaces externas
Número de interfaces ext

1

Eliminado: 2

Tabla AP.C.5 Computación de los Puntos de Función del Sistema MECA [36]

Computación de métricas de los puntos de función					
Parámetros de medición	Cuenta	Simple	Medio	Complejo	Cuenta total
Número de entradas de usuario	10	3	4	8	30
Número de salidas de usuario	9	4	6	7	48
Número de peticiones de usuario	18	3	4	8	70
Número de archivos	45	7	10	15	315
Número de interfaces externas	0	5	7	10	0
Cuenta total					463

Eliminado: 11



Eliminado: 2

En la fórmula (1) Puntos de Función, el valor de F_i se determina mediante la suma de una encuesta con catorce preguntas planteadas al diseñador. Estas se presentan en la Figura AP.C.1 y corresponden al nivel de complejidad del Sistema MECA, que han sido contestadas empíricamente; cada una de ellas ubican valores que oscilan de cero a cinco (factor de complejidad).

Figura AP.C.1 Computación de los Puntos de Función [36]

Computación de los puntos de función					
Evaluar cada factor en una escala de 0 a 5:					
0	1	2	3	4	5
No influencia	Incidental	Moderado	Medio	Significativo	Esencial
$F_i = 57$					
1.- ¿Requiere el sistema copias de seguridad y recuperación fiables?	_____	_____	_____	_____	5
2.- ¿Se requiere comunicación de datos?	_____	_____	_____	_____	5
3.- ¿Existen funciones de procesamiento distribuido?	_____	_____	_____	_____	5
4.- ¿Es crítico el rendimiento?	_____	_____	_____	_____	4
5.- ¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?	_____	_____	_____	_____	4
6.- ¿Requiere el sistema entrada de datos interactiva?	_____	_____	_____	_____	5
7.- ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	_____	_____	_____	_____	3
8.- ¿Se actualizan los archivos maestros de forma interactiva?	_____	_____	_____	_____	5
9.- ¿Son complejas las entradas, las salidas, los archivos o las peticiones?	_____	_____	_____	_____	2
10.- ¿Es complejo el procesamiento interno?	_____	_____	_____	_____	3
11.- ¿Se ha diseñado el código para ser reutilizable?	_____	_____	_____	_____	4
12.- ¿Están incluidas en el diseño la conversión y la instalación?	_____	_____	_____	_____	4
13.- ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	_____	_____	_____	_____	4
14.- ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	_____	_____	_____	_____	4

Eliminado: 3

Al sustituir las variables de la fórmula de los Puntos de Función (1) por valores reales aplicados al sistema, se tiene lo siguiente:

$$PF = 463 \times [0.65 + 0.01 \times 57]$$

$$PF = 463 \times [0.65 + 0.57]$$

$$PF = 463 \times [1.22] = 564.86$$

El algoritmo del Sistema MECA se clasifica como Complejo. Para los cuales, se usa una ampliación de la fórmula llamada Puntos de Función Ajustados, como se muestra en la Figura AP.C.2 en la cual el multiplicador es una constante en ésta fórmula.

Figura AP.C.2 Cálculo de los Puntos de Función Ajustados [36]

Características del programa	Puntos de función		
	Baja	Media	Alta
Número de entradas	10 x 3 = 30	0 x 4 = 0	0 x 6 = 0
Número de salidas	3 x 4 = 12	6 x 5 = 30	0 x 7 = 0
Consultas	6 x 3 = 18	11 x 4 = 44	1 x 6 = 6
Archivos lógicos internos	0 x 7 = 0	45 x 10 = 450	0 x 15 = 0
Archivos de interfaz externos	0 x 5 = 0	0 x 7 = 0	0 x 10 = 0
Total de puntos de función sin ajustar		590	
Multiplicador		x 1.15	
Total de puntos de función ajustados		678.5	

A continuación, se encuentra la estimación de las LDC (Líneas De Código) requerida para cada Punto de Función de acuerdo a la complejidad media LDC del lenguaje de Programación Orientado a Objetos.

$$TLDC = \text{Número Medio de LDC del lenguaje Java} \times \text{PF ajustado} \quad (2)$$

De acuerdo a la Figura AP.C.3, el puntaje para el lenguaje de programación Java es 30, por lo que el Total de las Líneas de Código del Sistema MECA se obtiene de la fórmula siguiente:

Eliminado: 1

Eliminado: ampliación se

Eliminado: adapta a las aplicaciones donde es alta la complejidad del algoritmo

Eliminado: 4

Eliminado: 1

Eliminado: 4

Eliminado: 5

Figura AP.C.3 Media de las Líneas De Código de acuerdo al lenguaje de programación [36]

Lenguaje de programación	LDC/PF (Media)
Ensamblador	320
C	128
Cobol	105
Fortran	105
Pascal	90
Ada	70
Lenguajes Orientados a objetos	30
Lenguajes de cuarta generación (L4G)	20
Generadores de códigos	15
Hojas de cálculo	6
Lenguajes gráficos(iconos)	4

Eliminado: ¶

¶
¶
¶
¶
¶
¶
¶

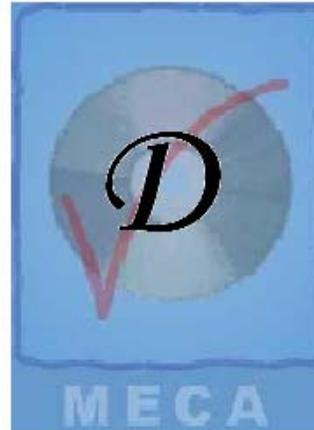
Eliminado: 5

Sustituyendo los valores de la fórmula (2) se tiene la siguiente cantidad total de Líneas De Código para el Sistema MECA.

$$TLDC = 30 \times 678.5 = 20,355$$

Apéndice

COCOMO Básico



Barry Boehm [14], en su libro clásico sobre “*Economía de la Ingeniería del Software*”, introduce una jerarquía de modelos de estimación de software con el nombre de COCOMO (Constructive Cost Model, Modelo Constructivo de Costo).

La jerarquía de modelos de Boehm está constituida por:

Modelo 1. El modelo COCOMO básico calcula el esfuerzo (y el costo) del desarrollo de software en función del tamaño del programa, expresado en las líneas estimadas de código (LDC).

Modelo 2. El modelo COCOMO intermedio calcula el esfuerzo del desarrollo de software en función del tamaño del programa y de un conjunto de “conductores de costo” que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.

Modelo 3. El modelo COCOMO avanzado incorpora todas las características de la versión intermedia y lleva acabo una evaluación del impacto de los conductores de costo en cada fase (análisis, diseño, etc.) del proceso de ingeniería del software.

Dada las condiciones de tamaño y esfuerzo, el Sistema MECA ha implementado el COCOMO Básico.

Los modelos COCOMO están definidos para tres tipos de proyectos de software. Utilizando la terminología de Boehm son:

1.-*Modo orgánico.* Proyectos de software relativamente pequeños y sencillos en los que trabajan pequeños equipos, con buena experiencia en la aplicación, sobre un conjunto de requisitos poco rígidos

2.-*Modo semiacoplado.* Proyectos de software intermedios (en tamaño y

complejidad) en los que equipos, con variados niveles de experiencia, deben satisfacer requisitos poco o medio rígidos.

3.-*Modo empotrado*: proyectos de software que deben ser desarrollados en un conjunto de hardware, software y restricciones operativas muy restringido.

El tipo de proyecto que se adapta al Sistema MECA es el *semiacoplado* por ser un software de gestión de Base de Datos.

Figura AP.D.1 COCOMO Básico.

The screenshot shows a software window titled "Cálculo del Costo del Proyecto" with the following fields and values:

- Nivel del COCOMO : BASICO
- Modo de desarrollo : Intermedio (Semibre)
- Cantidad de miles de instrucciones fuente: 9
- Esfuerzo (ESF) : 20.8 Hombres/Mes
- Tiempo de Desarrollo : 5.20
- Cant. Pers. : 4
- Cantidad de Categorías Ocupacionales : 2
- Categoría Ocupacional : 2
- Cant. de trabajadores de la Categ. Ocup. 2 : 3
- Salario de trabajadores de la Categ. Ocup. 2 : 10000
- Salario promedio : 11000
- CFT : 272477.92 pes
- Cantidad de medios técnicos: 4
- Horas de tiempo de máquina: 10
- Costo por hora de los medios técnicos: 30
- CUMT : 300 pesos
- CUMAT : 6 pesos
- Costos Directos : 272783.92 Pesos
- Costos Indirectos : 13639.19 Pesos
- COSTO TOTAL PROYECTO** : 286423.11 pesos

At the bottom of the window, there are buttons for "Cerrar" (with a green checkmark), "Ayuda" (with a question mark), and a digital clock showing "Hora : 08:43:41 p.m.".

Apéndice D

Manual de Instalación de J2SDK 1.4.1

El J2SDK es un paquete de desarrollo de la plataforma Java 2, la nueva versión resulta ser un Java más rápido y estable.

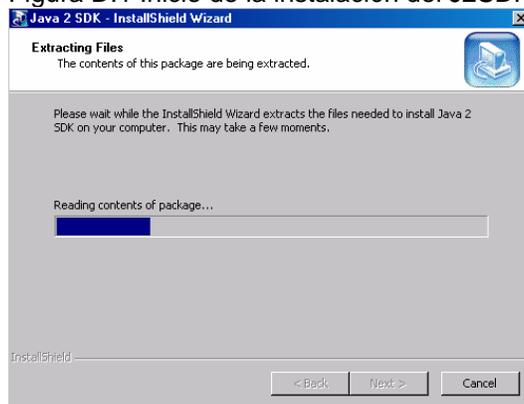
La plataforma Java 2 incluye importantes novedades y mejoras, entre las que destacan, el incremento de la velocidad de ejecución, la recolección de basura (garbage collection), nuevas librerías de clases, modelo de seguridad mucho más completo, y un mayor soporte en la creación de software.

Para instalar el J2SDK es preciso descargar primero el archivo binario, este se encuentra en Internet en la dirección:

<http://java.sun.com/j2se/1.4.1/download.html>

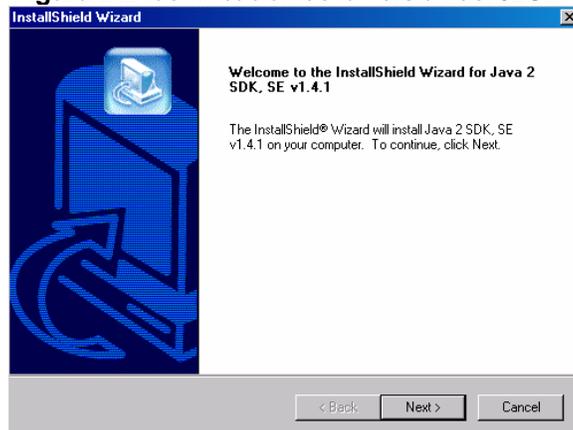
Una vez teniendo el archivo binario, se ejecuta dando inicio al proceso de la instalación, (véase la Figura D.1).

Figura D.1 Inicio de la instalación del J2SDK



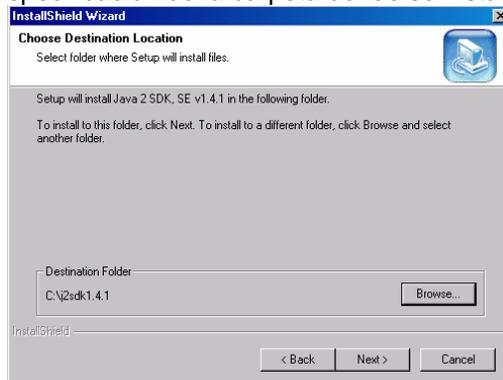
En la Figura D.2 se presenta la bienvenida e identificación de la versión del J2SDK.

Figura D.2 identificación de la versión del J2SDK



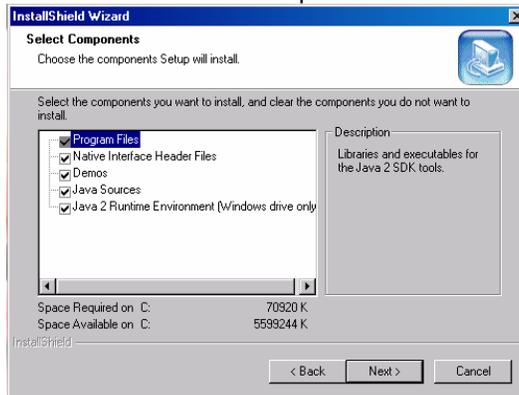
Para la correcta definición de las variables de entorno (indicadas en el manual del Usuario), es preciso indicar la instalación en la carpeta raíz, específicamente en el nombre mostrado en la Figura D.3.

Figura D.3 Especificación de la carpeta donde se instalará el J2SDK



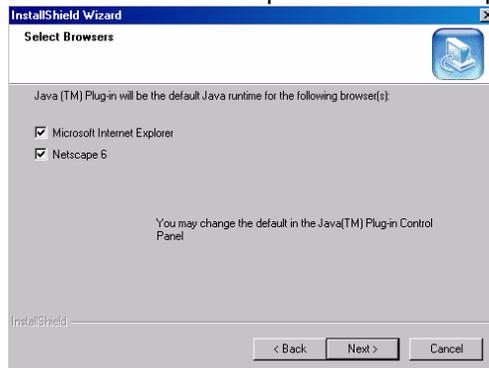
Es recomendable que se conserven los componentes que indica la instalación por defecto (véase la Figura D.4).

Figura D.4 Seleccionar los componentes de la instalación

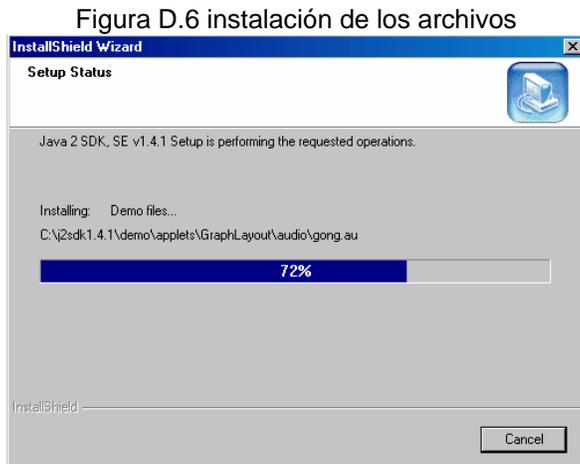


Se indica las aplicaciones con las que se podrá ejecutar el J2SDK, estos se observan en la Figura D.5.

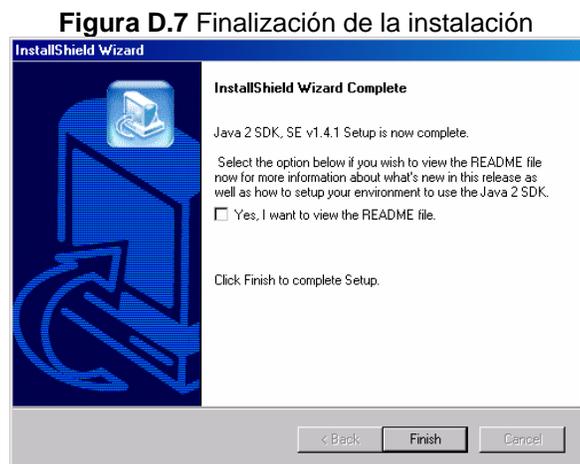
Figura D.5 seleccionar las aplicaciones de despliegue



Posteriormente da inicio a la instalación de los archivos y componentes seleccionados (véase la Figura D.6)



La instalación ha finalizado, es preciso configurar las variables de entorno y reiniciar la PC para la correcta ejecución del J2SDK.



La definición de las variables de entorno se encuentran explicadas en el tema **Requerimientos para la configuración del producto de software** en el punto *Configurar variables de ambiente* en el Manual del Usuario.

Apéndice E

Manual de Instalación de Tomcat 4.1.24

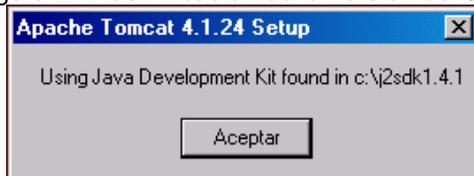
La instalación y configuración del servidor *Jakarta-Tomcat*, es la implementación oficial de referencia de la especificación Servlet 2.3 y JavaServerPages 1.2.

La explicación que sigue trata de la instalación de Tomcat en modo aislado, es decir, que es el propio Tomcat quien servirá todas las peticiones web, ya sean éstas de contenido HTML o servlets. La instalación de de Tomcat en conjunción con Apache, resulta mucho más complicada y sale del ámbito de este trabajo.

Es necesario tener instalado el JDK 1.4.1 Estándar Edition para la instalación de Tomcat. Las indicaciones que a continuación se proporcionan van dirigidas al uso de Tomcat con Windows 2000.

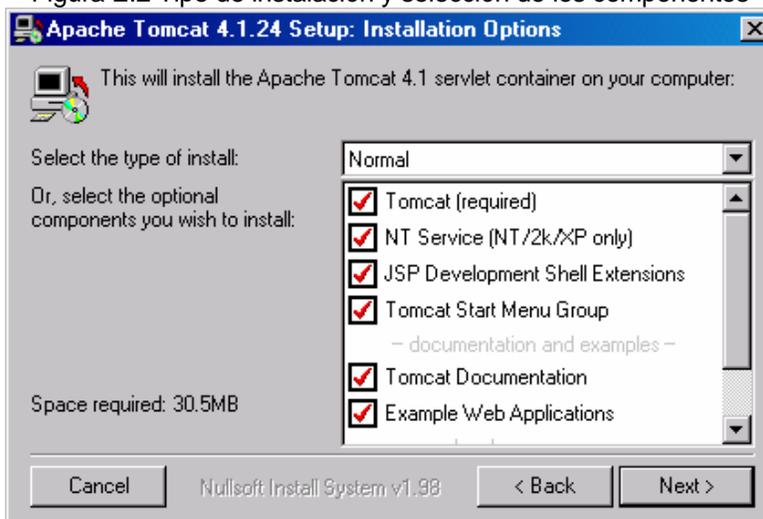
Es importante la versión binaria de Tomcat, la que se ha descargado para la instalación del sistema MECA es Tomcat 4.1.24. Al dar doble clic en el archivo binario se presenta la Figura E.1.

Figura E.1 Identificación de la versión del JDK



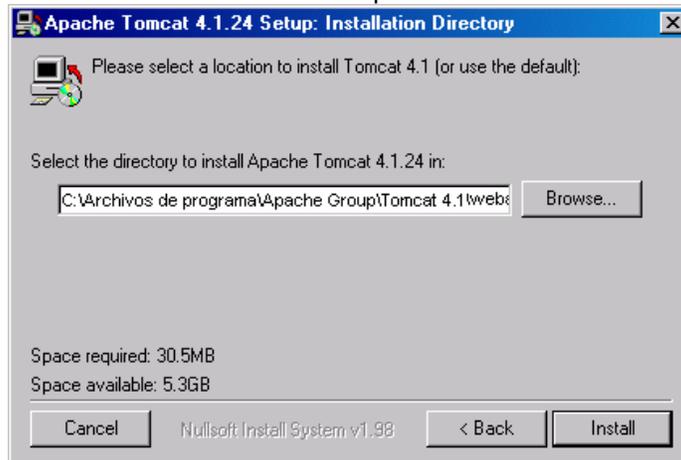
La instalación se realiza seleccionando el tipo Normal, se sugiere conservar los componentes por defecto (véase la Figura E.2).

Figura E.2 Tipo de instalación y selección de los componentes



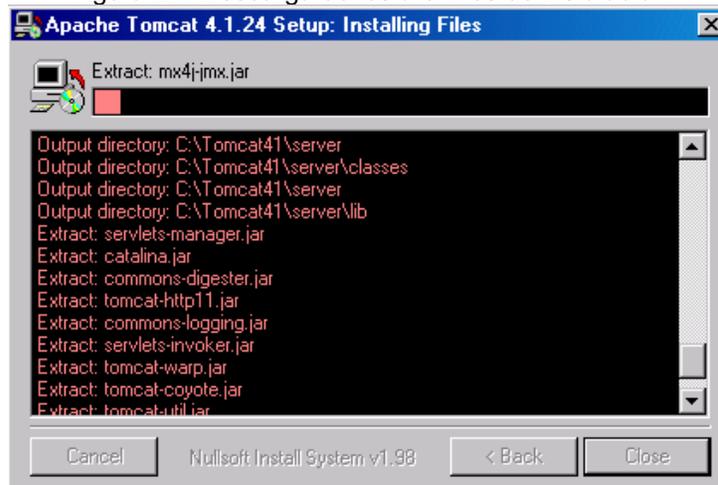
Para que la instalación de la herramienta MECA resulte exitosa, es importante que se indique Next para que guarde los archivos en las direcciones por defecto de Tomcat (véase Figura E.3).

Figura E.3 indicación del nombre de la carpeta donde ha de instalarse el Tomcat



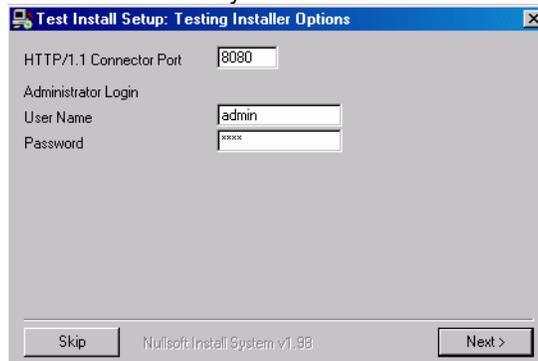
Posteriormente da inicio a la descarga de los archivos de instalación (véase la Figura E.4)

Figura E.4 Descarga de los archivos de instalación



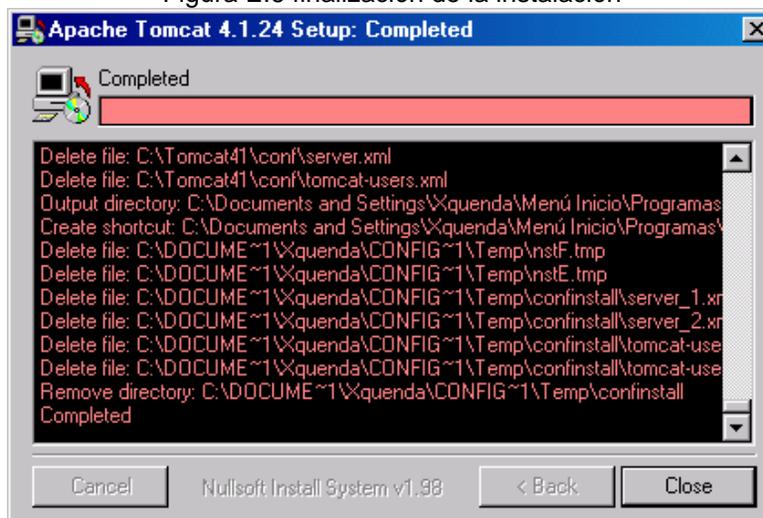
Es importante definir un nombre y clave de acceso al servidor, con la finalidad de poder registrar la aplicación en el Administrador del Servidor (véase la Figura E.5)

Figura E.5 Definición de un nombre y clave de acceso al Manejador del Servidor



Finaliza la instalación de Tomcat (véase la Figura E.6)

Figura E.6 finalización de la instalación



La definición de las variables de entorno se encuentran explicadas en el tema **Requerimientos para la configuración del producto de software** en el punto *Configurar variables de ambiente* en el Manual del Usuario.

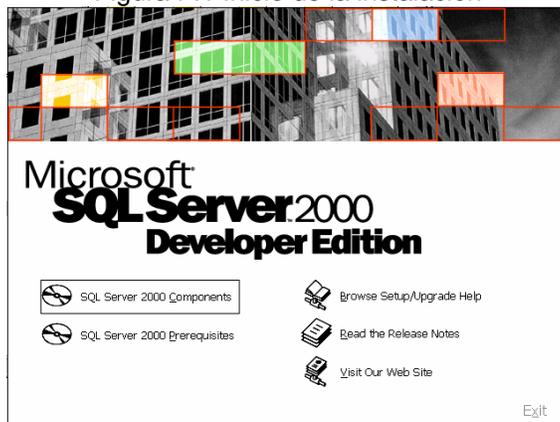
Apéndice F

Manual de Instalación de SQL Server 2000

Instalar SQL Server es muy sencillo, dispone de un asistente que se seguirá para realizar la instalación proporcionando unos datos que pedirá a lo largo del proceso. El proceso es sencillo y hay que tener en cuenta durante la instalación las opciones que se va a instalar de SQL Server.

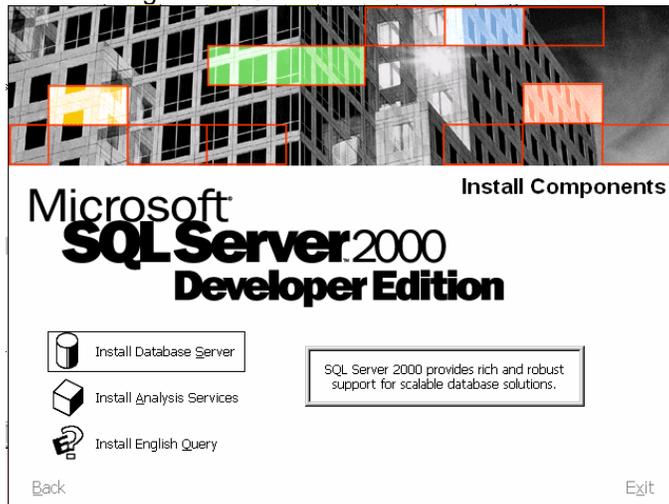
Después de ejecutarse el archivo *autorun.exe* se seleccionan las ventanas previas a la instalación: Componentes de SQL Server 2000 -> Instalar Servidor de bases de datos. La Figura F.1 muestra el inicio de la instalación.

Figura F.1 Inicio de la instalación



Elegir la instalación del servidor de Base de datos (véase Figura F.2).

Figura F.2 Servidor de Base de Datos



la forma de autenticación (personalmente utilizaría la autenticación en modo mixto) y el Nivel de intercalación opción que solo está disponible durante la instalación si se selecciona una instalación personalizada.

¿Qué es el Nivel de Intercalación? O Collation Settings en ingles, es la forma que

SQL Server tratará los caracteres de un idioma, el orden para utilizar los tipos de datos y la página de códigos. A diferencia de SQL Server 7.0 donde esta opción solo estaba disponible durante la instalación y una vez instalado ya no se podía modificar. SQL Server 2000 trata la intercalación de forma distinta permitiendo definirla a nivel de base de datos, de tabla y de columna, lo que quiere decir que se podrá tener Bases de Datos con distinto nivel de intercalación.

Con niveles de intercalaciones iguales en distintos servidores de bases de datos se puede disponer de un SQL Server compatible con otros SQL Server de la empresa que se instalaron de una determinada forma.

Por ejemplo, si se tiene instalado SQL Server con un juego de caracteres determinado como el de la imagen 9 (no distingue entre mayúsculas y minúsculas ni acentos), esto va a ser de gran utilidad a la hora de mover o restaurar Bases de Datos, si el juego de caracteres es diferente no se puede hacer. Aunque esta opción se puede aplicar después a cada Base de Datos que se cree nueva es recomendable hacerlo desde un principio y general para todo el SQL Server.

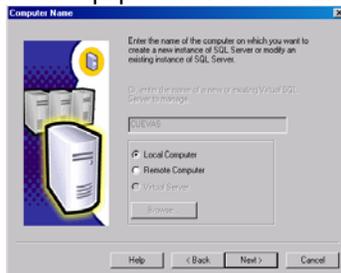
Si por alguna razón la instalación no es la misma que se tiene en otros servidores y se instala con un juego de caracteres diferente existe una solución para estos casos que permite hacer un “rebuild” a la Base de Datos Master del SQL Server y cambiar la instalación, pero este proceso se tiene que hacer con el SQL Server recién instalado porque de lo contrario se pierden las referencias a las Bases de Datos que se tenga y no las reconocerá. Para hacer el “rebuild” desde la línea de comandos escribir “rebuild” y aparecerá la imagen 10 si se pulsa en “Settings” aparece otra vez la imagen 9 donde se puede cambiar el juego de caracteres.

Para instalar el SQL Server 2000 Enterprise hay que seguir los pasos como muestran las Figuras F.3 y posteriores.

SQL Server 2000 puede instalarse de dos formas diferentes o mejor dicho pueden instalarse varias instancias en una misma maquina, esto es muy útil si por ejemplo se tiene un SQL Server 7.0 y se quiere instalar el 2000, se puede hacer y además se puede tener ejecutándose en la misma maquina los dos al mismo tiempo.

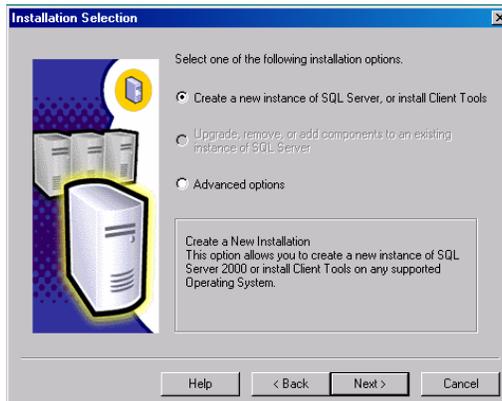
En la Figura F.3 el instalador pide el nombre del equipo y la forma de instalación, si es local o remota, si es remota se debe informar el equipo desde el cual se va a instalar.

Figura F.3 equipo en el cual se instalará



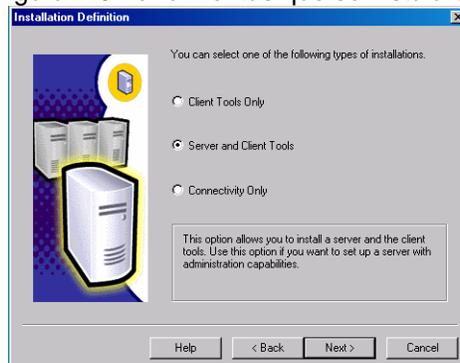
Seleccionar una opción de instalación, instalar una nueva instancia del SQL Server, instalar las partes clientes u opciones avanzadas donde se puede crear instalaciones desatendidas o reconstruir el registro del SQL Server si estuviera dañado. La opción central solo estará activa si ya se tiene un SQL Server instalado y se quiere modificar su instalación (véase la Figura F.4).

Figura F.4. crear una nueva instancia de SQL Server



Herramientas que se instalarán, es decir, si se instala solo las herramientas de cliente para tener acceso a un servidor remoto, herramientas de cliente y servidor como gestor de bases de datos o solo conectividad (véase Figura F.5).

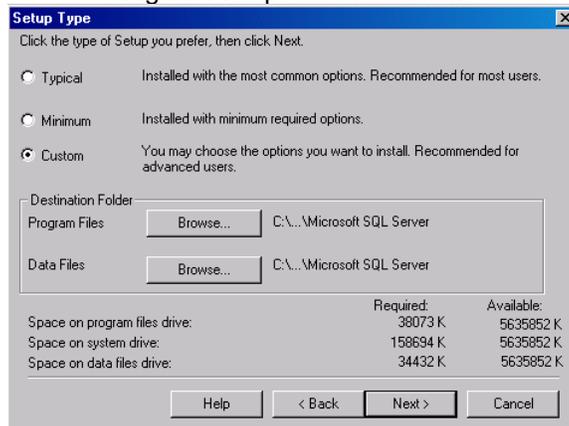
Figura F.5 herramientas que se instalarán



Nombre de la instancia que se va a crear, si es nueva instancia (no hay ningún SQL Server instalado) por defecto coge el nombre de la máquina aunque se puede cambiar, si es una segunda instalación se debe dar un nuevo nombre (elegir Default).

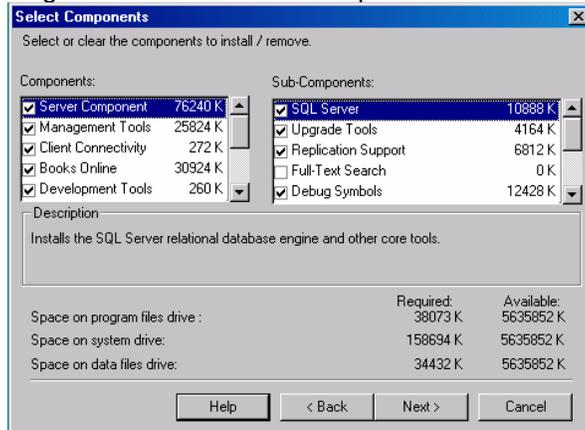
Tipo de instalación, personalmente se recomienda una instalación personalizada por lo comentado anteriormente es importante escoger el juego de caracteres (Figura F.9) para no tener problemas en futuras instalaciones y no tener que ir tocando el nivel de intercalación en cada base de datos que creamos, y elegir la ruta de los datos a una unidad con suficiente espacio en disco (véase Figura F.6).

Figura F.6 tipo de instalación



Selección de componentes a instalar, entre ellos ejemplos y ayudas (muy importante la ayuda, es el mejor manual de SQL Server). Véase la Figura F.7.

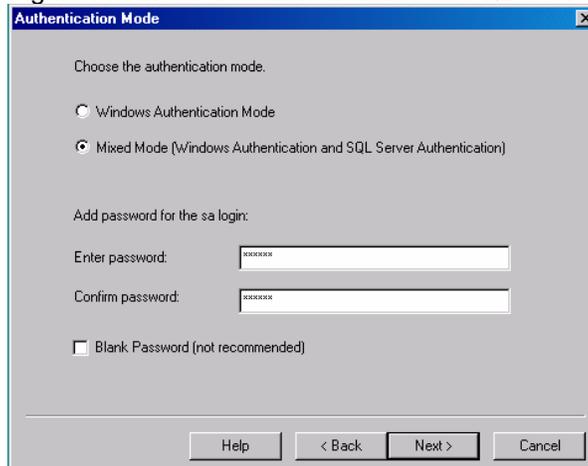
Figura F.7 Selección de componentes a instalar



Usuario que ejecutará los servicios del SQL Server y de SQL Agent, por defecto lo ejecuta el usuario administrador.

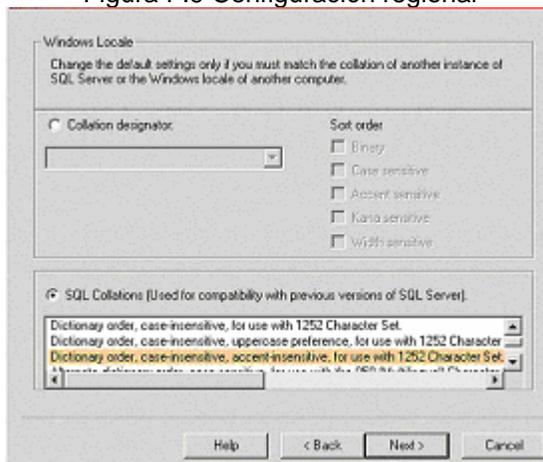
Modo de autenticación a SQL Server, se puede elegir entre autenticación windows (la autenticación se realiza por medio de usuarios pertenecientes al dominio) o modo mixto que la autenticación se realiza por medio de usuarios dados de alta en el SQL Server, si se elige esta segunda opción, no es recomendable dejar el password en blanco (véase Figura F.8).

Figura F.8 modo de autenticación de SQL Server



Configuración regional, en esta opción es donde se va a elegir la forma que nuestro SQL Server trabajará con los datos, es decir, se elegirá el nivel de intercalación. Es recomendable no dejarlo por defecto y seleccionar un nivel de intercalación apropiado a un lenguaje como por ejemplo el que muestra la imagen, en la imagen selecciona un SQL Server que no distinguirá entre mayúsculas ni entre acentos, ¿qué significa esto? Que cuando se realicen búsquedas, consultas o transacciones dará lo mismo poner “Tabla” que “tabla”. Si en todos los servidores SQL Server de la organización tienen la misma opción de nivel de intercalación, cuando se tenga la necesidad de restaurar Bases de Datos o moverlas de un SQL Server a otro será tan fácil como hacer un *restore* o copiarlas con *sp_attach_db* (véase Figura F.9)

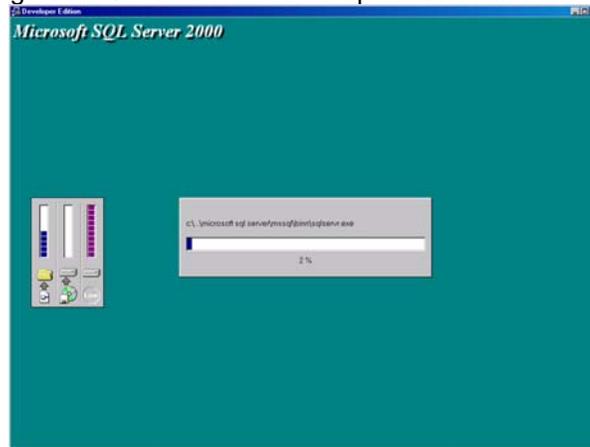
Figura F.9 Configuración regional



Por último pedirá el puerto y las bibliotecas de red que utilizará, este paso se puede dejar por defecto.

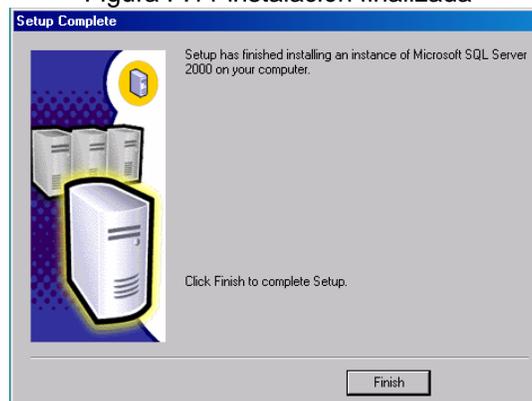
A partir de este punto SQL Server instalará las opciones seleccionadas (véase la Figura F.10).

Figura F.10 instalación de las opciones seleccionadas



Cuando finalice la instalación se tiene un SQL Server listo para trabajar (véase la Figura F.11).

Figura F.11 instalación finalizada



Apéndice G

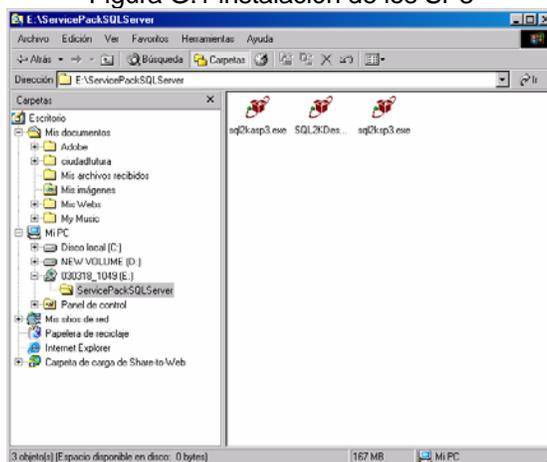
Manual de Instalación de los Service Pack SP3 de SQL Server 2000

Para los servidores de Base de Datos en ambiente de producción, microsoft recomienda implementar el Service Pack 3 (SP3) de SQL Server 2000 y las herramientas de seguridad de SQL Sever, los cuales contienen la actualización MS02-061. esta configuración ayudará a proteger el sistema contra una infección por el gusano Slammer.

La dirección de internet donde se puede descargar el SP3 es: <http://www.microsoft.com/sql/downloads/2000/sp3.asp>

Se recomienda descargar los tres archivos que presenta el SP3 y guardarlo en disco para futuras implementaciones (véase la Figura G.1).

Figura G.1 instalación de los SP3



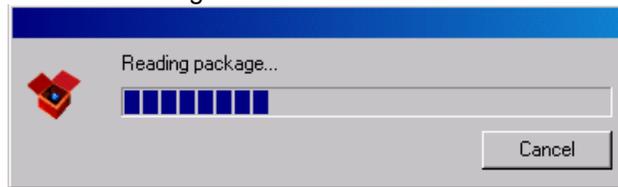
El SP3 se instalará en una carpeta indicada por defecto (véase la Figura G.2).

Figura G.2 Carpeta por defecto del SP3



Al accionar el botón *Finish* el sistema comienza a leer y descargar el SP3 (véase Figura G.3).

Figura G.3 lectura del SP3



Una vez descargado el archivo, indica la instalación correcta (véase **la Figura G.4**), y así con los siguientes dos archivos que integran el SP3.

Figura G.4 instalación correcta



Documentación adicional

Se sugiere al usuario leer el material de la lista siguiente para un mejor desempeño del Sistema MECA.

- Manual de instalación de J2SDK
- Manual de instalación de Jakarta-Tomcat
- Manual de instalación de SQL Server 2000 básico y avanzado

Anexo

Conclusiones de la evaluación

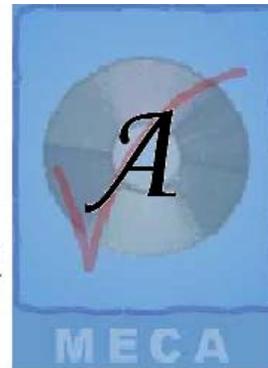


Tabla A.1 Resumen de la medición, clasificación, categorización y conclusiones de la Evaluación de la Calidad del Software

RANGO DE VALORES	NIVEL DE CLASIFICACIÓN	CATEGORÍA DE CALIDAD	CRITERIO DE EVALUACIÓN	CONCLUSIÓN
2,5 - 3	MB - MUY BUENO	SUPERIOR	SIN MODIFICACIONES	ACEPTADO
1,5 - 2,4	B - BUENO	PRIMERA	PEQUEÑAS MODIFICACIONES	ACEPTADO
0,5 - 1,4	R - REGULAR	SEGUNDA	GRANDES MODIFICACIONES	RECHAZADO
0 - 0,4	M - MALO	TERCERA	NUEVA ELABORACIÓN	RECHAZADO

CALIDAD SUPERIOR: Es la máxima categoría de calidad que puede alcanzar un software evaluado.

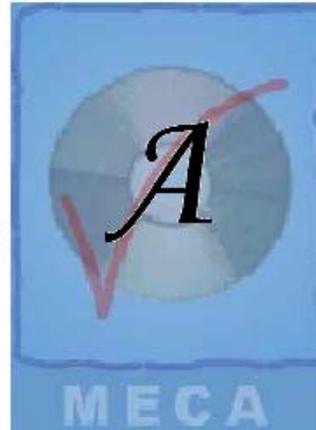
PRIMERA CALIDAD: Es la categoría alta de calidad otorgada al software que ha alcanzado un nivel muy bueno en su evaluación. Quiere decir que el producto es aceptado, aun cuando existen algunos defectos en él que pueden corregirse.

SEGUNDA CALIDAD: Es la categoría baja otorgada al software que ha alcanzado un nivel regular en su evaluación. Esta calidad es inferior a la esperada en la satisfacción de los requerimientos de los clientes. Quiere decir que el software es rechazado para que sea mejorado.

TERCERA CALIDAD: Es la calidad muy baja otorgada al software que ha alcanzado un nivel malo en su evaluación. Quiere decir que el software es rechazado, solicitando una nueva elaboración.

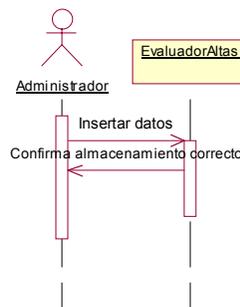
Apéndice

Diagramas de secuencia



La Figura AP.A.1 muestra el diagrama de secuencia a un alto nivel de abstracción en el que el Administrador ingresa los datos de los evaluadores, se activa una secuencia de eventos [31] que se muestran en la Tabla AP.A.1

Figura AP.A.1 Diagrama de secuencia de alto nivel de abstracción del Registro de evaluadores.



Se muestra el caso de uso *Registrar* en cual la columna izquierda contiene el comportamiento del Actor y en la derecha el del Sistema MECA.

Tabla AP.A.1 Secuencia de eventos del Registro

Acción del actor	Acción del sistema
1.- Este caso de uso comienza cuando el Actor (Administrador) llega al sistema.	2.- Pide la clave de acceso.
3.- El Administrador ingresa la clave de acceso a través del teclado.	4.- Clave Incorrecta entonces ve a paso 8 sino ve a paso 5.
	5.- Ofrece las opciones del menú para el Administrador.
6.- Selecciona la opción registro: a) Si se registra el Proyecto, véase la Sección Proyecto. b) Si se registra el Evaluador, véase la Sección Evaluador c) Si se elige la opción Asignación de proyectos, véase la Sección Asignación d) Si se elige la opción Relación de Proyectos y Evaluadores, véase la	7.- Devuelve la clave del evaluador registrado o en el caso de la opción <i>Asignación de proyectos</i> , actualiza los datos del sistema relacionando al evaluador el proyecto correspondiente. ve a paso 13

Sección Relación	
	8.- Muestra el formato de registro del Administrador
9.- Escribe los datos que pide el formato y los envía.	10.- Registra los datos recibidos y proporciona una clave de acceso al Administrador.
11.- Toma nota de la clave proporcionada	12.- ve a paso 5
13.- Anota la clave de registro y se marcha.	

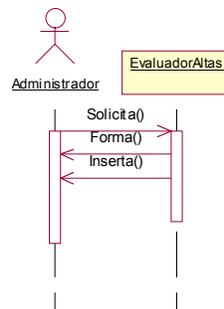
Las secciones independientes que se muestran a continuación constituyen los submódulos del Registro. La Tabla AP.A.2 Sección Evaluador muestra la secuencia de eventos para la sección Evaluador.

Tabla AP.A.2 Sección Evaluador

Acción del actor	Acción del sistema
1.- El Administrador introduce los datos del Evaluador que se le solicite.	2.- Verifica que los datos estén completos.
	3.- Guarda los datos en los archivos del sistema

La Figura AP.A.2 muestra el diagrama de secuencia a un bajo nivel de abstracción, en el que el Administrador solicita el ingreso de los datos de un evaluador, la clase EvaluadorAltas ejecuta el método *forma()* que contiene todos los datos requeridos del evaluador y hace referencia al método *inserta()* que almacena estos datos en la tabla Evaluador.

Figura AP.A.2 Diagrama de secuencia de bajo nivel de abstracción del Registro de Evaluadores.



La Figura AP.A.3 muestra el diagrama de secuencia del Registro de proyectos, en el cual el Administrador solicita el ingreso de los datos del Proyecto, en la Tabla AP.A.3 se encuentra el flujo de eventos del Registro del proyecto.

Figura AP.A.3 Diagrama de secuencia del Registro de Proyectos

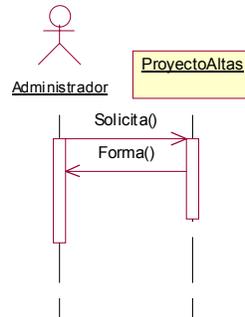


Tabla AP.A.3 Sección Proyecto

Acción del actor	Acción del sistema
1.- El Administrador introduce los datos del Proyecto que se le solicite.	2.- Verifica que los datos estén completos.
	3.- Guarda los datos en los archivos del sistema.

La Figura AP.A.4 muestra los métodos de la clase *ProyectoAltas*, particularmente el método *forma()* muestra el formulario con los datos que Administrador deberá cubrir para que el sistema administre correctamente sus datos en la evaluación, el método *Inserta()* de esta misma clase efectúa la inserción en la tabla Proyecto.

Figura AP.A.4 Diagrama de secuencia de bajo nivel del Registro de Proyectos.



La Figura AP.A.5 muestra la secuencia que sigue el sistema ante la asignación de proyectos a evaluadores, la Tabla AP.A.4 muestra el flujo de los eventos del diagrama.

Figura AP.A.5 Diagrama de secuencia de la Asignación de Proyectos.

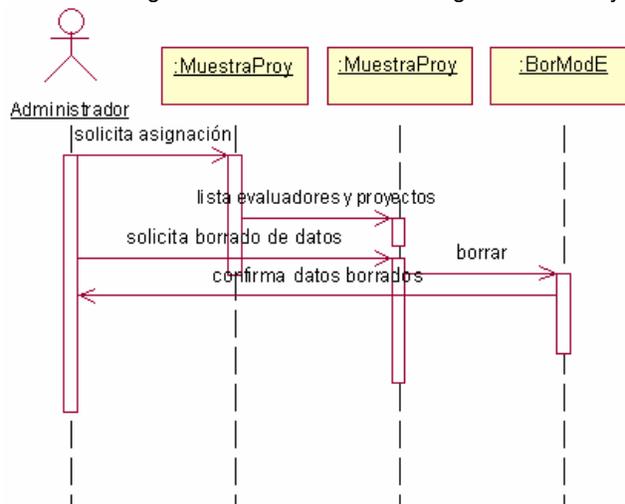
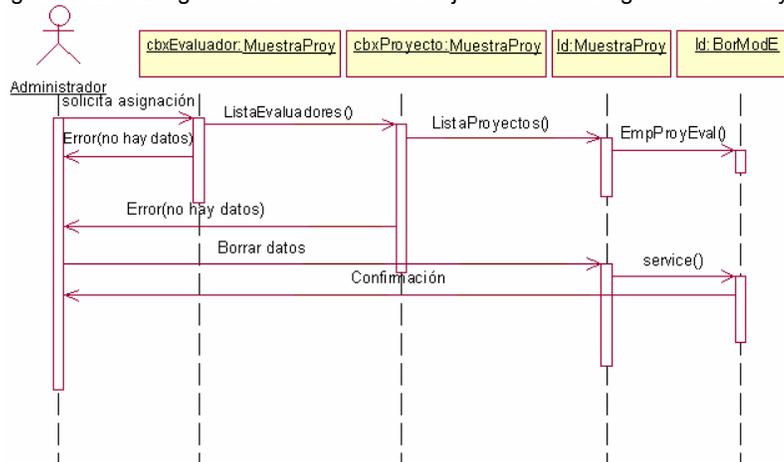


Tabla AP.A.4 Asignación de Proyecto

Acción del actor	Acción del sistema
1.- El Administrador solicita la Asignación de proyectos.	2.- Muestra la lista de Proyectos y Evaluadores de la Empresa.
3.- Elige el Proyecto, el evaluador y el papel que desempeñará el Evaluador en el Proyecto.	4.- Verifica que los datos estén correctamente asignados (no admite la asignación de dos Presidentes en la evaluación del mismo Proyecto y la asignación de Presidente a un Evaluador de la Calidad en Uso).
	5.- Guarda los datos en los archivos del sistema.
6.- Verifica que los datos hayan sido correctamente asignados y pasa a 9 sino solicita el borrado de datos incorrectos.	7.- Borra los datos elegidos de la Base de Datos MECA
8.- Pasa a 4	
9.- Se marcha	

La Figura AP.A.6 muestra la ejecución de la clase *MuestraProy* cuyos métodos hacen posible la asignación de evaluadores y Proyectos, *cbxEvaluador* hace posible el despliegue de la lista de los Evaluadores registrados en el sistema, *cbxProyecto* muestra la lista de Proyectos a evaluar, el Administrador elegirá un elemento de cada lista mostrada, el sistema a su vez registrará esta selección y mostrará mediante *EmpProyEval()* una lista con los elementos elegidos. En caso de ocurrir una equivocación durante la asignación, el Administrador podrá borrar la selección de la lista.

Figura AP.A.6 Diagrama de secuencia de bajo nivel de la Asignación de Proyectos.



En la Figura AP.A.7 muestra el diagrama de secuencia de bajo nivel de la evaluación. En el que un usuario pide el servicio mediante la clase *solicita*. Esta clase presenta la interfaz al Evaluador solicitante; así mismo, envía parámetros (la clave del Evaluador, la clave de la empresa y la clave del Administrador) a la clase *EligeProy*.

La clase *EligeProy* muestra la lista de los Proyectos asignados al Evaluador; quien elige su proyecto. Después de esta elección, se ejecuta la clase *Etapas*.

La clase *Etapas* contiene los métodos que muestran en pantalla el formato de la etapa elegida por el Evaluador. Inicia la inserción de datos al formato; al almacenar estos datos, se ejecuta la clase *Evaluación*.

La clase *Evaluación* almacena el avance, iniciando la asignación de valores a los atributos, subcaracterísticas, características y etapa. Este proceso activa la clase *Retorno* para conocer la clave del atributo.

Evaluación solicita a la clase *Met_Param* los parámetros del atributo; con la finalidad de obtener el valor de este.

Con los parámetros, *Evaluacion* pide a la clase *Metricas* el Método o fórmula correspondiente al atributo, almacena el resultado en su ubicación de la Base de Datos; esta operación se repite para todos los atributos de la subcaracterística, característica y etapa.

La clase *Evaluacion* devuelve a *Etapa* el promedio de la etapa. Esta clase, a su vez, muestra al Evaluador la posibilidad de imprimir el reporte.

La clase *Reporte* muestra el informe de la evaluación, también muestra el promedio del comité, (solo a los usuarios reconocidos como Presidente del comité evaluador); en tal caso, *Reporte* se comunica con la clase *ClaveDeAcceso*.

ClaveDeAcceso autentifica al usuario y se comunica con la clase *Promedio*. *Promedio* muestra el promedio correspondiente a la etapa.

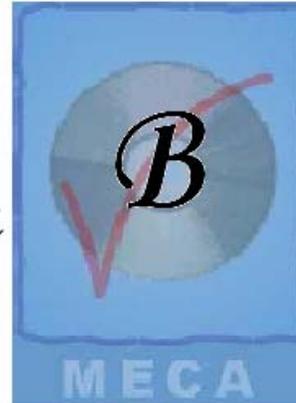
El otro informe de evaluación, lo presenta la clase *ReporteEval*, este es el reporte final.

El diagrama de secuencias muestra la parte de la evaluación en la que se desempeña el método *TwoInputA* de la clase *Métricas*; este método es usado frecuentemente en la evaluación para obtener el valor del atributo.

Una vez obtenido el valor del atributo, el método *TwoInputA* referencia al método *Evaluation* para darle un valor al atributo en el rango de 0 a 3. En esta fase se solicitan y aplican los métodos correspondientes.

Apéndice

Diagramas de colaboración



Como se muestra en la Figura AP.B.1 El diagrama de colaboración se construye a partir del objeto *MaxIdEvaluador* que participa en la colaboración como nodo del grafo, este objeto realiza el método *Forma()* que muestra el formulario, *Inserta()* almacena en la tabla *Evaluador* los datos involucrados e *IsNull()* como validación de los datos.

Figura AP.B.1 Diagrama de colaboración del Registro del Evaluador



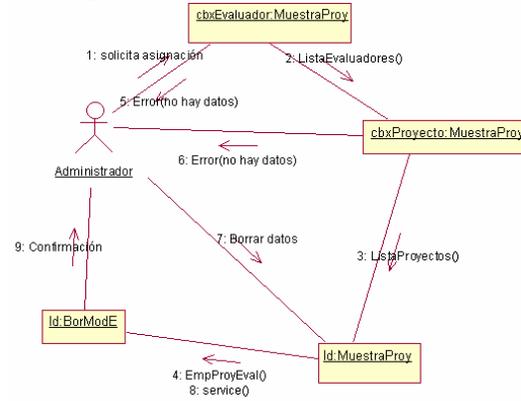
La Figura AP.B.2 Muestra la interacción entre los objetos y métodos para el registro del proyecto, tiene muchas similitudes en la lógica del registro de Evaluadores.

Figura AP.B.2 Diagrama de colaboración del Registro del Proyecto



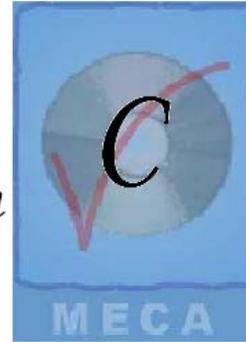
La Figura AP.B.3 muestra los objetos que intervienen en la asignación de proyectos a evaluadores, el Administrador solicita la asignación, el sistema muestra mediante el atributo *cbxEvaluador* y *cxpProyecto* de la clase *MuestraProy* las listas de los Evaluadores y Proyectos, el sistema muestra mediante el método *EmpProyEval()* los elementos elegidos por el Administrador, de haberse elegido erróneamente, la clase *BorModE* conectada a esta clase efectúa el borrado de los datos.

Figura AP.B.3 Diagrama de colaboración de la Asignación de Proyectos



Apéndice

Cálculo de los Puntos de Función



C.1 Características del dominio de información

Al analizar el dominio de la información [36] del Sistema MECA se puede partir de tres puntos de vista sobre los datos y el control:

- **Contenido de la información y sus relaciones:** se analizan los atributos que conforman cada etapa de la evaluación del Sistema MECA y cómo se interrelacionan.
- **Flujo de la información y sus relaciones:** se refiere a la forma en que cambian los datos introducidos por el usuario, relacionándose posteriormente con la biblioteca de métricas de MECA, para producir respuestas tales como el nivel de la calidad en la etapa evaluada.
- **Estructura de la información:** requiere del análisis del tipo de estructura de datos que se aplicará en el diseño del Sistema MECA, tal es el caso del uso de tablas que administran el modelo de evaluación; o bien, típicamente en la Administración de la Base de Datos, que contiene el resultado de la evaluación.

A continuación, se describe las características del dominio de información del Sistema MECA con las actividades que se realizan para obtener el grado de complejidad de la herramienta; estas actividades consisten en medir el número de líneas de código del software, las orientadas a la función que fueron propuestas por primera vez por Albretch [36]. El método para hallar los Puntos de Función consiste en contabilizar los elementos del software tales como las entradas, las salidas, archivos lógicos e interfaces externas y aplicarles la fórmula de los Puntos de Función. Esta fórmula contabiliza los elementos del software con los valores que determinan su complejidad, en él intervienen constantes definidas [36].

Eliminado: métricas

Eliminado: orientadas al tamaño,

Eliminado: que se miden a través d

Eliminado: y

Eliminado: las métricas

Eliminado: a través de los Puntos de Función

Eliminado: ,

Eliminado: el

Eliminado: a

Eliminado: métricas

C.2 Número de entradas de usuario

Una de las medidas del proyecto de software para calcular los Puntos de Función consiste en sumar el número de entradas del usuario. Se debe contar cada dato único de usuario o entrada de control que se introduce en los límites de la aplicación y actualiza un archivo lógico interno, conjunto de datos, tabla o dato

independiente. Esto incluye archivos de entrada y transacciones recibidas de otras aplicaciones (el Sistema MECA no cuenta con este tipo de entrada).

Una entrada se considera única si:

- Tiene un formato diferente
- Tiene el mismo formato que otra entrada pero requiere una lógica diferente de procesamiento, o se modifica un archivo interno lógico diferente.
- Las funciones de actualización (insertar, modificar, eliminar) requiere diferente lógica de procesamiento; por lo tanto, se cuentan por separado.

Las entradas se clasifican en simples, medias y complejas [36]. Las entradas del Sistema MECA, se consideran simples porque no implican el uso de más de dos archivo lógicos y mayor de cuatro datos referenciados [28].

En la Tabla AP.C.1 se presenta el total de las entradas simples de usuarios que proporcionan datos para el funcionamiento de la herramienta de evaluación.

Tabla AP.C.1 [Tabla de entradas de usuario](#)

Entradas de usuario		
Administrador	Clave de acceso	1
	Registro	1
	Alta a Evaluadores y Proyectos	1
	Asignación de Proyectos	1
	Eliminación por asignación errónea	1
Evaluador	Atributos comunes	1
	Formato de evaluación	1
	Promedio del comité	1
Evaluador de la Calidad en Uso	Formato de la encuesta	1
	Comentarios	1
Total		10

C.3 Número de salidas de usuario

Se debe contar cada dato único o salida de control generado proceduralmente. Esto incluye informes y mensajes a otras aplicaciones y usuarios.

Una salida se considera única si:

- Tiene formato diferente.
- Tiene el mismo formato que otra salida pero requiere diferente lógica de procesamiento.

Además de las pantallas y los listados (papel o pantalla), también pueden ser salidas [28]:

- Archivos de transacción enviado a otra aplicación.
- Facturas.
- Cheques.
- Transacciones automáticas.
- Mensajes al usuario.
- Gráficos.
- Archivos back-up, etc.

No se deben contar como salidas:

- Cabeceras de columna, títulos, número de página.
- Mensajes individuales (información, confirmación o respuestas a consultas de error).
- Salida en igual formato y lógica que ya se han contado para otro soporte.

Al igual que en las entradas, las salidas de usuario se clasifican en simples, medias y complejas [36]. Las salidas del Sistema MECA se consideran simples y medias. Simples porque no implican la ejecución de procedimientos para la obtención de la información y medias porque implican hasta catorce elementos de datos referenciados y hasta cinco tablas interactuando durante su procesamiento.

En la Tabla AP.C.2 se observa la totalidad de salidas de usuario que muestran información como resultado del funcionamiento de la herramienta de evaluación.

Tabla AP.C.2 [Tabla de salidas de usuario](#)

Salidas de usuario			
Administrador	Simple	Registro de Evaluadores y Proyectos (Papel)	1
		Asignación de Proyectos a Evaluadores (Papel)	1
		Relación de Proyectos y Evaluadores (Papel)	1
Evaluador	Media	Calificación de la etapa	1
		Promedio	1
		Resumen de la evaluación	1
		Informe de la evaluación	1
Evaluador de la Calidad en Uso	Media	Proporción de la clave de acceso	1
		Calificación de la Calidad en Uso	1
Total			9

C.4 Número de peticiones de usuario

Se debe contar cada combinación única de entrada y salida en la que la entrada “en línea” definida por el usuario genera una salida inmediata también en “línea”. Las peticiones o consultas se pueden proporcionar hacia o desde otra aplicación; por ejemplo, responder a otra aplicación que pregunta por el precio de un producto se contaría como una consulta [28].

Una consulta se considera única si:

- Tiene un formato diferente de otras, ya sea en su entrada o salida.
- Tiene el mismo formato, tanto de entrada como de salida, pero requiere diferente lógica de procesamiento.

Una consulta directa en la Base de Datos es aquella que:

- Utiliza claves simples para recuperar datos específicos.
- Requiere respuesta inmediata.
- No realiza funciones de actualización (aunque se pueden efectuar cálculos).

Las consultas pueden aparecer en:

- Consulta de usuario sin actualización de archivo u otra entidad lógica.
- Archivo de transacción que sale del límite de la aplicación si está accesible al usuario “en línea”.
- Pantalla de selección de menú (todas las pantallas de menú cuentan como una consulta).
- Mensaje de información o pantalla de ayuda.

Las consultas del Sistema MECA se consideran de complejidad simple, media y compleja. Simple porque interviene solo un archivo lógico y las referencias son solo mensajes (menú y pantallas de ayuda), media porque implica el uso de hasta tres entidades y hasta diez datos referenciados (Relación de Proyectos y Evaluadores) y compleja, porque intervienen hasta ocho entidades y con un máximo de cuarenta y cinco datos referenciados (Reporte de la evaluación).

En la Tabla AP.C.3 se totalizan las peticiones o consultas de usuario que muestran información como resultado de una solicitud o entrada de datos en el Sistema MECA.

Tabla AP.C.3 Tabla de peticiones de usuario

Peticiones de usuario			
Administrador	Simple	Guía para administrar	1
		Registrar	1
	Media	Bienvenida (interfaz del administrador)	1
		Relación de proyectos y evaluadores	1
Evaluador	Simple	Guía para la evaluación	1
		Evaluar	1
	Media	Bienvenida (interfaz del evaluador)	1
		Ayuda de etapas	1
		Promedio del comité	1
	Compleja	Reporte	1
Evaluador de la Calidad en Uso	Simple	Glosario	1
		Evaluar	1
	Media	Clave de acceso	1
		Bienvenida (interfaz del eval. de la Cal. en Uso)	1
		Guía para la encuesta	1
		Elegir proyecto	1
		Calificación de la Calidad en Uso	1
		Informe de Evaluación	1
Total			18

C.5 Número de archivos lógicos

Se debe contar cada grupo lógico de datos de usuario o de información de control mantenidos dentro de los límites de la aplicación [28].

El Análisis de los Puntos de Función distingue entre dos tipos de archivos: Archivos con transacciones temporales y los Archivos con registros lógicos de datos permanentes.

Solo los almacenamientos de datos permanentes se ven como archivos lógicos. Cuando se mantienen dentro de la aplicación se clasifican como "archivos internos lógicos". Si se comparten entre aplicaciones se clasifican como interfaces y como archivos lógicos. El Sistema MECA cuenta con archivos internos lógicos. Los archivos con transacciones temporales los administra el manejador de la Base de Datos, por lo que únicamente se cuentan con archivos permanentes.

Se pueden encontrar archivos en:

- Base de Datos : uno por cada vista lógica o camino de acceso
- Archivos maestros: uno por cada grupo de claves
- Tablas mantenidas por los usuarios: métrica, código, evaluador, etc.
- Archivos de procesamiento batch
- Archivos de multimedia

El Sistema MECA proporciona Archivos clasificados como simples, ya que cuenta con hasta cuatro registros lógicos (clave compuesta formada de cuatro claves primarias) y hasta cuatro datos referenciados (claves foráneas).

La Tabla AP.C.4 muestra el número de archivos del Sistema, se contabilizan los archivos multimedios y gráficos.

Tabla AP.C.4 **Tabla de archivos lógicos**

Archivos lógicos	
Número de archivos lógicos	45

C.6 Número de interfaces externas

Se debe contar cada archivo lógico de otro grupo de datos (o información de control) que se envía fuera de los límites de la aplicación, se comparte o es recibido desde otra aplicación.

En el Sistema MECA no existen interfaces externas.

C.7 Fórmula para calcular los Puntos de Función con relación a la complejidad para cada dominio de información

En base al análisis de los valores de las actividades anteriores, se considera que el Sistema MECA es de complejidad *Media*. A continuación se calculan los Puntos de Función del Sistema MECA, mediante la siguiente fórmula:

$$PF = \text{Cuenta Total} \times [0.65 + 0.01 \times \sum F_i] \quad (1)$$

Donde: Cuenta Total es la suma de todas las entradas del Sistema MECA (véase Tabla AP.C.5).

Los valores constantes de la ecuación y los factores de peso que se aplican a las cuentas se determinan empíricamente.

Eliminado: 1

-----Salto de página-----

Figura 2.1 Tabla de métricas de Puntos de Función 1

Entradas de usuario
Registro
Estudio, Análisis e Invest
Análisis detallado y Dete
Diseño general y detallad
Prueba de aceptación
Operación y mantenimier
Documentación de sopor
Calidad en Uso
Total de entradas de usu
Salidas de usuario
Número de salidas de us
Peticiones de usuario
Número de peticiones de
Archivos lógicos
Número de archivos lógic
Interfaces externas
Número de interfaces ext

1

Eliminado: 2

Tabla AP.C.5 Computación de los Puntos de Función del Sistema MECA [36]

Computación de métricas de los puntos de función					
Parámetros de medición	Cuenta	Simple	Medio	Complejo	Cuenta total
Número de entradas de usuario	10	3	4	8	30
Número de salidas de usuario	9	4	6	7	48
Número de peticiones de usuario	18	3	4	8	70
Número de archivos	45	7	10	15	315
Número de interfaces externas	0	5	7	10	0
Cuenta total					463

Eliminado: 11



Eliminado: 2

En la fórmula (1) Puntos de Función, el valor de F_i se determina mediante la suma de una encuesta con catorce preguntas planteadas al diseñador. Estas se presentan en la Figura AP.C.1 y corresponden al nivel de complejidad del Sistema MECA, que han sido contestadas empíricamente; cada una de ellas ubican valores que oscilan de cero a cinco (factor de complejidad).

Figura AP.C.1 Computación de los Puntos de Función [36]

Computación de los puntos de función					
Evaluar cada factor en una escala de 0 a 5:					
0	1	2	3	4	5
No influencia	Incidental	Moderado	Medio	Significativo	Esencial
$F_i = 57$					
1.- ¿Requiere el sistema copias de seguridad y recuperación fiables?	_____	_____	_____	_____	5
2.- ¿Se requiere comunicación de datos?	_____	_____	_____	_____	5
3.- ¿Existen funciones de procesamiento distribuido?	_____	_____	_____	_____	5
4.- ¿Es crítico el rendimiento?	_____	_____	_____	_____	4
5.- ¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?	_____	_____	_____	_____	4
6.- ¿Requiere el sistema entrada de datos interactiva?	_____	_____	_____	_____	5
7.- ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	_____	_____	_____	_____	3
8.- ¿Se actualizan los archivos maestros de forma interactiva?	_____	_____	_____	_____	5
9.- ¿Son complejas las entradas, las salidas, los archivos o las peticiones?	_____	_____	_____	_____	2
10.- ¿Es complejo el procesamiento interno?	_____	_____	_____	_____	3
11.- ¿Se ha diseñado el código para ser reutilizable?	_____	_____	_____	_____	4
12.- ¿Están incluidas en el diseño la conversión y la instalación?	_____	_____	_____	_____	4
13.- ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	_____	_____	_____	_____	4
14.- ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	_____	_____	_____	_____	4

Eliminado: 3

Al sustituir las variables de la fórmula de los Puntos de Función (1) por valores reales aplicados al sistema, se tiene lo siguiente:

$$PF = 463 \times [0.65 + 0.01 \times 57]$$

$$PF = 463 \times [0.65 + 0.57]$$

$$PF = 463 \times [1.22] = 564.86$$

El algoritmo del Sistema MECA se clasifica como Complejo. Para los cuales, se usa una ampliación de la fórmula llamada Puntos de Función Ajustados, como se muestra en la Figura AP.C.2 en la cual el multiplicador es una constante en ésta fórmula.

Figura AP.C.2 Cálculo de los Puntos de Función Ajustados [36]

Características del programa	Puntos de función		
	Baja	Media	Alta
Número de entradas	10 x 3 = 30	0 x 4 = 0	0 x 6 = 0
Número de salidas	3 x 4 = 12	6 x 5 = 30	0 x 7 = 0
Consultas	6 x 3 = 18	11 x 4 = 44	1 x 6 = 6
Archivos lógicos internos	0 x 7 = 0	45 x 10 = 450	0 x 15 = 0
Archivos de interfaz externos	0 x 5 = 0	0 x 7 = 0	0 x 10 = 0
Total de puntos de función sin ajustar		590	
Multiplicador		x 1.15	
Total de puntos de función ajustados		678.5	

A continuación, se encuentra la estimación de las LDC (Líneas De Código) requerida para cada Punto de Función de acuerdo a la complejidad media LDC del lenguaje de Programación Orientado a Objetos.

$$TLDC = \text{Número Medio de LDC del lenguaje Java} \times \text{PF ajustado} \quad (2)$$

De acuerdo a la Figura AP.C.3, el puntaje para el lenguaje de programación Java es 30, por lo que el Total de las Líneas de Código del Sistema MECA se obtiene de la fórmula siguiente:

Eliminado: 1

Eliminado: ampliación se

Eliminado: adapta a las aplicaciones donde es alta la complejidad del algoritmo

Eliminado: 4

Eliminado: 1

Eliminado: 4

Eliminado: 5

Figura AP.C.3 Media de las Líneas De Código de acuerdo al lenguaje de programación [36]

Lenguaje de programación	LDC/PF (Media)
Ensamblador	320
C	128
Cobol	105
Fortran	105
Pascal	90
Ada	70
Lenguajes Orientados a objetos	30
Lenguajes de cuarta generación (L4G)	20
Generadores de códigos	15
Hojas de cálculo	6
Lenguajes gráficos(iconos)	4

Eliminado: ¶

¶
¶
¶
¶
¶
¶
¶

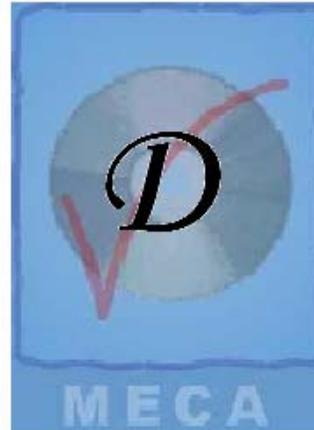
Eliminado: 5

Sustituyendo los valores de la fórmula (2) se tiene la siguiente cantidad total de Líneas De Código para el Sistema MECA.

$$TLDC = 30 \times 678.5 = 20,355$$

Apéndice

COCOMO Básico



Barry Boehm [14], en su libro clásico sobre “*Economía de la Ingeniería del Software*”, introduce una jerarquía de modelos de estimación de software con el nombre de COCOMO (Constructive Cost Model, Modelo Constructivo de Costo).

La jerarquía de modelos de Boehm está constituida por:

Modelo 1. El modelo COCOMO básico calcula el esfuerzo (y el costo) del desarrollo de software en función del tamaño del programa, expresado en las líneas estimadas de código (LDC).

Modelo 2. El modelo COCOMO intermedio calcula el esfuerzo del desarrollo de software en función del tamaño del programa y de un conjunto de “conductores de costo” que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.

Modelo 3. El modelo COCOMO avanzado incorpora todas las características de la versión intermedia y lleva acabo una evaluación del impacto de los conductores de costo en cada fase (análisis, diseño, etc.) del proceso de ingeniería del software.

Dada las condiciones de tamaño y esfuerzo, el Sistema MECA ha implementado el COCOMO Básico.

Los modelos COCOMO están definidos para tres tipos de proyectos de software. Utilizando la terminología de Boehm son:

1.-*Modo orgánico.* Proyectos de software relativamente pequeños y sencillos en los que trabajan pequeños equipos, con buena experiencia en la aplicación, sobre un conjunto de requisitos poco rígidos

2.-*Modo semiacoplado.* Proyectos de software intermedios (en tamaño y

complejidad) en los que equipos, con variados niveles de experiencia, deben satisfacer requisitos poco o medio rígidlos.

3.-*Modo empotrado*: proyectos de software que deben ser desarrollados en un conjunto de hardware, software y restricciones operativas muy restringido.

El tipo de proyecto que se adapta al Sistema MECA es el *semiacoplado* por ser un software de gestión de Base de Datos.

Figura AP.D.1 COCOMO Básico.

The screenshot shows a software window titled "Cálculo del Costo del Proyecto" with the following data:

Parameter	Value
Nivel del COCOMO	BASICO
Modo de desarrollo	Intermedio (Semibre)
Cantidad de miles de instrucciones fuente	9
Esfuerzo (ESF) Hombres/Mes	20.8
Tiempo de Desarrollo	5.20
Cant. Pers.	4
Cantidad de Categorías Ocupacionales	2
Categoría Ocupacional	2
Cant. de trabajadores de la Categ. Ocup. 2	3
Salario de trabajadores de la Categ. Ocup. 2	10000
Salario promedio	11000
CFT	272477.92 pes
Cantidad de medios técnicos	4
Horas de tiempo de máquina	10
Costo por hora de los medios técnicos	30
CUMT	300 pesos
CUMAT	6 pesos
Costos Directos	272783.92 Pesos
Costos Indirectos	13639.19 Pesos
COSTO TOTAL PROYECTO	286423.11 pesos

At the bottom of the window, there are buttons for "Cerrar" (with a green checkmark), "Ayuda" (with a question mark), and a digital clock showing "Hora : 08:43:41 p.m."

Apéndice D

Manual de Instalación de J2SDK 1.4.1

El J2SDK es un paquete de desarrollo de la plataforma Java 2, la nueva versión resulta ser un Java más rápido y estable.

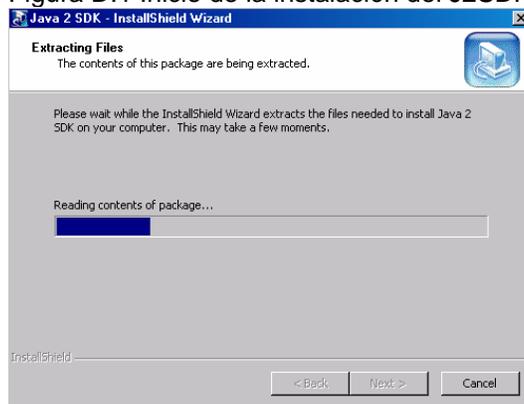
La plataforma Java 2 incluye importantes novedades y mejoras, entre las que destacan, el incremento de la velocidad de ejecución, la recolección de basura (garbage collection), nuevas librerías de clases, modelo de seguridad mucho más completo, y un mayor soporte en la creación de software.

Para instalar el J2SDK es preciso descargar primero el archivo binario, este se encuentra en Internet en la dirección:

<http://java.sun.com/j2se/1.4.1/download.html>

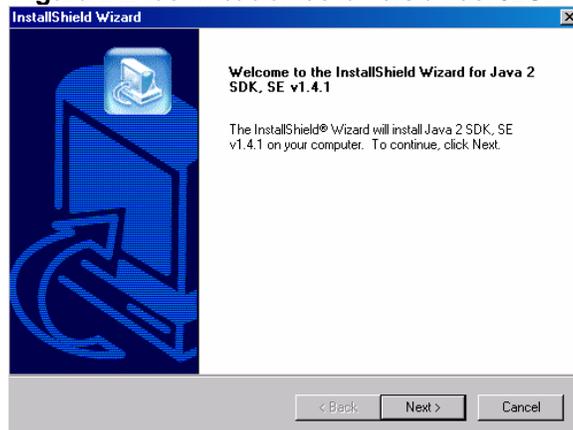
Una vez teniendo el archivo binario, se ejecuta dando inicio al proceso de la instalación, (véase la Figura D.1).

Figura D.1 Inicio de la instalación del J2SDK



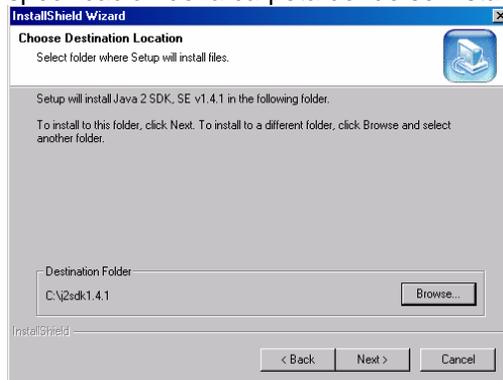
En la Figura D.2 se presenta la bienvenida e identificación de la versión del J2SDK.

Figura D.2 identificación de la versión del J2SDK



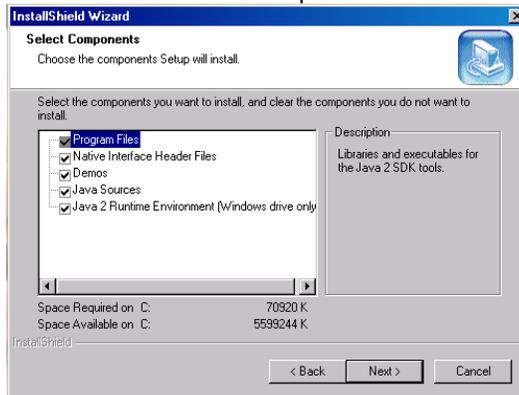
Para la correcta definición de las variables de entorno (indicadas en el manual del Usuario), es preciso indicar la instalación en la carpeta raíz, específicamente en el nombre mostrado en la Figura D.3.

Figura D.3 Especificación de la carpeta donde se instalará el J2SDK



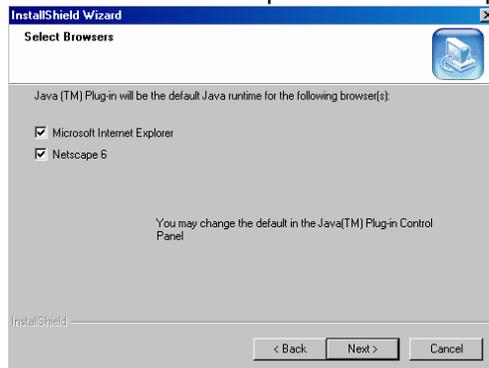
Es recomendable que se conserven los componentes que indica la instalación por defecto (véase la Figura D.4).

Figura D.4 Seleccionar los componentes de la instalación

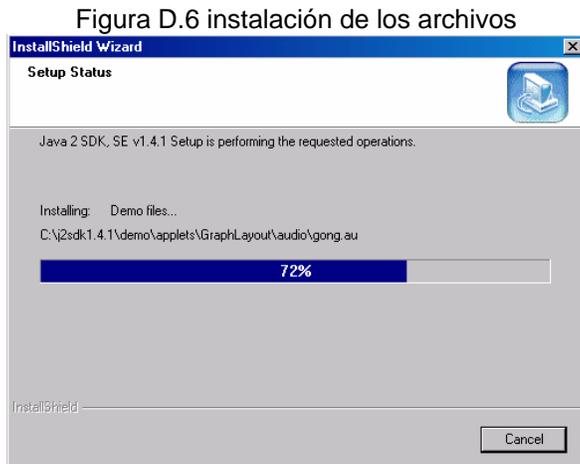


Se indica las aplicaciones con las que se podrá ejecutar el J2SDK, estos se observan en la Figura D.5.

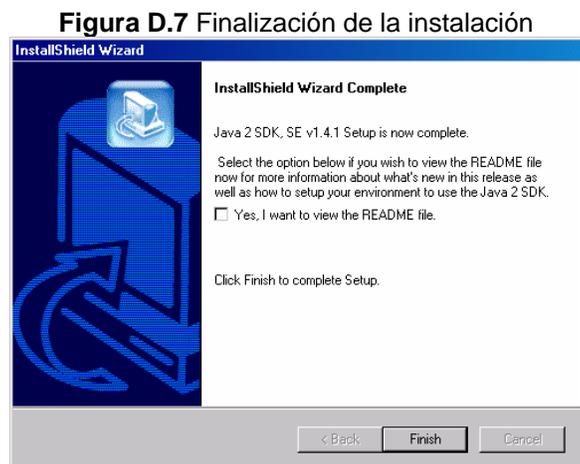
Figura D.5 seleccionar las aplicaciones de despliegue



Posteriormente da inicio a la instalación de los archivos y componentes seleccionados (véase la Figura D.6)



La instalación ha finalizado, es preciso configurar las variables de entorno y reiniciar la PC para la correcta ejecución del J2SDK.



La definición de las variables de entorno se encuentran explicadas en el tema **Requerimientos para la configuración del producto de software** en el punto *Configurar variables de ambiente* en el Manual del Usuario.

Apéndice E

Manual de Instalación de Tomcat 4.1.24

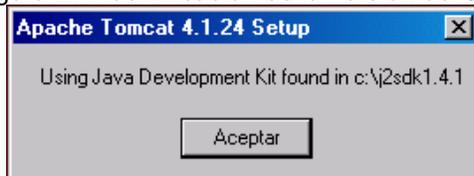
La instalación y configuración del servidor *Jakarta-Tomcat*, es la implementación oficial de referencia de la especificación Servlet 2.3 y JavaServerPages 1.2.

La explicación que sigue trata de la instalación de Tomcat en modo aislado, es decir, que es el propio Tomcat quien servirá todas las peticiones web, ya sean éstas de contenido HTML o servlets. La instalación de de Tomcat en conjunción con Apache, resulta mucho más complicada y sale del ámbito de este trabajo.

Es necesario tener instalado el JDK 1.4.1 Estándar Edition para la instalación de Tomcat. Las indicaciones que a continuación se proporcionan van dirigidas al uso de Tomcat con Windows 2000.

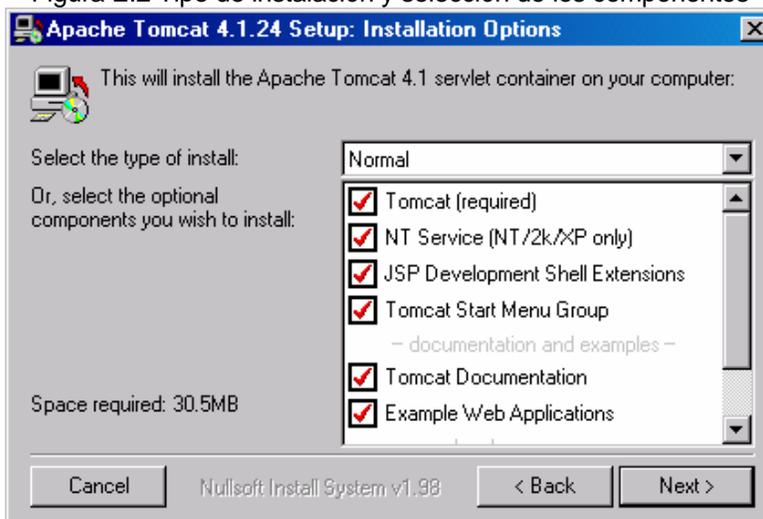
Es importante la versión binaria de Tomcat, la que se ha descargado para la instalación del sistema MECA es Tomcat 4.1.24. Al dar doble clic en el archivo binario se presenta la Figura E.1.

Figura E.1 Identificación de la versión del JDK



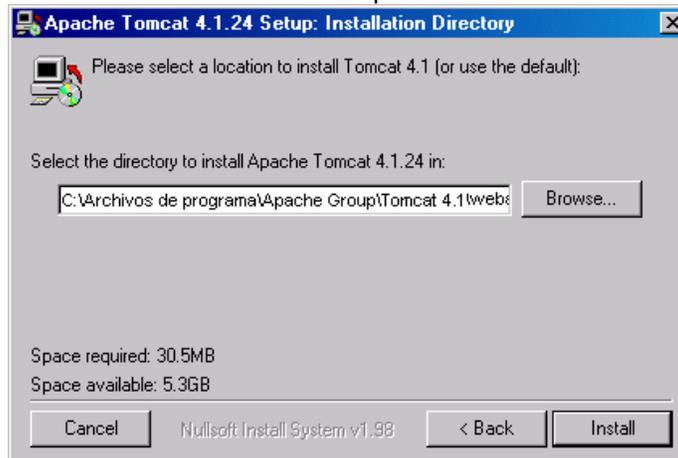
La instalación se realiza seleccionando el tipo Normal, se sugiere conservar los componentes por defecto (véase la Figura E.2).

Figura E.2 Tipo de instalación y selección de los componentes



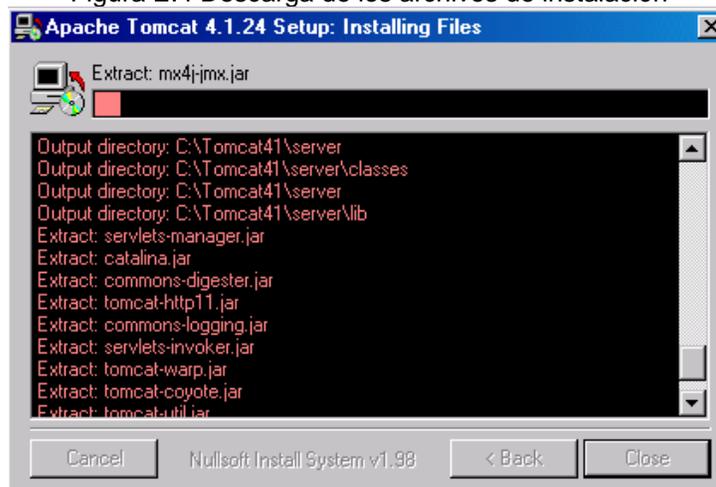
Para que la instalación de la herramienta MECA resulte exitosa, es importante que se indique Next para que guarde los archivos en las direcciones por defecto de Tomcat (véase Figura E.3).

Figura E.3 indicación del nombre de la carpeta donde ha de instalarse el Tomcat



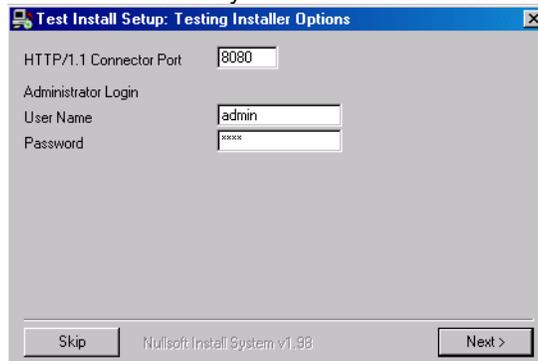
Posteriormente da inicio a la descarga de los archivos de instalación (véase la Figura E.4)

Figura E.4 Descarga de los archivos de instalación



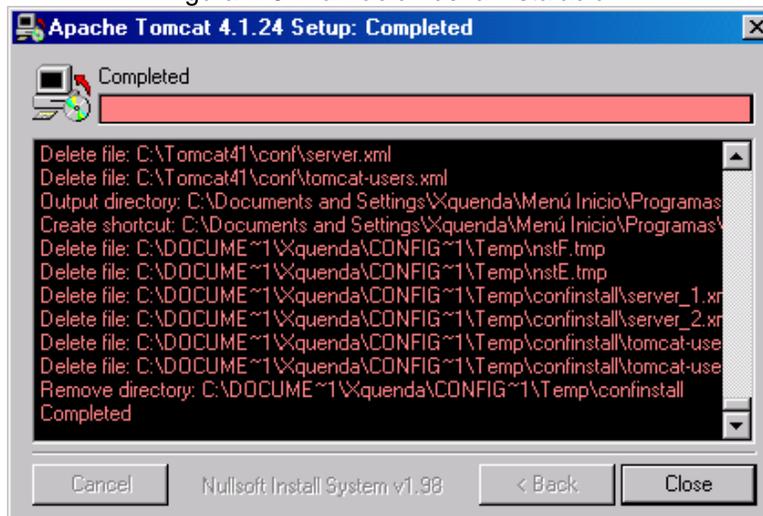
Es importante definir un nombre y clave de acceso al servidor, con la finalidad de poder registrar la aplicación en el Administrador del Servidor (véase la Figura E.5)

Figura E.5 Definición de un nombre y clave de acceso al Manejador del Servidor



Finaliza la instalación de Tomcat (véase la Figura E.6)

Figura E.6 finalización de la instalación



La definición de las variables de entorno se encuentran explicadas en el tema **Requerimientos para la configuración del producto de software** en el punto *Configurar variables de ambiente* en el Manual del Usuario.

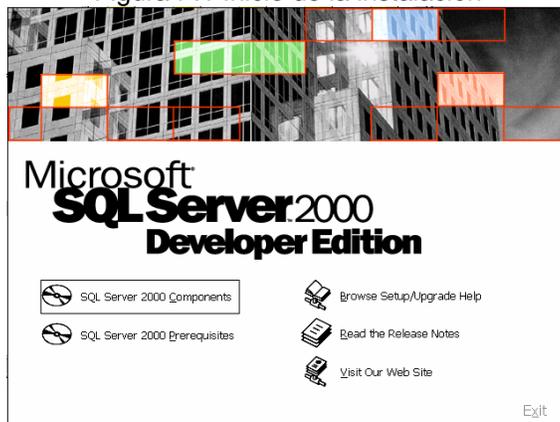
Apéndice F

Manual de Instalación de SQL Server 2000

Instalar SQL Server es muy sencillo, dispone de un asistente que se seguirá para realizar la instalación proporcionando unos datos que pedirá a lo largo del proceso. El proceso es sencillo y hay que tener en cuenta durante la instalación las opciones que se va a instalar de SQL Server.

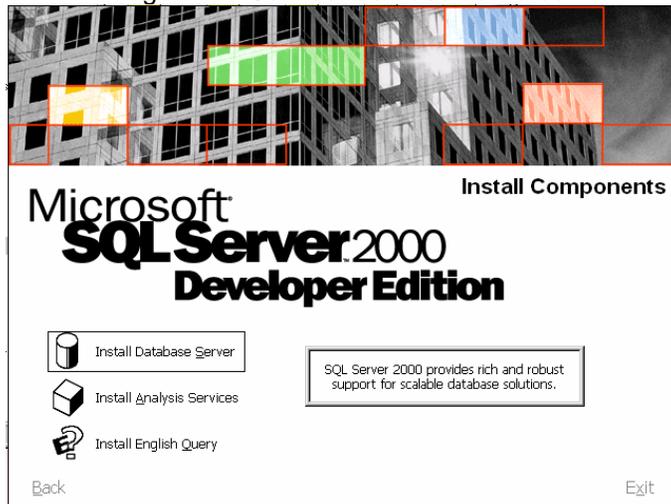
Después de ejecutarse el archivo *autorun.exe* se seleccionan las ventanas previas a la instalación: Componentes de SQL Server 2000 -> Instalar Servidor de bases de datos. La Figura F.1 muestra el inicio de la instalación.

Figura F.1 Inicio de la instalación



Elegir la instalación del servidor de Base de datos (véase Figura F.2).

Figura F.2 Servidor de Base de Datos



la forma de autenticación (personalmente utilizaría la autenticación en modo mixto) y el Nivel de intercalación opción que solo está disponible durante la instalación si se selecciona una instalación personalizada.

¿Qué es el Nivel de Intercalación? O Collation Settings en ingles, es la forma que

SQL Server tratará los caracteres de un idioma, el orden para utilizar los tipos de datos y la página de códigos. A diferencia de SQL Server 7.0 donde esta opción solo estaba disponible durante la instalación y una vez instalado ya no se podía modificar. SQL Server 2000 trata la intercalación de forma distinta permitiendo definirla a nivel de base de datos, de tabla y de columna, lo que quiere decir que se podrá tener Bases de Datos con distinto nivel de intercalación.

Con niveles de intercalaciones iguales en distintos servidores de bases de datos se puede disponer de un SQL Server compatible con otros SQL Server de la empresa que se instalaron de una determinada forma.

Por ejemplo, si se tiene instalado SQL Server con un juego de caracteres determinado como el de la imagen 9 (no distingue entre mayúsculas y minúsculas ni acentos), esto va a ser de gran utilidad a la hora de mover o restaurar Bases de Datos, si el juego de caracteres es diferente no se puede hacer. Aunque esta opción se puede aplicar después a cada Base de Datos que se cree nueva es recomendable hacerlo desde un principio y general para todo el SQL Server.

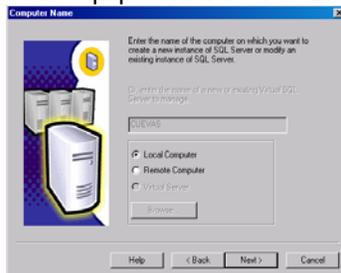
Si por alguna razón la instalación no es la misma que se tiene en otros servidores y se instala con un juego de caracteres diferente existe una solución para estos casos que permite hacer un “rebuild” a la Base de Datos Master del SQL Server y cambiar la instalación, pero este proceso se tiene que hacer con el SQL Server recién instalado porque de lo contrario se pierden las referencias a las Bases de Datos que se tenga y no las reconocerá. Para hacer el “rebuild” desde la línea de comandos escribir “rebuild” y aparecerá la imagen 10 si se pulsa en “Settings” aparece otra vez la imagen 9 donde se puede cambiar el juego de caracteres.

Para instalar el SQL Server 2000 Enterprise hay que seguir los pasos como muestran las Figuras F.3 y posteriores.

SQL Server 2000 puede instalarse de dos formas diferentes o mejor dicho pueden instalarse varias instancias en una misma maquina, esto es muy útil si por ejemplo se tiene un SQL Server 7.0 y se quiere instalar el 2000, se puede hacer y además se puede tener ejecutándose en la misma maquina los dos al mismo tiempo.

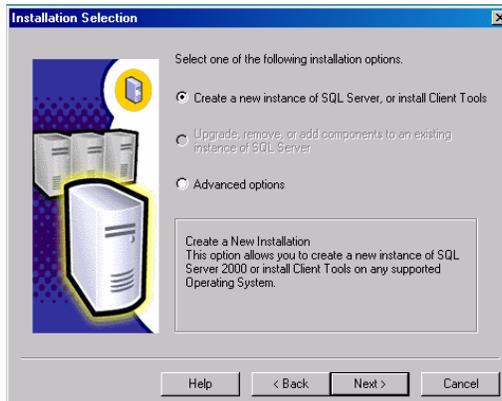
En la Figura F.3 el instalador pide el nombre del equipo y la forma de instalación, si es local o remota, si es remota se debe informar el equipo desde el cual se va a instalar.

Figura F.3 equipo en el cual se instalará



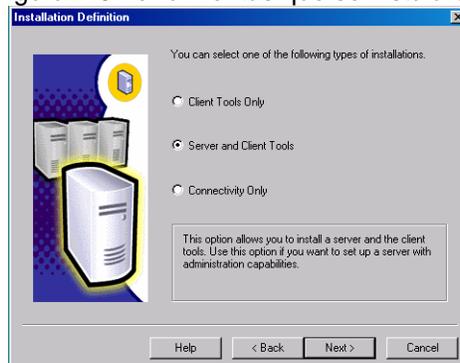
Seleccionar una opción de instalación, instalar una nueva instancia del SQL Server, instalar las partes clientes u opciones avanzadas donde se puede crear instalaciones desatendidas o reconstruir el registro del SQL Server si estuviera dañado. La opción central solo estará activa si ya se tiene un SQL Server instalado y se quiere modificar su instalación (véase la Figura F.4).

Figura F.4. crear una nueva instancia de SQL Server



Herramientas que se instalarán, es decir, si se instala solo las herramientas de cliente para tener acceso a un servidor remoto, herramientas de cliente y servidor como gestor de bases de datos o solo conectividad (véase Figura F.5).

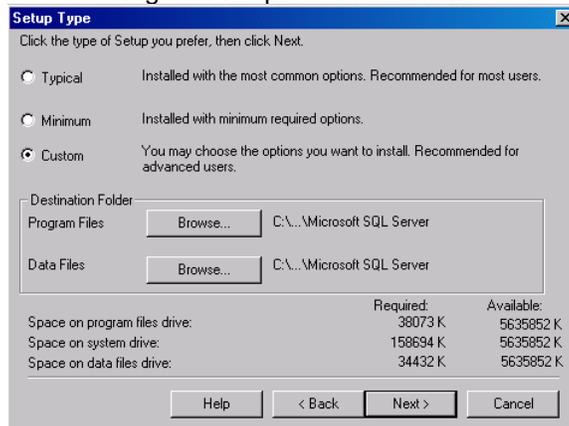
Figura F.5 herramientas que se instalarán



Nombre de la instancia que se va a crear, si es nueva instancia (no hay ningún SQL Server instalado) por defecto coge el nombre de la máquina aunque se puede cambiar, si es una segunda instalación se debe dar un nuevo nombre (elegir Default).

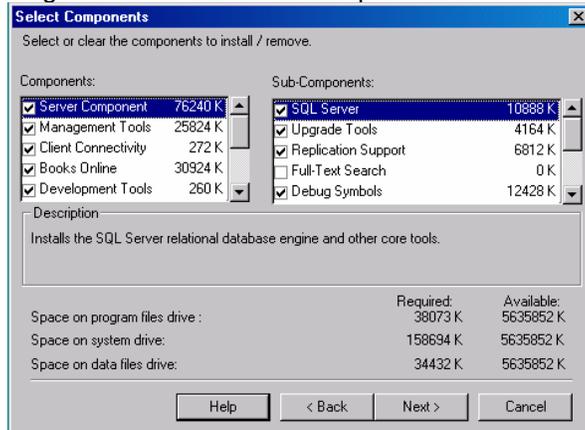
Tipo de instalación, personalmente se recomienda una instalación personalizada por lo comentado anteriormente es importante escoger el juego de caracteres (Figura F.9) para no tener problemas en futuras instalaciones y no tener que ir tocando el nivel de intercalación en cada base de datos que creamos, y elegir la ruta de los datos a una unidad con suficiente espacio en disco (véase Figura F.6).

Figura F.6 tipo de instalación



Selección de componentes a instalar, entre ellos ejemplos y ayudas (muy importante la ayuda, es el mejor manual de SQL Server). Véase la Figura F.7.

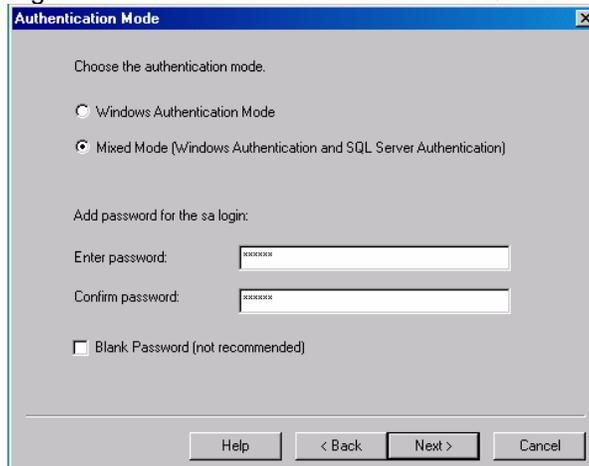
Figura F.7 Selección de componentes a instalar



Usuario que ejecutará los servicios del SQL Server y de SQL Agent, por defecto lo ejecuta el usuario administrador.

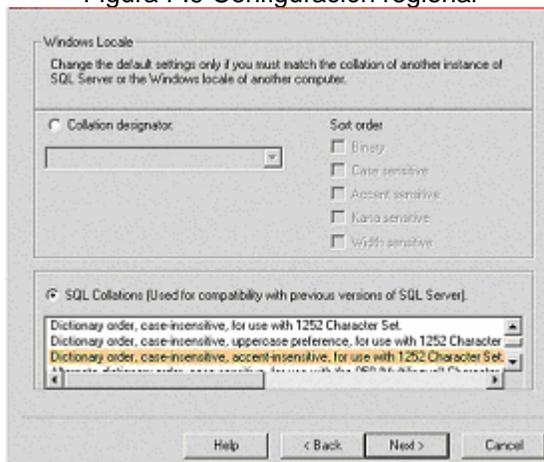
Modo de autenticación a SQL Server, se puede elegir entre autenticación windows (la autenticación se realiza por medio de usuarios pertenecientes al dominio) o modo mixto que la autenticación se realiza por medio de usuarios dados de alta en el SQL Server, si se elige esta segunda opción, no es recomendable dejar el password en blanco (véase Figura F.8).

Figura F.8 modo de autenticación de SQL Server



Configuración regional, en esta opción es donde se va a elegir la forma que nuestro SQL Server trabajará con los datos, es decir, se elegirá el nivel de intercalación. Es recomendable no dejarlo por defecto y seleccionar un nivel de intercalación apropiado a un lenguaje como por ejemplo el que muestra la imagen, en la imagen selecciona un SQL Server que no distinguirá entre mayúsculas ni entre acentos, ¿qué significa esto? Que cuando se realicen búsquedas, consultas o transacciones dará lo mismo poner “Tabla” que “tabla”. Si en todos los servidores SQL Server de la organización tienen la misma opción de nivel de intercalación, cuando se tenga la necesidad de restaurar Bases de Datos o moverlas de un SQL Server a otro será tan fácil como hacer un *restore* o copiarlas con *sp_attach_db* (véase Figura F.9)

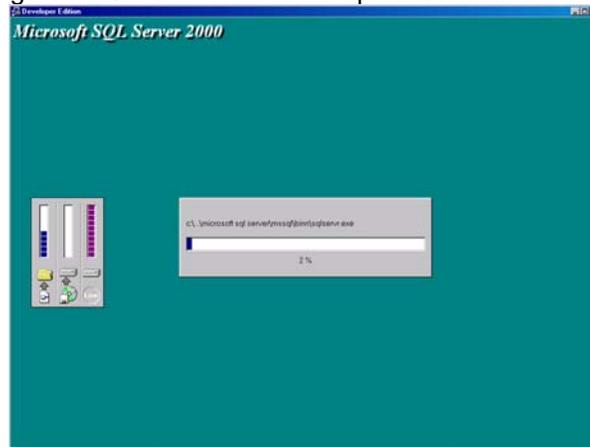
Figura F.9 Configuración regional



Por último pedirá el puerto y las bibliotecas de red que utilizará, este paso se puede dejar por defecto.

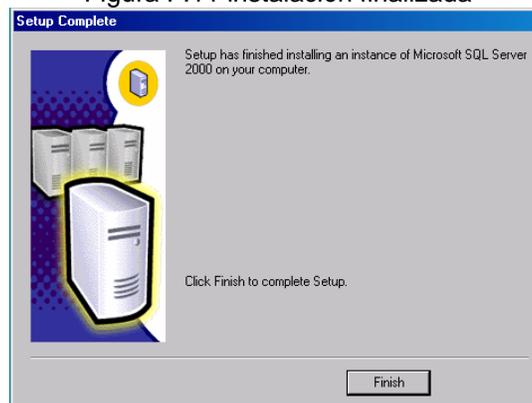
A partir de este punto SQL Server instalará las opciones seleccionadas (véase la Figura F.10).

Figura F.10 instalación de las opciones seleccionadas



Cuando finalice la instalación se tiene un SQL Server listo para trabajar (véase la Figura F.11).

Figura F.11 instalación finalizada



Apéndice G

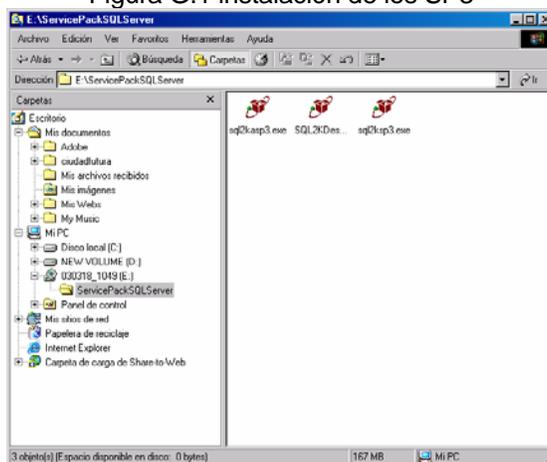
Manual de Instalación de los Service Pack SP3 de SQL Server 2000

Para los servidores de Base de Datos en ambiente de producción, microsoft recomienda implementar el Service Pack 3 (SP3) de SQL Server 2000 y las herramientas de seguridad de SQL Sever, los cuales contienen la actualización MS02-061. esta configuración ayudará a proteger el sistema contra una infección por el gusano Slammer.

La dirección de internet donde se puede descargar el SP3 es: <http://www.microsoft.com/sql/downloads/2000/sp3.asp>

Se recomienda descargar los tres archivos que presenta el SP3 y guardarlo en disco para futuras implementaciones (véase la Figura G.1).

Figura G.1 instalación de los SP3



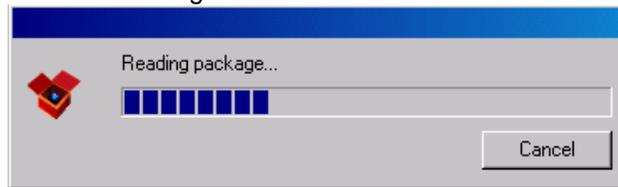
El SP3 se instalará en una carpeta indicada por defecto (véase la Figura G.2).

Figura G.2 Carpeta por defecto del SP3



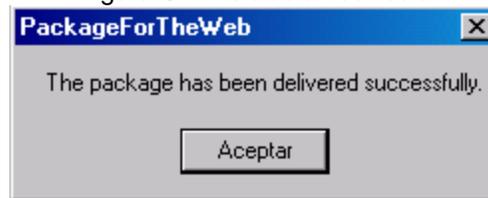
Al accionar el botón *Finish* el sistema comienza a leer y descargar el SP3 (véase Figura G.3).

Figura G.3 lectura del SP3



Una vez descargado el archivo, indica la instalación correcta (véase **la Figura G.4**), y así con los siguientes dos archivos que integran el SP3.

Figura G.4 instalación correcta



Documentación adicional

Se sugiere al usuario leer el material de la lista siguiente para un mejor desempeño del Sistema MECA.

- Manual de instalación de J2SDK
- Manual de instalación de Jakarta-Tomcat
- Manual de instalación de SQL Server 2000 básico y avanzado

Bibliografía

- [1] ANAYA, Raquel. "Métricas de calidad en esquemas conceptuales". Fondo editorial Universidad EAFIT. (Noviembre de 1988).
 - [2] ALBRETCH, A. J. "Measuring Application Development Productivity". Proc. IBM Application Development Symposium, Monterrey, CA, (Octubre 1971), pp. 83-92.
 - [3] ARANGO Q. Ana María. "Conjunto de métricas para el proceso de desarrollo de software", Tesis de grado. Universidad EAFIT, Escuela de Ingeniería, Departamento de informática y sistemas (1998).
 - [4] BERND Bruegge y ALLEN H. Dutoit. "Ingeniería de Software Orientado a Objetos". Edit. Pearson Educación, México (2002). pp. 23-281
 - [5] BOEHM, B.W., BROWN J.R. Lipow M. "Quantitative Evaluation of Software Quality". Proceedings 2nd International Conference on Software Engineering, (1976), pp. 592-605.
 - [6] BOOCH Grady. RUMBAUGH James. JACOBSON Ivar. "El lenguaje unificado de modelado". Addison Wesley, Madrid. pp. 11-327
 - [7] CERVERA P. Ángel. "El modelo de McCall como aplicación de la calidad a la revisión del software de gestión empresarial". Dpto. organización de Empresas Núñez moraleda, bernardo M, depto lenguajes y sistemas informáticos, universidad Cádiz. (2000)
 - [8] CORTÉS Felisa. "Actualización y automatización de un modelo cualimétrico para la evaluación de calidad del software". CIC – IPN, México (2003).
 - [9] DROMEY R. Geoff. "A model for Software Product Quality". IEEE Transactions on software Engineering, Vol 21, No 2, (February 1995).
 - [10] DROMEY R. Geoff. "Cornering the Chimera". IEEE Software (1996).
 - [11] DOLADO C. José Javier. FERNÁNDEZ S. Luis. "Medición para la gestión en la ingeniería del software". Editorial RaMa Madrid, (2000). pp. 149-166
 - [12] Fast facts de SQL Server
<http://www.microsoft.com/latam/sql/evaluation/overview/2000/fastfacts.asp>
 - [13] FEDCHACK Elaine. VIENNEAU Robert. "A history of software measurement at rome laboratory". A catalogue of rome lab-sponsored, Rome laboratory. Data&Analysis Center for Software (DACS), (25 Nov 1996).
 - [14] GARCIA, Ana María. "Métricas de Software" Universidad de las Villas. Cuba. Curso dictado en la Compañía Suramericana de Seguros S.A. Medellín – Colombia.
 - [15] GUTIÉRREZ Tornés Agustín, "Metodología para el Aseguramiento de la Calidad del Software", CIC-IPN. México (En edición 2000).
 - [16] HADLEY G., "PROBABILIDAD Y ESTADÍSTICA Una Introducción a la Teoría de la Decisión". Editorial. Fondo de Cultura Económica, México (1967). pp. 20-80
 - [17] HOWARD, J. A. SHETH, J. N. "*The theory of buyer behavior*". John Wiley & Sons, Inc, (1969).
 - [18] IEEE. Standard Glossary of Software Engineering Terminology. In *IEEE Software Engineering Standards Collection*. IEEE, Std 610.12-190. (1994).
 - [19] IEEE Std. 1061, IEEE Standard for a Software Quality Metrics Methodology, (1992).
 - [20] ISO 9000-3 Normas para la administración de la calidad y aseguramiento de la calidad Parte 3: Directrices para la aplicación de ISO 9001 al desarrollo, suministro, y mantenimiento del software, (1991).
 - [21] ISO/IEC 12207, Standard for Developing Software Life Cycle Processes (Febrero 1998)
 - [22] ISO 9001 Sistemas de gestión de la calidad. Requisitos (2000)
 - [23] ISO/IEC 9126, Software Product Evaluation, Quality, Characteristics and Guidelines for their Use. Information Technology, (1991).
 - [24] ISO/IEC 9126-2: Métricas externas para una validación de la calidad de software, (1991).
 - [25] ISO/IEC 9126-3: Métricas internas para una validación de la calidad de software, (1991)
 - [26] ISO/IEC TR 15504 Tecnología de información – Valoración del Proceso de software,
-

-
- (1993).
- [27] ISO/IEC 14598-5. Information Technology. Software Product Evaluation. Process for Evaluators, (1998)
- [28] J.B. Dreger. "Function Point Analysis". Prentice-Hall, 1989
<http://www.sc.ehu.es/jiwdocoj/mmis/fpa.htm>
- [29] KAN Stephen. "Metrics and models in Software Quality Engineering". Addison-Wesley, (1995)
- [30] KERNAN, J. B. y SCHIMIDT, S. L., "The many meanings (and implications) of satisfaction guaranteed". Journal of retailing, vol. 61, Num. 4, (1985).
- [31] LARMAN, C. "UML y Patrones Introducción al análisis y diseño orientado a objetos". Primera Edición. Prentice Hall, México, (1999) pp. 83-405.
- [32] LAWRENCE E. Hyatt. LINDA H. Rosenberg. "A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality". (24 de Abril de 1996).
- [33] NICK Feamster. "On Patrol Security in Software Engineering", ACM Crossroads Student Magazine The ACM's First Electronic Publication, Copyright 2000-2002 by ACM, Inc.(25 abril 2001).
- [34] NIGEL Bevan. MOTOEI Azuma. "Quality un Use: Incorporating Human Factors into the Software Engineering Lifecycle". National Physical Laboratory Tedding, Middlesex, and Department of Industrial Engineering and Management, waseda University Okubo, Shinjuku-ku, Tokyo Japan (1997).
- [35] [OLIVER. R.L. "A cognitive model of the antecedents and consequences of satisfaction decisions", *journal of marketing research*, vol. 17, (1980), pp. 460-469.
- [36] PRESSMAN S. Roger "Ingeniería del software Un enfoque práctico" Cuarta Edición. McGrawHill (1998) pp. 69-87.
- [37] PURSNANI Vandana, "An Introduction to Java Servlet Programming", ACM Crossroads Student Magazine The ACM's First Electronic Publication, Copyright 2000-2002 by ACM, Inc.(19 november 2001).
- [38] QUALYSOFT. "Sistema para el Aseguramiento de la Calidad del Software". Autores varios, Publicación electrónica del CDI Centro de Desarrollo Informático, la Habana, Cuba (1991) .
- [39] SANTAOLALLA S. Javier, "Concepción de un modelo para el aseguramiento de calidad de componentes reusables de software". Tesis de doctorado CIC-IPN, México (2003).
- [40] SOTO, Martín. CARRILLO Angela. "Técnicas de Calidad de software para desarrolladores profesionales". Universidad de los Andes, Facultad de Ingeniería, Departamento de sistemas y computación. Programa de educación continuada corporativa Suramericana de Seguros S.A. (1998).
- [41] ZDNet Where Technology Means Business <http://downloads-zdnet.com.com/3120-20-0.html?qt=ER-Win&tq=dl-2001>
-

Sistema MECA

Manual o Guía del Usuario

Índice.

Temas	Pág.
1 Introducción	2
1.1 <i>Ventajas del sistema</i>	2
1.2 <i>Limitaciones del sistema</i>	2
1.3 <i>Organización o estructura del documento</i>	2
2 Generalidades	4
3 Condiciones para la instalación	4
3.1 <i>Requerimientos para la configuración del producto de software</i>	4
4 Instalación del Sistema MECA (Manual de Instalación)	13
5 Desinstalación del Sistema MECA (Manual de desinstalación)	16
6 Relación de los documentos de soporte del software	18
7 Requerimientos del ambiente de operación	19
7.1 <i>Hardware</i>	19
7.2 <i>Software</i>	19
7.3 <i>Organizativo</i>	19
8 Procedimientos para leer el contenido del archivo LEEME.TXT	20
9 Procedimientos para la creación de las copias de respaldo o de trabajo	20
10 Iniciación	20
11 Utilización o Entrenamiento	21
<i>Apéndice A. Glosario</i>	34
<i>Apéndice B. Relación de todas las abreviaturas</i>	35
<i>Apéndice C. Relación de todos los mensajes de error</i>	36
<i>Apéndice D. Manual de Instalación de J2SDK</i>	37
<i>Apéndice E. Manual de Instalación de Tomcat</i>	40
<i>Apéndice F. Manual de Instalación de SQL Server 2000</i>	43
<i>Apéndice G. Manual de Instalación de Service Pack 3</i>	49
Documentación adicional	51

1 Introducción.

El Sistema MECA tiene como propósito la agilización del proceso de evaluación del software; es decir, trata del desarrollo de una herramienta para automatizar MECA (Modelo Cualimétrico para la Evaluación de la Calidad del Software). Éste se basa en la ampliación y desarrollo de los principios expuestos en la Norma ISO/IEC (Organización Internacional para la Estandarización/Comisión Internacional Electrotécnica) 9126 .

En la actualidad no existe una herramienta que realice esta tarea. Un instrumento de este tipo facilita la aplicación y propagación de la metodología MECA.

La herramienta ha sido diseñada utilizando los principios y lineamientos de la metodología UML (Lenguaje Unificado de Modelado) y se ha programado en Java en un ambiente de red. Para la Base de Datos se utilizó el SQL Server.

Además, se instrumentó y elaboró un método para incorporarle al modelo la evaluación de la Calidad en Uso.

1.1 Las ventajas del sistema:

- La implementación de la herramienta en Internet permite disponer del servicio de evaluación las 24 horas del día, durante los 365 días del año.
- El conocimiento de un modelo cualimétrico que permite mejorar la calidad de los sistemas informáticos.
- El conocimiento e implementación de un sistema para el aseguramiento de la Calidad del Software en la organización.
- El conocimiento del grado de satisfacción con respecto al cliente que tiene el producto de software diseñado.

1.2 Limitaciones del sistema:

No se conoce alguno

1.3 Organización del documento:

El sistema está estructurado en base a dos aspectos importantes:

Funciones del negocio: se trata del registro de datos para la correcta administración de la evaluación, así como la evaluación misma.

Funciones de soporte informático: se trata del establecimiento de módulos de ayuda, seguridad mediante claves de acceso, mantenimiento debido a la adaptación de nuevas características de calidad al modelo.

De usarse el manual por primera vez, se sugiere realizar la lectura secuencial, de lo contrario consultar solo el tema de interés.

Este Manual del usuario está organizado en catorce temas:

Introducción. En el cual se exponen las ventajas y limitaciones del Sistema MECA.

Generalidades. Esta parte presenta un panorama general del documento.

Condiciones para la instalación. En esta parte se presentan los productos y configuración de variables de entorno previos a la instalación del Sistema MECA.

Instalación del Sistema MECA. Esta parte presenta los pasos que integran instalación del sistema.

Desinstalación del Sistema MECA. Esta parte presenta los pasos que integran la desinstalación de los archivos del sistema.

Soportes magnéticos. Esta parte presenta la forma en que se encuentra estructurado el árbol de archivos en el disco compacto del Sistema MECA.

Relación de los documentos de soporte del software. Esta parte presenta los manuales de instalación que acompañan el presente Manual de usuario.

Requerimientos del ambiente de operación. Esta parte presenta los requerimientos de ambiente de hardware, software y organizativo donde se implementará el Sistema MECA.

Procedimientos para leer el contenido del archivo LEEME. Esta parte de lo sencillo que resulta la lectura del archivo LEEME.TXT ubicado en los archivos de instalación del sistema.

Procedimientos para la creación de las copias de respaldo o de trabajo. Esta parte presenta la sugerencia de productos comerciales que apoyan a la creación de respaldos del Sistema MECA.

Iniciación. Esta parte presenta el procedimiento para ejecutar la herramienta de evaluación.

Utilización o entrenamiento. Esta parte presenta los pasos para la correcta administración y Evaluación de la Calidad del Software.

Además el trabajo incluye un glosario en el que se presenta una lista de términos que complementan sus temas; varios apéndices que presentan gráficamente en su contenido, una relación de errores comunes, manuales de instalación de J2SDK, Tomcat, SQL Server, Service Pack 3 para SQL Server y una documentación adicional donde se recomiendan los documentos que facilitarían la instalación y administración de la Evaluación de la calidad del Software.

2 Generalidades.

La herramienta para la evaluación de la calidad del software fue realizada en el Centro de Investigación en Computación del Instituto Politécnico Nacional (CIC-IPN) <http://www.cic.ipn.mx> como parte de la tesis de la maestría en Ciencias de la Computación de la Lic. Alma Delia Cuevas Rasgado cralma@sagitario.cic.ipn.mx docente adscrita al departamento de Sistemas y computación del Instituto Tecnológico de Oaxaca <http://www.itox.mx/> y bajo el asesoramiento del Dr. Agustín Gutiérrez Tornés atornes@cic.ipn.mx docente investigador del CIC-IPN y participante en la creación de la primera versión del modelo cualimétrico que implementa la herramienta.

Los responsables del uso, mantenimiento y conservación de la Herramienta para evaluar la calidad de los software: la Lic. Alma Delia Cuevas Rasgado, tel: 57296000 ext 56554 y el Dr Agustín Gutiérrez ext. 56600 se encuentran actualmente laborando en el laboratorio de Tecnología de Software del CIC-IPN.

La herramienta es gratuita al público en general su registro está a consideración de las autoridades académicas correspondientes del IPN.

Se brinda las copias para la instalación del servidor que atienda las evaluaciones que se requieran bajo los términos que regula la administración de software del CIC-IPN.

La herramienta proporciona de una manera automática una clave de seguridad para el Administrador o Evaluador que lo use.

La ayuda sobre el manejo de la herramienta y de la comprensión de la metodología MECA se presentan como módulos internos del sistema, el mantenimiento de la herramienta y del modelo corresponde a las personas antes citadas, de surgir alguna duda en su uso, efecto causado en su instalación o manejo, se recomienda consultar a las personas responsables de su elaboración en los teléfonos y direcciones de correo citados anteriormente.

3 Condiciones para la instalación.

3.1 Requerimientos para la configuración del producto de software.

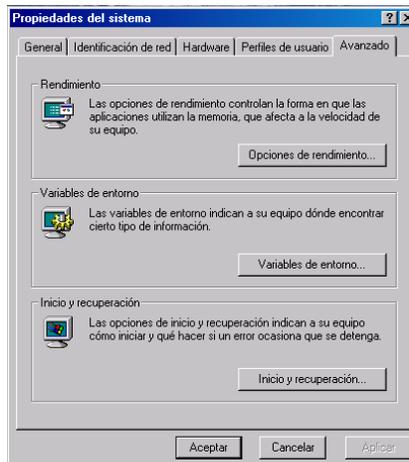
La herramienta por el lado del cliente solo requiere de la instalación de Internet Explorer 4.0 o más.

La herramienta por el lado del servidor requiere los siguientes prerrequisitos:

- Instalación de J2SDK 1.4.1 (véase Apéndice D)
- Instalación de Catalina Tomcat 4.1.24 (véase apéndice E)

- Configurar Variables de ambiente. Las variables de ambiente se encuentran en *MIPC –Propiedades –Avanzado* (ver Figura 1)

Figura 1. Ventana de definición de las variables de ambiente o entorno



Se elige *–Variables de entorno, –Variables del sistema* con la finalidad de *Crear, Editar o Eliminar* variables de entorno. Las variables que requieren definirse se presentan a continuación:

Nombre de la variable: CATALINA_HOME

Valor de la variable: c:\Tomcat41 (o la ruta que se haya indicado en la instalación de Tomcat)

La creación de la variable se presenta en la Figura 2, y así con las siguientes:

Nombre de la variable: CLASSPATH

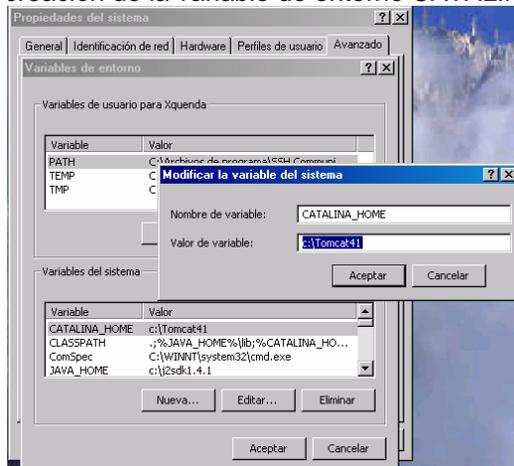
Valor de la variable:

.;%JAVA_HOME%\lib;%CATALINA_HOME%\common\lib\servlet.jar

Nombre de la variable: JAVA_HOME

Valor de la variable: c:\j2sdk1.4.1 (o la ruta que se haya indicado en la instalación de J2SDK)

Figura 2. creación de la variable de entorno CATALINA_HOME

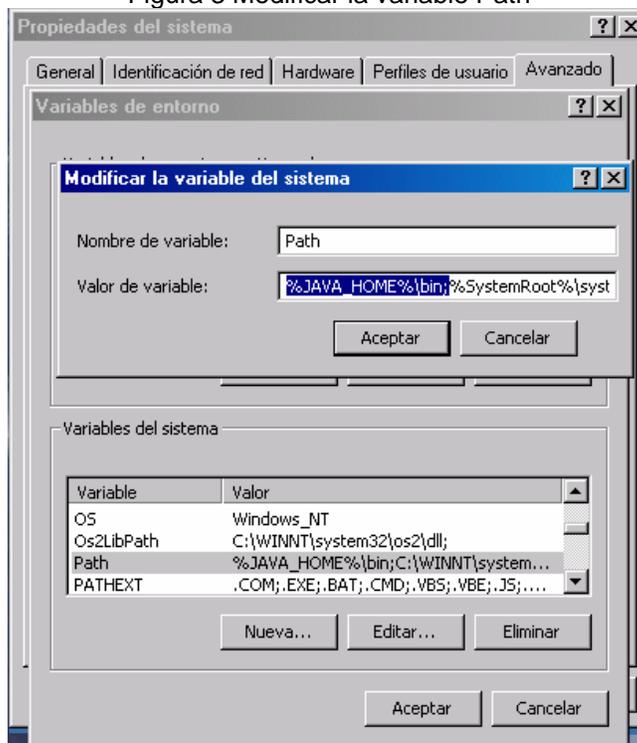


La siguiente variable de entorno se debe modificar (las instrucciones indicadas se añadirán al principio del valor de la variable, como se presenta en la parte sombreada de la Figura 3) porque ya se encuentra incluida en la lista de variables del sistema:

Nombre de la variable: Path

Valor de la variable: %JAVA_HOME%\bin;

Figura 3 Modificar la variable Path



Nombre de la variable: SERVLET

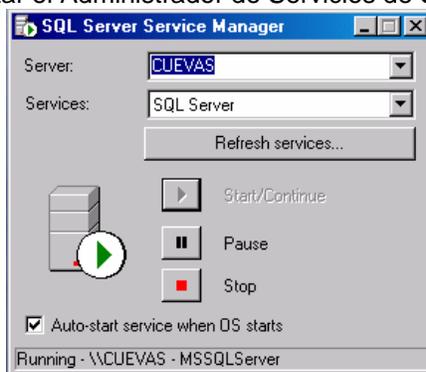
Valor de la variable: %CATALINA_HOME%\webapps\examples\WEB-INF\classes

- Instalar SQL Server 2000, como modo Administrador de Base de Datos (véase Apéndice F)
- Instalar los parches (Service Pack 3) de SQL Server 2000 (véase Apéndice G)
- Copiar archivos: meca.mdf y meca_Log.LDF en: C:\Archivos de programa\Microsoft SQL Server\MSSQL\Data

En caso de que la carpeta de la PC no se llame “Archivos de programa” es posible que sea “Program files” las demás carpetas tendrán el mismo nombre.

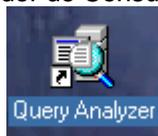
Verificar que el Administrador de Servicios de SQL Server 2000 esté ejecutándose, de manera contraria accionarlo (véase Figura 4).

Figura 4. Ejecutar el Administrador de Servicios de SQL Server 2000



Enlazar, agregar o adjuntar los archivos meca.mdf y meca_Log.LDF desde el *Analizador de Consultas* de SQL Server 2000, mediante un Script. Este procedimiento se presenta en la Figura 5 en el cual se presenta el ícono correspondiente a este proceso

Figura 5 abrir el Analizador de Consultas de SQL Server 2000



Una vez elegido el ícono y dar doble clic en él, se presenta la conexión con el servidor local llamado “Cuevas”, (véase la Figura 6) se escribe el nombre de entrada y la contraseña definidos en la instalación (el nombre del servidor donde se está instaló el Analizador deberá aparecer en este caso).

Figura 6 Acceso al Analizador de Consultas de SQL Server 2000

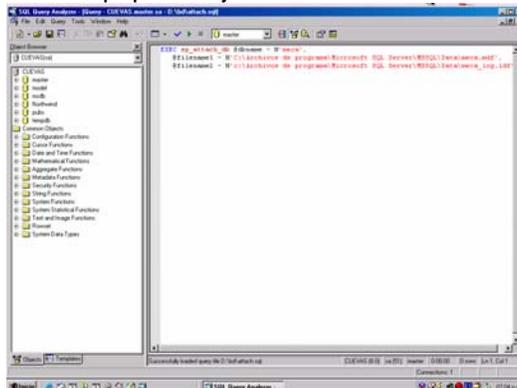


Desde la Base de datos *master* presentada por defecto por el Analizador de Consultas, se escriben dos instrucciones que sirven para adjuntar la Base de Datos MECA:

```
EXEC sp_attach_db @dbname = N'meca',  
@filename1 = N'c:\Archivos de programa\Microsoft SQL Server\MSSQL\Data\meca.mdf',  
@filename2 = N'c:\Archivos de programa\Microsoft SQL Server\MSSQL\Data\meca_log.ldf'
```

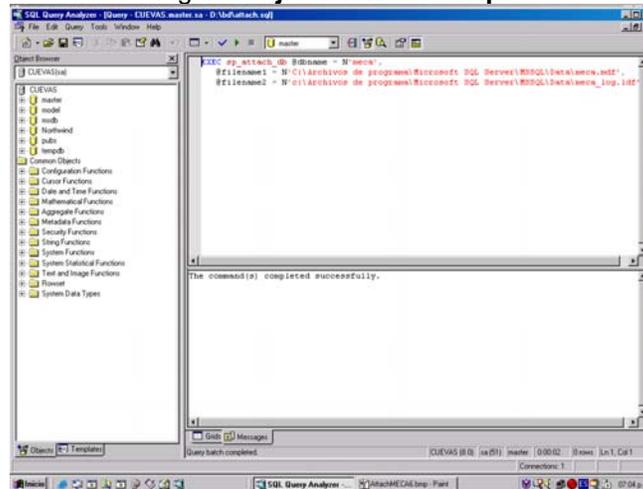
Las instrucciones se presentan en la Figura 7, la carpeta principal “Archivos de programa” puede aparecer en la PC como “Program Files”; si es así, sustituirlo por el que indique.

Figura 7 Script para adjuntar la Base de Datos MECA



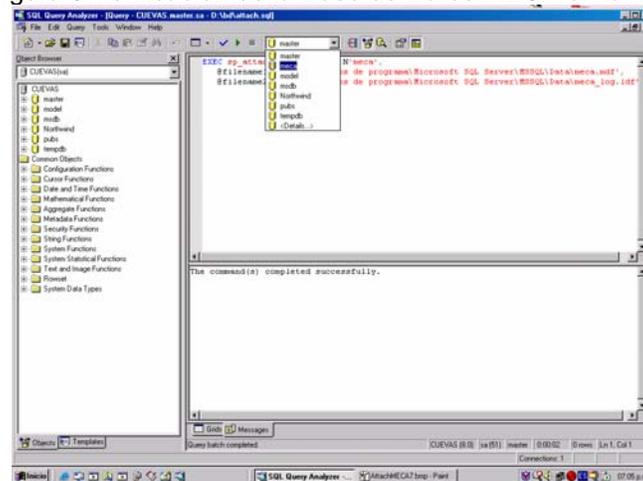
Una vez escritas las tres líneas de código, pulsar la tecla F5. El SQL Server 2000 presenta la pantalla que se observa en la Figura 8 con la indicación de que el *Script* se ha ejecutado satisfactoriamente.

Figura 8 ejecución del Script



Ahora aparecerá la Base de Datos MECA en la lista que ofrece el SQL Server 2000 (véase la Figura 9) después de este paso se recomienda cerrar el Procesador de Consultas y abrir nuevamente para que cargue correctamente la Base de Datos MECA.

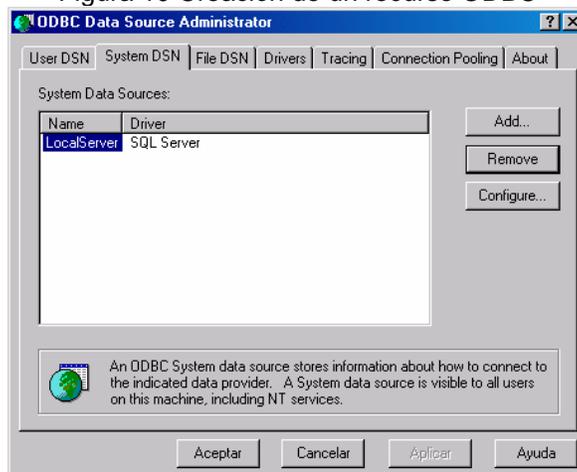
Figura 9 Verificación de la Base de Datos MECA en la lista



- Configurar el recurso en ODBC.

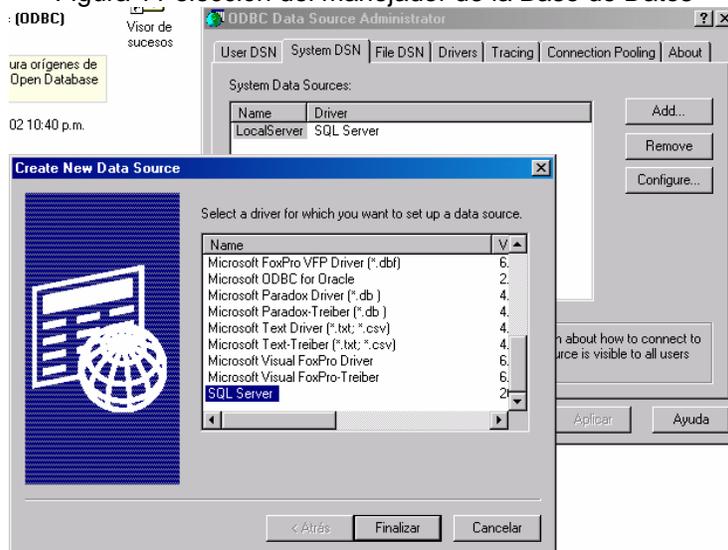
Para realizar este paso es necesario acceder el menú *-Inicio, -Configuración, -Panel de control, -Herramientas administrativas* y posicionarse en el ícono *-Orígenes de datos(ODBC)*, al dar doble clic en él, se presenta una pantalla con varias pestañas, se debe elegir la de *System DNS*, (véase la Figura 10).

Figura 10 Creación de un recurso ODBC



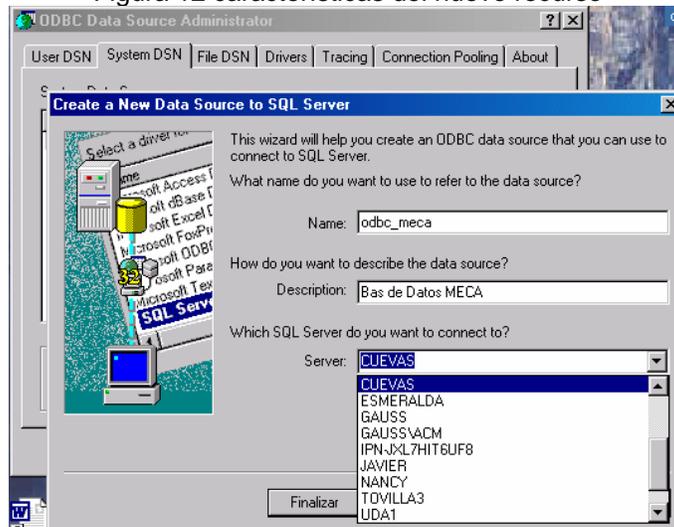
Para Añadir un recurso se pulsa la opción *Add* y se elige el manejador de la Base de Datos, en este caso SQL Server, (véase la Figura 11).

Figura 11 elección del manejador de la Base de Datos



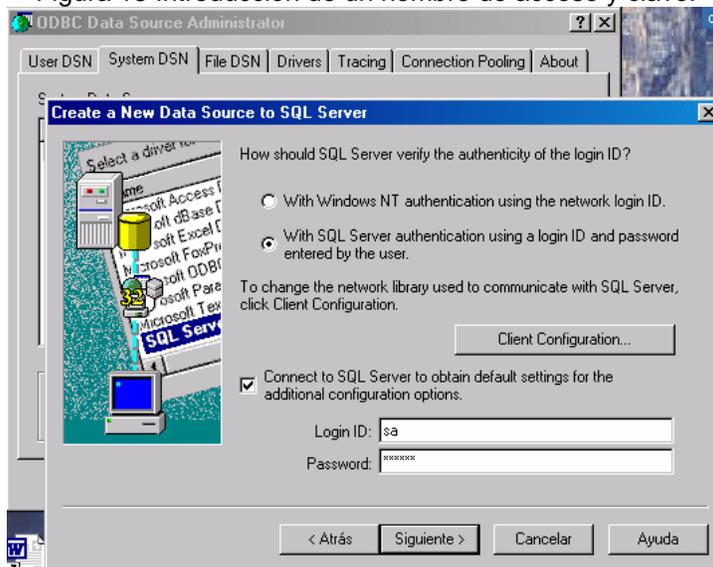
Al teclear la Opción *Finalizar* el sistema pide el nombre del recurso, una breve descripción y el nombre del servidor donde estará implementado, (véase la Figura 12).

Figura 12 características del nuevo recurso



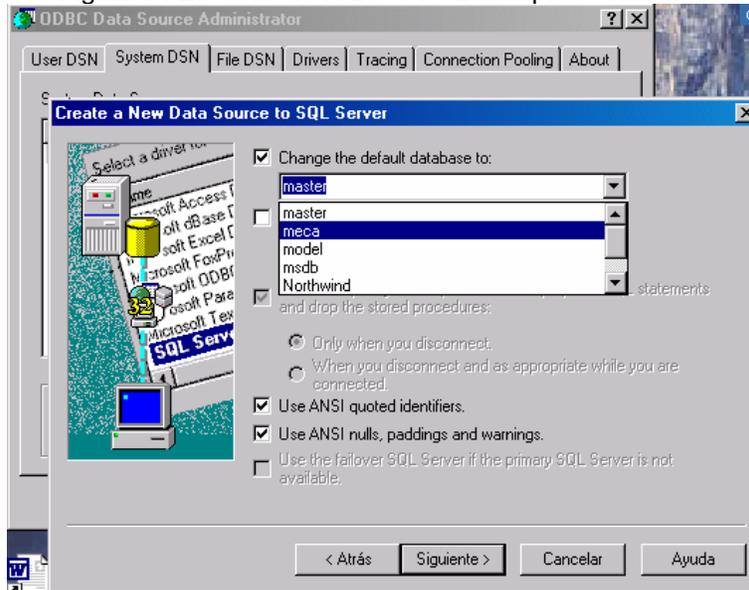
El sistema presenta el tipo de autenticación a la Base de Datos MECA, en la Figura 13 se presentan las opciones a elegir para este proceso.

Figura 13 Introducción de un nombre de acceso y clave.



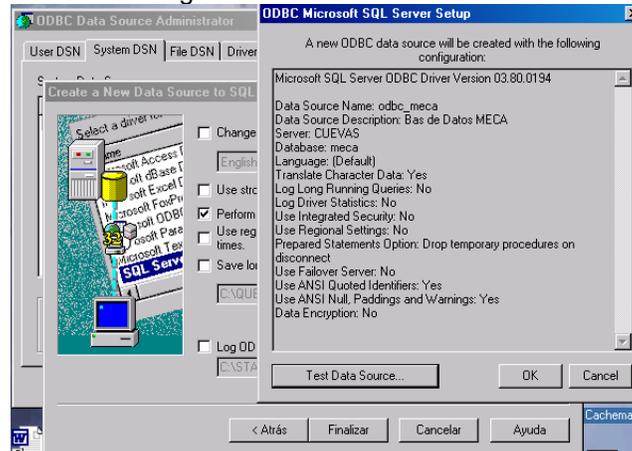
A continuación, se elige la Base de Datos por defecto para el recurso creado, (véase la Figura 14).

Figura 14 Elección de la Base de Datos para el recurso



Posteriormente, se recomienda probar el recurso, este proceso se presenta en la Figura 15. En el que se verifica la buena conexión con el servidor y la realización correcta de la creación del recurso, si el sistema indica una falla, se procede a su verificación o eliminación del recurso y aplicar los pasos anteriores.

Figura 15 Prueba del recurso



- Instalar los archivos de Meca.

El Sistema MECA contiene varios archivos que integran el proceso de instalación y ayuda, estos se presentan en la Figura 16. mismos que se distribuyen para su correcta ejecución.

Figura 16 conjunto de archivos del Sistema MECA

Instalar_Meca_Build_Output		Carpeta de archivos	04/05/2003 08:04 p...
Instalar_Mecalocales		Carpeta de archivos	04/05/2003 08:04 p...
Instalar_Meca.iap.xml	197 KB	InstallAnywhere Proj...	04/05/2003 08:04 p...
Manual de j2sdk.doc	4,104 KB	Documento de Micro...	04/05/2003 05:48 p...
Manual de Servicepack SP3.doc	1,758 KB	Documento de Micro...	04/05/2003 05:51 p...
Manual de SQL Server 2000.doc	6,280 KB	Documento de Micro...	04/05/2003 05:50 p...
Manual de Tomcat.doc	2,323 KB	Documento de Micro...	04/05/2003 06:12 p...
Manual del sistema MECA.doc	25,818 KB	Documento de Micro...	04/05/2003 11:51 p...
LEEME.txt	7 KB	Documento de texto	04/05/2003 11:54 p...

Para instalar los archivos, se recomienda leer el "Manual de Instalación del Sistema MECA".

- Levantar Tomcat.

Este proceso se puede realizar mediante cualquiera de los dos pasos siguientes:

- Desde una ventana de sistema (DOS), teclear lo siguiente desde la línea de comandos: `c:\Tomcat41\bin` el sistema responde:

```
c:\Tomcat41\bin>
```

```
c:\Tomcat41\bin>startup
```

La respuesta del sistema se presenta en la Figura 17.

- Desde el menú *-Inicio, -Programas, -Apache Tomcat 4.1 -Start Tomcat* (véase la Figura 17).

Para la detención de Tomcat se teclea Shutdown o Stop Tomcat desde cualquiera de los pasos anteriores.

Figura 17. inicio de Tomcat

```

Tomcat
2/05/2003 07:23:23 PM org.apache.commons.modeler.Registry loadRegistry
INFO: Loading registry information
2/05/2003 07:23:24 PM org.apache.commons.modeler.Registry getRegistry
INFO: Creating new Registry instance
2/05/2003 07:23:24 PM org.apache.commons.modeler.Registry getServer
INFO: Creating MBeanServer
2/05/2003 07:23:26 PM org.apache.coyote.http11.Http11Protocol init
INFO: Initializing Coyote HTTP/1.1 on port 8080
Starting service Tomcat-Standalone
Apache Tomcat/4.1.24
2/05/2003 07:23:35 PM org.apache.coyote.http11.Http11Protocol start
INFO: Starting Coyote HTTP/1.1 on port 8080
2/05/2003 07:23:35 PM org.apache.jk.common.ChannelSocket init
INFO: JK2: ajp13 listening on /0.0.0.0:8009
2/05/2003 07:23:35 PM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=50/40 config=c:\Tomcat41\conf\jk2.properties

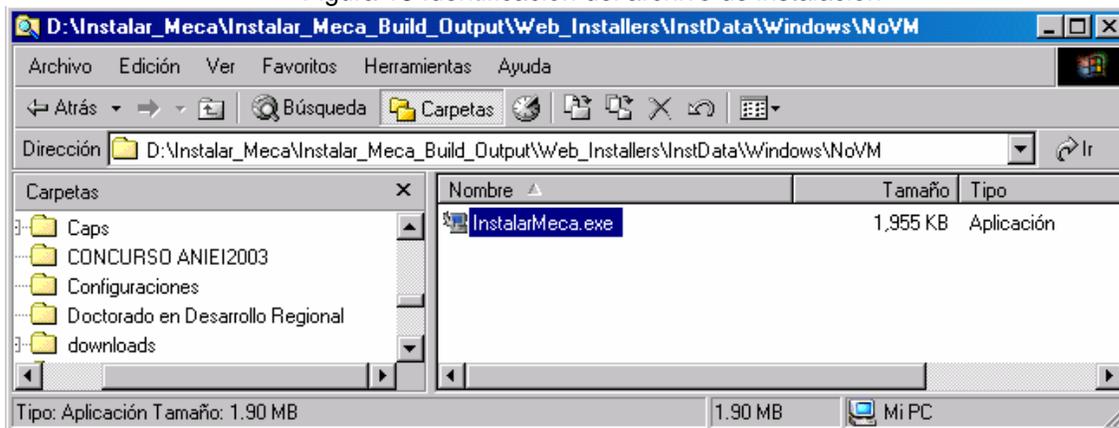
```

4 Instalación del Sistema MECA

El Sistema MECA cuenta con un programa instalador que está contenido en la carpeta llamada "Instalar MECA", el ejecutable se llama "InstalarMECA.exe" y está ubicado en el disco de instalación, en la ruta:

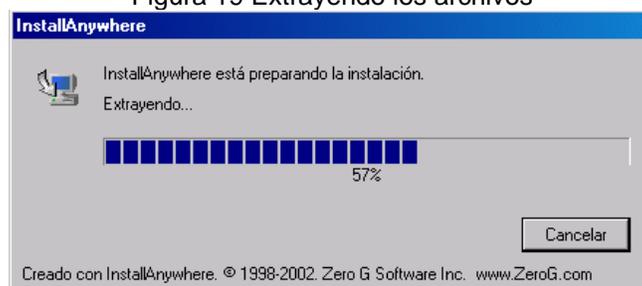
“InstalarMECA\Instalar_Meca_Build_Output\Web_Installers\InstData\Windows\NoVM”
Como se observa en la Figura 18.

Figura 18 Identificación del archivo de instalación



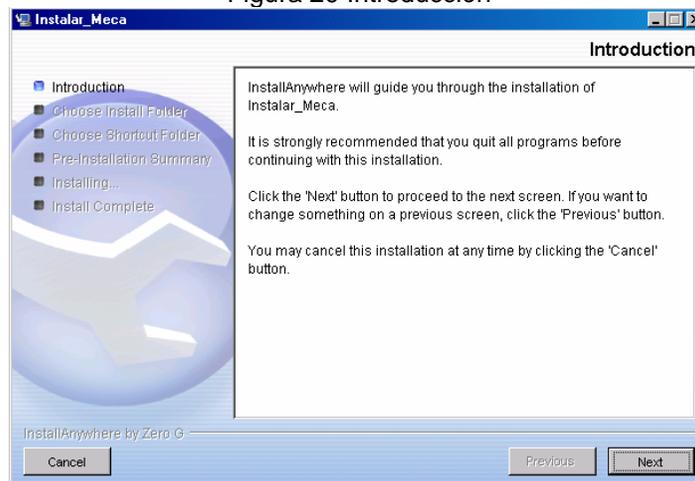
Al dar doble clic en el nombre del archivo, se presenta la pantalla mostrada en la Figura 19, en la cual se extraen los archivos a instalar.

Figura 19 Extrayendo los archivos



Posteriormente, el instalador presenta la pantalla mostrada en la Figura 20 en la cual se especifican algunas recomendaciones.

Figura 20 Introducción



Como primera versión del Sistema MECA, requiere instalarse específicamente entre los archivos del servidor de aplicaciones. El instalador da por defecto una carpeta destino llamada “*Instalar_Meca*” pero es necesario cambiarla, pulsando el botón “*Choose*” para elegir la ubicación correcta en el servidor de aplicaciones (véase la Figura 21).

Figura 21 elegir la ubicación de la instalación



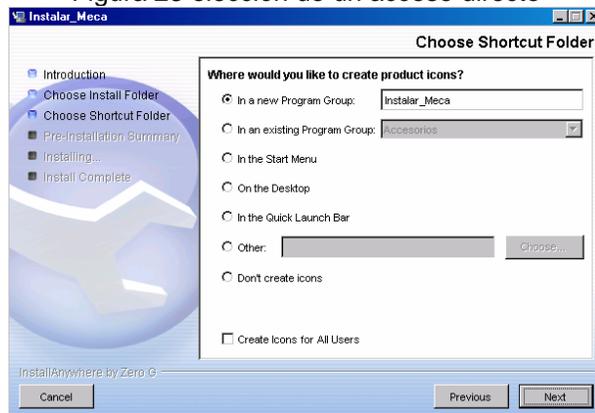
Al pulsar el botón “*Choose*” se presenta el árbol de directorios, el usuario podrá elegir la ruta donde se instalarán los archivos de MECA, esta ruta es: “*c:\Archivos de programas\Apache Group\Tomcat 4.1\webapps*” como se presenta en la Figura 22.

Figura 22 Indicación de la ruta



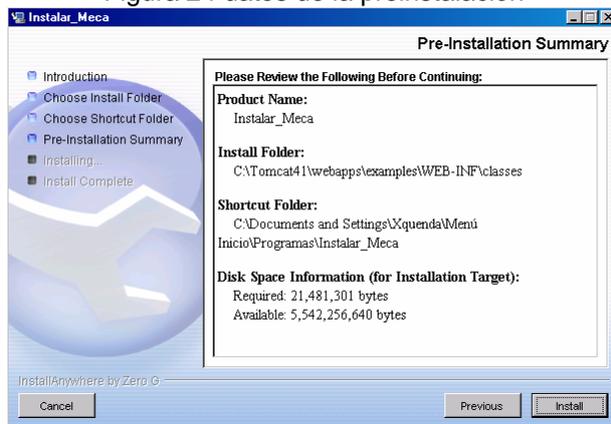
Posteriormente, se presenta la posibilidad de crear los íconos para el producto, (véase la Figura 23) esto es opcional ya que el Sistema MECA se instalará como servidor de evaluación y se accederá desde el navegador y no mediante un ícono de acceso directo.

Figura 23 elección de un acceso directo



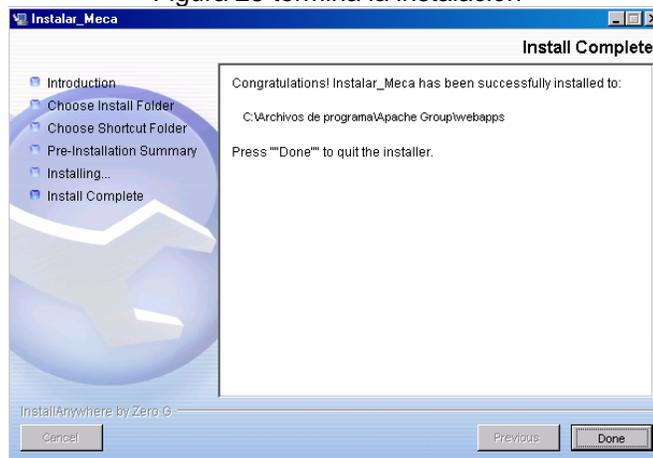
Se presenta la información de elección durante la configuración, como parte de la preinstalación, véase la Figura 24.

Figura 24 datos de la preinstalación



Posterior a este paso se inicia la instalación del Sistema MECA, este proceso es rápido y finaliza con la pantalla indicada en la Figura 25.

Figura 25 termina la instalación



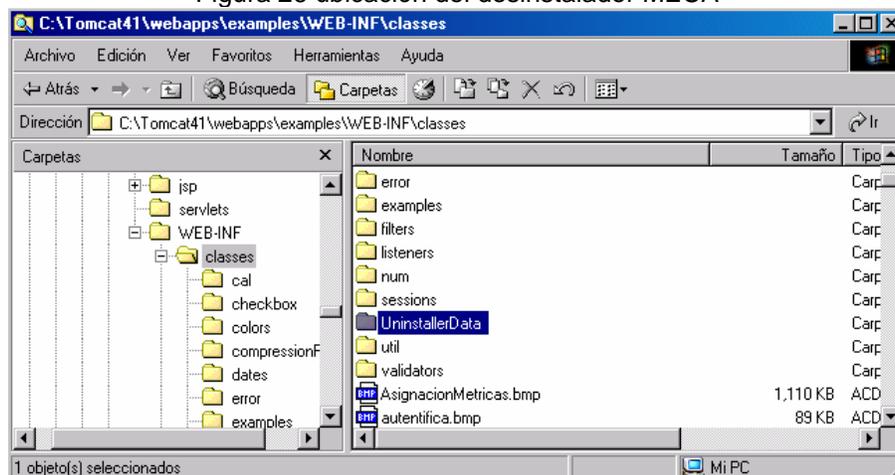
Se recomienda probar la instalación, para esto es necesario levantar el servidor de aplicaciones, en cualquiera de los dos modos indicados en el tema “Requerimientos para la instalación de los productos de software” (véase la Figura 17) y correr la aplicación de acuerdo al tema “Iniciación” de este Manual del usuario.

5 Desinstalación del Sistema MECA

El desinstalador del Sistema MECA se encuentra en la unidad C, específicamente en la ruta:

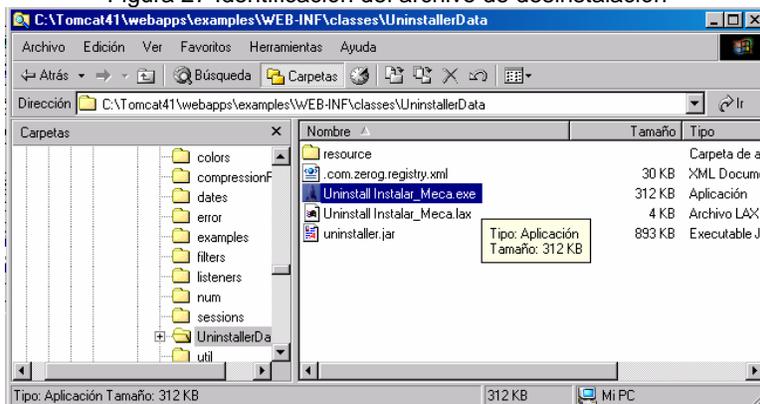
“C:\Tomcat41\webapps\examples\WEB-INF\classes” como se presenta en la Figura 26, el archivo ejecutable se llama “UninstallerData”.

Figura 26 ubicación del desinstalador MECA



En esta carpeta se encuentra el nombre del archivo a accionar: “Uninstall Instalar_Meca.exe” (véase la Figura 27).

Figura 27 Identificación del archivo de desinstalación



Al accionar el archivo de desinstalación aparece un mensaje de prevención indicando la remoción de los archivos creados en la instalación y no los posteriores a éste (véase la Figura 28).

Figura 28 Indicación de remoción de archivos



Después de indicar la orden de desinstalación, éste proceso da comienzo y presenta la pantalla final (véase la Figura 29).

Figura 29 Desinstalación exitosa



6 Relación de los documentos de soporte del software.

El Sistema MECA se encuentra grabado en un disco compacto, contiene un conjunto de archivos, tales como: los del sistema, el Manual del usuario, Instalación, desinstalación, instalación de Tomcat, J2SDK, SQL Server y Service Pack 3 (véase la Figura 30).

Figura 30 Lista de archivos del MECA

Instalar_Meca_Build_Output		Carpeta de archivos	04/05/2003 08:04 p...
Instalar_Mecalocales		Carpeta de archivos	04/05/2003 08:04 p...
Instalar_Meca.iap.xml	197 KB	InstallAnywhere Proj...	04/05/2003 08:04 p...
Manual de j2sdk.doc	4,104 KB	Documento de Micro...	04/05/2003 05:48 p...
Manual de Servicepack SP3.doc	1,758 KB	Documento de Micro...	04/05/2003 05:51 p...
Manual de SQL Server 2000.doc	6,280 KB	Documento de Micro...	04/05/2003 05:50 p...
Manual de Tomcat.doc	2,323 KB	Documento de Micro...	04/05/2003 06:12 p...
Manual del sistema MECA.doc	25,818 KB	Documento de Micro...	04/05/2003 11:51 p...
LEEME.txt	7 KB	Documento de texto	04/05/2003 11:54 p...

7 Requerimientos del ambiente de operación.

Es importante contar con los requerimientos mínimos especificados para la instalación y manejo del Sistema MECA.

7.1 Hardware.

Referente al servidor: Reconociendo como servidor a la máquina que proporcionará el funcionamiento de la evaluación a través de internet, sus elementos son:

- procesador pentium III 450 MHz
- memoria RAM 256 Mb
- disco duro de 6.5 Gb
- tarjeta red Ethernet 10/100

Referente al cliente: Reconociendo como cliente a una máquina que solicita el servicio de evaluación al servidor a través de Internet, sus elementos son:

- Procesador pentium
- memoria RAM 64 Mb
- tarjeta red Ethernet 10/100

7.2 Software.

A continuación se especifica el nombre y versión de las aplicaciones con las cuales se han realizado las pruebas de evaluación de la herramienta.

Referente al servidor:

- Jakarta Tomcat 1.4.1.como servidor de aplicaciones WEB
- Iexplorer 4.0. o posterior
- SQL Servler 2000
- Windows 2000 Profesional
- J2sdk 1.4.1 en la cual se han programado los Servlets, que se comunican con la Base de Datos vía el puente JDBC.

Referente al cliente:

- Iexplorer 4.0 o posterior

- Windows 98

7.3 Organizativo.

Toda empresa o institución gubernamental o no gubernamental, de cualquier tipo o giro social y tamaño podrá hacer uso del Sistema MECA. El requerimiento mínimo que debe cubrir el ambiente organizativo es la integración y capacitación del Comité Evaluador con respecto a la Metodología MECA, así como temas básicos de Auditoría Informática e Ingeniería del Software.

8 Procedimiento para leer el contenido del archivo LEEME.TXT.

El archivo LEEME.TXT se encuentra en el disco compacto de instalación del Sistema MECA y podrá leerse mediante cualquier editor de texto o código ASCII.

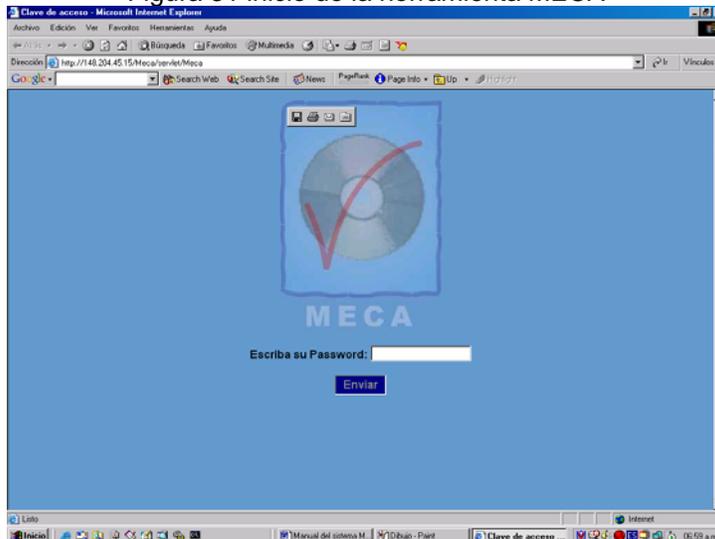
9 Procedimiento para la creación de las copias de respaldo o de trabajo.

Este procedimiento está regido de acuerdo a la licencia que indiquen las autoridades del CIC-IPN. Para la creación de copias del disco de instalación es preciso contar con un sistema copiador comercial (hay varios en el mercado, tales como Easy Creator, Nero Burning, etc), con respecto a la creación de respaldo a la Base de Datos se recomienda leer un manual avanzado de SQL Server, existen varios en Internet específicamente en la página de la compañía Microsoft.

10 Iniciación.

Trata del corrimiento de la aplicación. La herramienta MECA se ejecuta desde el Internet Explorer, se debe teclear la siguiente dirección en la parte superior de la ventana: <http://localhost:8080/Meca/servlet/Meca> existe otra forma de levantar el servidor encargado de la administración y evaluación del sistema, tecleando <http://127.0.0.1:8080/Meca/servlet/Meca> o en su defecto, tipeando la IP o Nombre lógico del equipo si se cuenta con un servidor de nombres de dominio, como ejemplo: <http://IP:8080/Meca/servlet/Meca> o bien; <http://Nombre:8080/Meca/servlet/Meca> la respuesta se presenta en la Figura 31.

Figura 31 inicio de la herramienta MECA



11 Utilización o Entrenamiento.

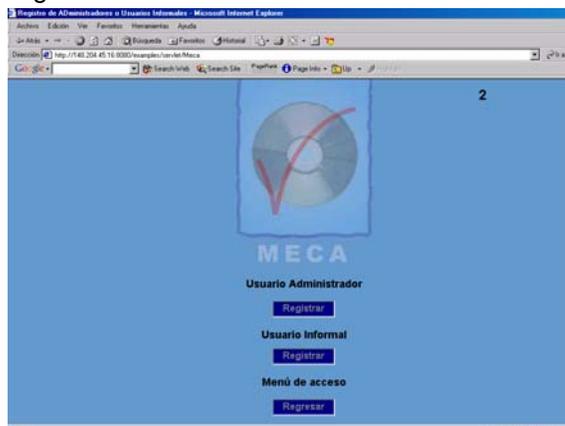
Existen cuatro tipos de Usuarios:

- Administrador
- Evaluador Formal
- Evaluador Informal
- Evaluador de la Calidad en Uso

Se explica la utilización de la herramienta MECA mediante el usuario **Administrador**.

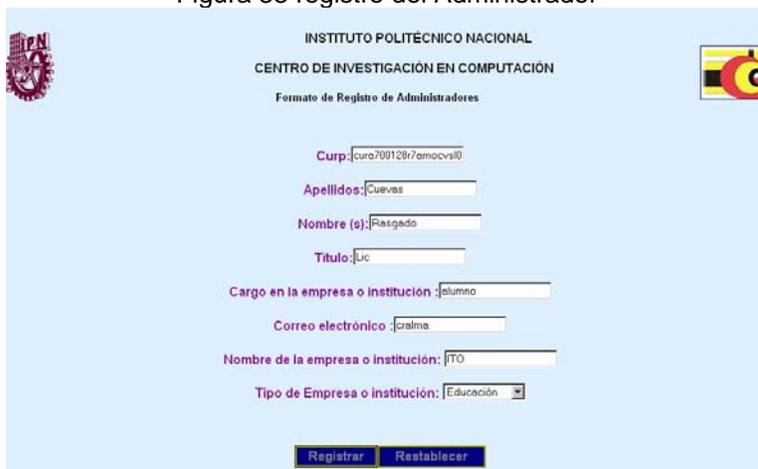
Suponiendo que el Administrador no cuenta con una clave de acceso, se pulsa *Intro*. El Sistema MECA presenta los tipos de usuarios (véase la Figura 32).

Figura 32 acceso del Administrador sin clave asignada



El Usuario elige el botón *Administrador* y se registra en el sistema ingresando sus datos generales, como se observa en la Figura 33.

Figura 33 registro del Administrador



INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN
Formato de Registro de Administradores

Corp:

Apellidos:

Nombre (s):

Título:

Cargo en la empresa o institución:

Correo electrónico:

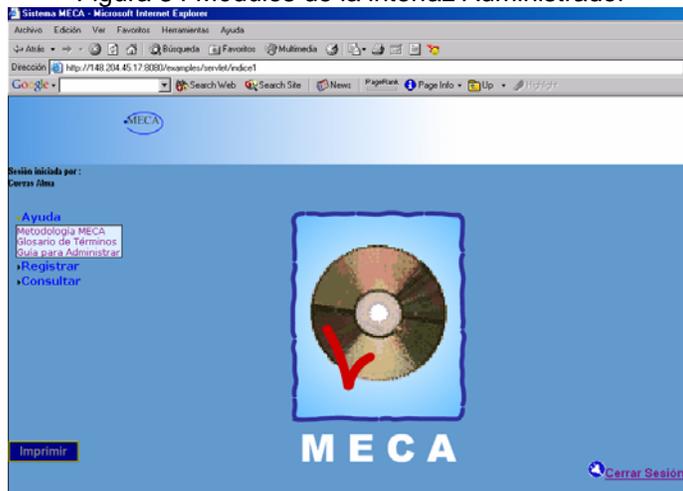
Nombre de la empresa o institución:

Tipo de Empresa o institución:

Al ingresar los datos del Administrador, el sistema le devuelve la clave de acceso para futuros ingresos. La interfaz del Administrador está integrada por los siguientes cuatro módulos (véase la Figura 34):

- Ayuda.
- Registrar.
- Consultar.

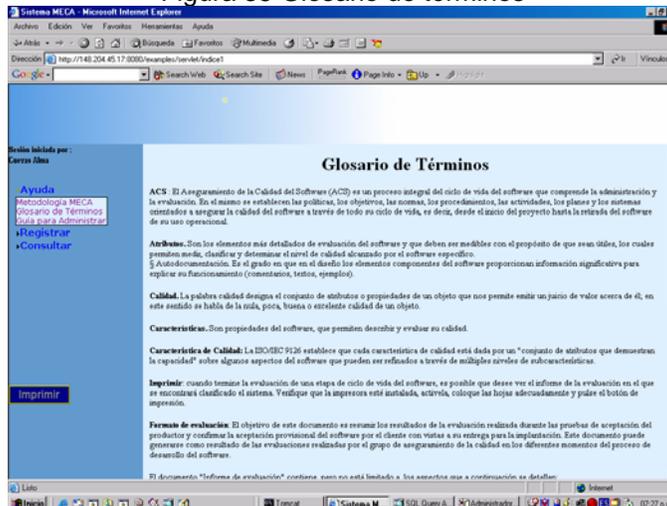
Figura 34 Módulos de la interfaz Administrador



Ayuda.

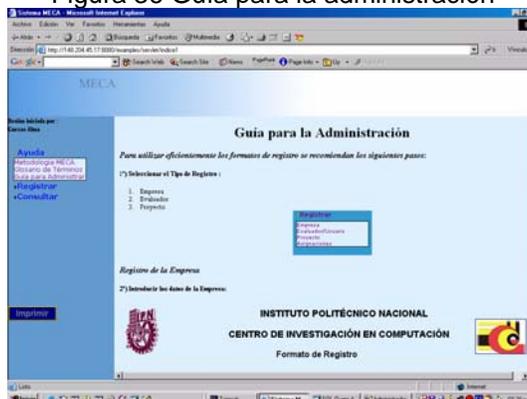
Esta opción presenta tres submódulos: la Metodología MECA, el Glosario de términos y la Guía para administrar. En la Figura 35 se observa el Glosario de términos.

Figura 35 Glosario de términos



Se presenta también una Guía para el Administrador con la finalidad de cubrir los aspectos a detalle que constituyen la tarea del Administrador. (véase la Figura 36).

Figura 36 Guía para la administración



Registrar.

Este es el módulo que le da sentido a la tarea de administrar la Calidad del Software, consiste en registrar Evaluadores, Proyectos y las asignaciones entre estos. En la Figura 37 se observa el registro de Evaluadores.

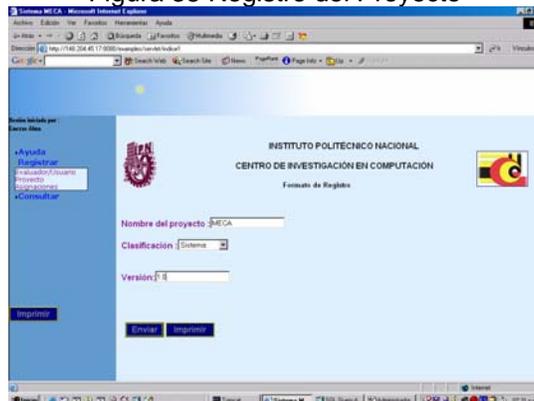
Figura 37 registro del Evaluador



The screenshot shows a web browser window displaying the MECA system interface. The page title is "MECA" and the header includes "INSTITUTO POLITÉCNICO NACIONAL" and "CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN". The main content area is titled "Formulario de Registro" and contains several input fields: "Carp" (with value "jsuman"), "Apellidos" (with value "suman"), "Nombre Completo" (with value "jsuman"), "Titulo" (with value " "), "Cargo en la empresa" (with value "Docente Investigador"), "Institución de procedencia" (with value "CIC"), "Tipo de evaluador" (with value "Presidente"), and "Correo Electrónico" (with value " "). A sidebar on the left contains navigation links: "Ayuda", "Registrar", "Proyecto", "Evaluadores", and "Consultar".

Si siguiendo con las tareas del Administrador, se presenta la Figura 38 en la cual se registran los datos del Proyecto.

Figura 38 Registro del Proyecto

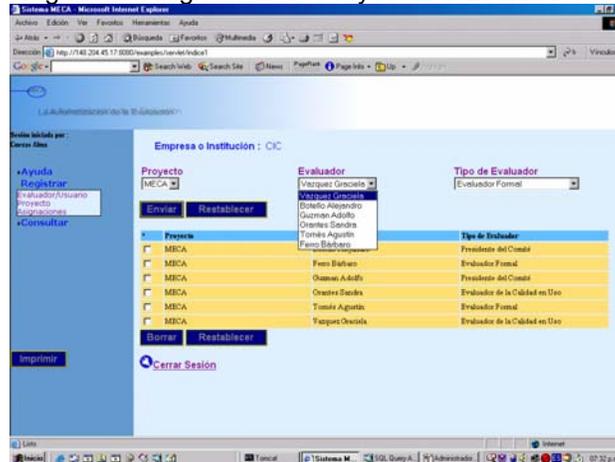


The screenshot shows the same MECA system interface as Figure 37, but with the "Formulario de Registro" fields populated with project information: "Nombre del proyecto" (with value "MECA"), "Clasificación" (with value "Sistema"), and "Version" (with value "1"). The "Enviar" and "Cancelar" buttons are visible at the bottom of the form area.

Asignaciones.

Este proceso consiste en proporcionar Proyectos a los Evaluadores del comité (véase Figura 39) es preciso ubicar el cursor en el menú del Proyecto, Evaluador o Tipo de Evaluador para observar y elegir los Proyectos, Evaluadores y los Tipos de éstos, que presenta el sistema. En caso de haber sucedido un error en la asignación, el sistema presenta una opción de borrado con solo ubicarse en la fila y en la primera columna errónea, marcarlo (aparece una flecha) y pulsar la opción Enviar. Posterior al paso de *Asignaciones*, el Administrador da las claves de acceso a cada Evaluador con la finalidad de que estos puedan acceder al sistema.

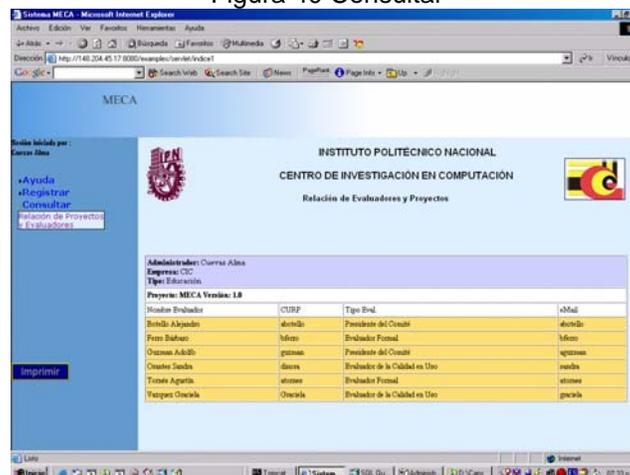
Figura 39 Asignación de Proyectos a Evaluadores



Consultar.

Para comprobar la asignación anterior, el sistema presenta una opción de Consultar las asignaciones, en la Figura 40 se presenta la pantalla que presenta la Relación entre Proyectos y Evaluadores.

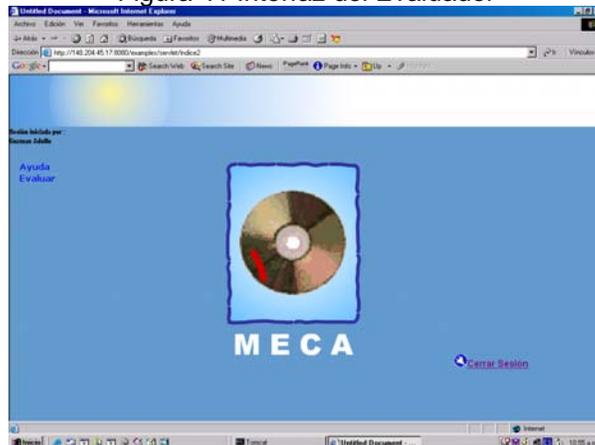
Figura 40 Consultar



A continuación, se explica la utilización de la herramienta MECA mediante el usuario **Evaluador Formal**. Este usuario se clasifica en: Presidente y Miembro del Comité Evaluador. Desde la perspectiva del sistema la diferencia entre estos usuarios es: El Presidente puede solicitar el promedio del comité. Desde la perspectiva de la evaluación de la Calidad den Software es el Presidente quien convocará a una reunión del comité para concentrar los resultados por Evaluador y decidir el dictamen final.

El sistema presenta una interfaz por cada Usuario, la interfaz del Evaluador Formal se presenta en la Figura 41.

Figura 41 Interfaz del Evaluador



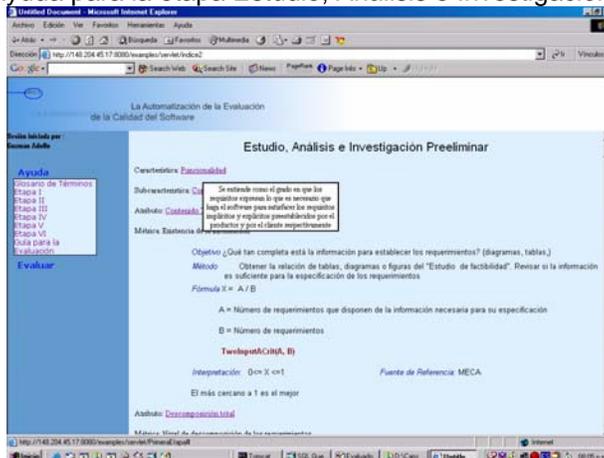
La interfaz del Evaluador está integrada por los dos módulos siguientes

- Ayuda.
- Evaluar.

Ayuda.

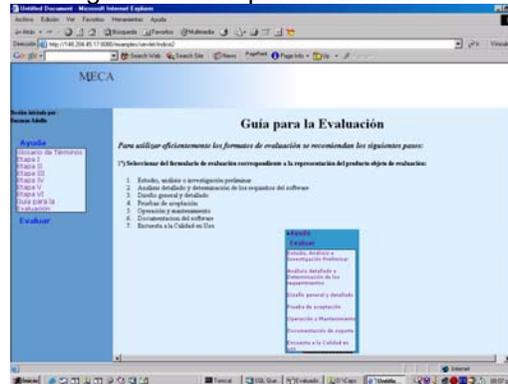
Este módulo cuenta con un submódulo que ya se ha presentado en la interfaz Administrador y tiene la misma función (Glosario de términos), además proporciona la Metodología MECA distribuída a través de las seis etapas en la cual, se presenta el concepto, el objetivo, el método y la fórmula de cada elemento de evaluación en cada etapa. En la Figura 42 se observa la opción de Ayuda para la etapa Estudio, Análisis e Investigación Preliminar.

Figura 42 Ayuda para la etapa Estudio, Análisis e Investigación Preliminar



El sistema presenta una Guía para la Evaluación (véase Figura 43) con la finalidad de minimizar el índice de fallas que pueda ocasionarse durante este proceso. Se recomienda la lectura de este submódulo antes de proceder a una evaluación; así mismo mantiene un enlace con la información de la Metodología MECA correspondiente a la etapa evaluada.

Figura 43 Guía para la Evaluación



Se recomienda usar el botón de *Atrás* y *Adelante* o *Back* y *Forward* para apoyarse en el módulo de la ayuda durante la evaluación, como se presenta en la Figura 44.

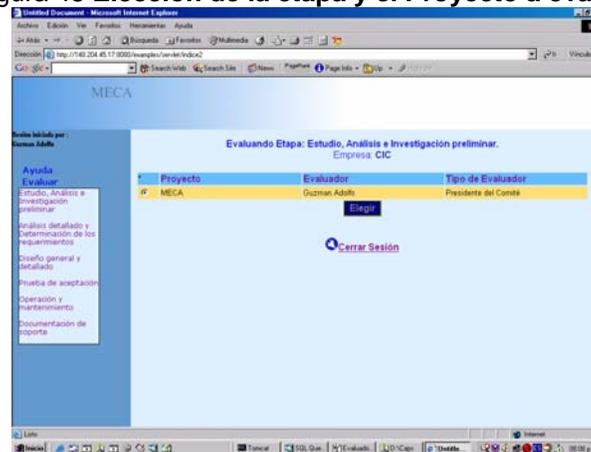
Figura 44 Navegación entre las pantallas de evaluación y ayuda



Evaluar.

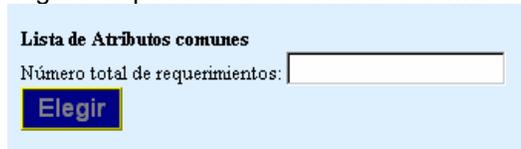
Este proceso es la razón de la implementación de la Metodología MECA. Se presenta la opción de evaluar las seis etapas del ciclo de vida del software, el Evaluador puede elegir una de estas etapas. En la Figura 45 se presenta la pantalla correspondiente a la elección de la etapa y el Proyecto.

Figura 45 Elección de la etapa y el Proyecto a evaluar



El sistema identifica los atributos comunes, como aquellos que se repiten durante la etapa. Para evitar que el Evaluador proporcione repetidamente el valor de estos atributos, el sistema pide estos valores al inicio de la evaluación, la pantalla correspondiente a este proceso se presenta en la Figura 46 en la cual se piden los atributos comunes para la etapa Estudio, Análisis e Investigación Preliminar. El número de estos atributos varía entre una y otra etapa.

Figura 46 petición de los atributos comunes



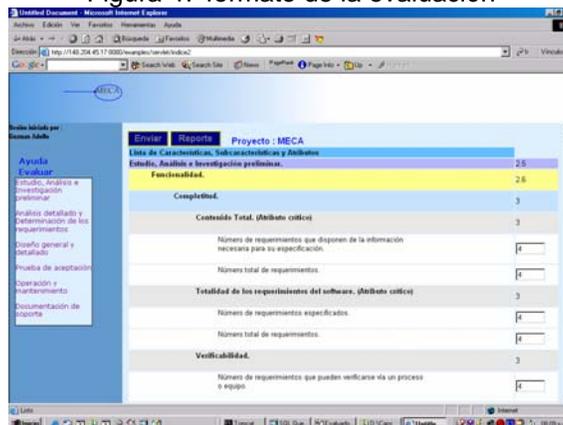
Lista de Atributos comunes

Número total de requerimientos:

Elegir

Posteriormente, se presenta el formato de la evaluación en la cual, el Evaluador procede a introducir el valor de los parámetros de acuerdo a su análisis con respecto a los requerimientos del software evaluado. Al pulsar la opción Enviar, el sistema registra los valores y devuelve una calificación (véase la Figura 47).

Figura 47 formato de la evaluación



Proyecto: MECA

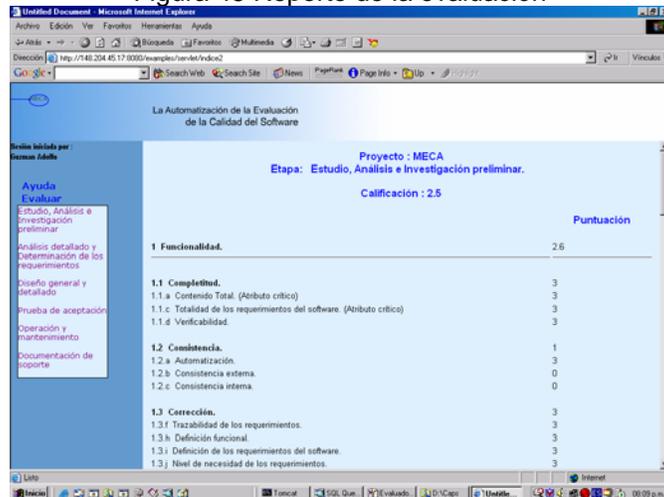
Lista de Características, Subcaracterísticas y Atributos

Característica	Puntuación
Estudio, Análisis e Investigación preliminar	2.5
Funcionalidad	2.6
Complejidad	3
Contenido Total (Atributos críticos)	3
Número de requerimientos que disponen de la información necesaria para su especificación	
<input type="text" value="4"/>	
Número total de requerimientos	
<input type="text" value="4"/>	
Totalidad de los requerimientos del software (Atributos críticos)	
3	
Número de requerimientos especificados	
<input type="text" value="4"/>	
Número total de requerimientos	
<input type="text" value="4"/>	
Verificabilidad	
3	
Número de requerimientos que pueden verificarse via un proceso o equipo	
<input type="text" value="4"/>	

Es posible enviar los datos aún cuando no se haya terminado la asignación de todos los parámetros de la evaluación, en este caso, el Evaluador deberá asignar ceros a cada casilla faltante, el sistema presenta un informe temporal (en muchos casos con una calificación menor a la realidad).

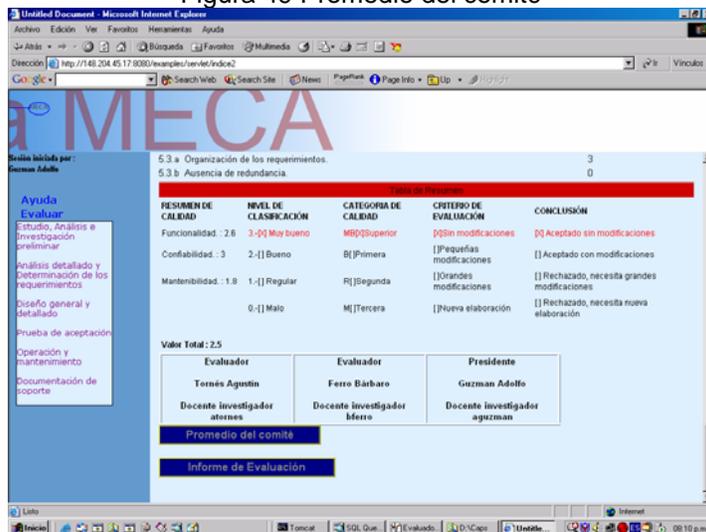
Posterior al ingreso de los parámetros y a la selección del botón *Enviar*, el sistema podrá proporcionar un Reporte de la evaluación pulsando el botón *Reporte* como se observa en la Figura 48.

Figura 48 Reporte de la evaluación



En la Figura 49 se presenta la diferencia entre el Evaluador Presidente y Miembro. Al pulsar el botón *Promedio del comité*, el sistema solicita la clave de acceso del Presidente, lo verifica y le proporciona el promedio de las evaluaciones del comité.

Figura 49 Promedio del comité



Posterior a este proceso se presenta un dictamen: reporte o informe de evaluación del software (véase la Figura 50).

Figura 50 informe de la evaluación

RESUMEN DE CARGO	NIVEL DE CLASIFICACION	CATEGORIA DE CALIDAD	CRITERIO DE EVALUACION	CONCLUSION
Funcionalidad : 2.6	3-3) Muy bueno	M)Quinto	D)Sin modificaciones	[X] Aceptado sin modificaciones
Confiabledad : 2	2-1) Bueno	R)Primera	I)Pequeñas modificaciones	[] Aceptado con modificaciones
Mantenibilidad : 1.8	1-1) Regular	R)Segunda	J)Grandes modificaciones	[] Rechazado, necesita grandes modificaciones
	0-1) Malo	M)Tercera	K)Nueva elaboración	[] Rechazado, necesita nueva elaboración

Valor Total: 2.5

Evaluador Teres Aguiar Docente investigador atomas	Evaluador Ferro Bárbaro Docente investigador bferro	Presidente Gezman Adolfo Docente investigador aguzman
--	---	---

Promedio del comité

Informe de Evaluación

Posteriormente, el sistema presenta un dictamen en el cual, clasifica el software de acuerdo a sus parámetros de evaluación de la calidad.

A continuación, se explica la utilización de la herramienta MECA mediante el usuario **Evaluador Informal**. La diferencia entre el Evaluador Formal y el Informal es que el segundo no cuenta con una designación oficial ni es miembro de un Comité Evaluador de la Calidad del Software, por lo cual, no cuenta con una clave de acceso al sistema. Éste es un navegante de Internet que desea saber la calidad de un producto y accede al Sistema MECA tecleando *Intro* en la pantalla inicial de acceso. El sistema solicita el tipo de Usuario, por lo que elige ser Evaluador Informal, en este caso el sistema le presenta la pantalla de registro (véase la Figura 51) en la cual, hay un mensaje de error en la parte superior indicando que el usuario no existe.

Figura 51 pantalla de registro del Evaluador Informal

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Formateo de Registro
No existe el usuario
Formateo de Registro Informal

SI NO desea registrarse como Evaluador Informal pulse: [Regresar](#)

Correo Electrónico: [input type="text"]

Nombre del Proyecto: [input type="text"]

El Evaluador Informal proporciona estos datos, por lo que el sistema los procesa y devuelve una clave de acceso (véase la Figura 52) a partir del cual el usuario podrá acceder al sistema, de olvidar esta clave, será necesario un nuevo registro

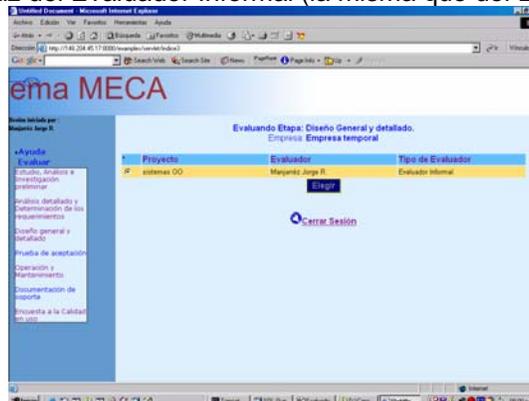
iniciando nuevamente el proceso de evaluación previamente almacenados en la Base de datos MECA.

Figura 52 proporción de una clave de acceso al Evaluador Informal



posteriormente, el sistema presenta la interfaz del Evaluador Informal, ésta es la misma que presenta al Evaluador Formal (véase Figura 53).

Figura 53 interfaz del Evaluador Informal (la misma que del Evaluador Formal)



Se explica la utilización de la herramienta MECA mediante el usuario **Evaluador de la Calidad en Uso**. Al introducir la clave de acceso, el sistema presenta la pantalla que se observa en la Figura 54.

Figura 54 Pantalla de bienvenida del Evaluador de la Calidad en Uso



La interfaz del Evaluador de la Calidad en Uso está integrada por los dos módulos siguientes

- Ayuda.
- Evaluar.

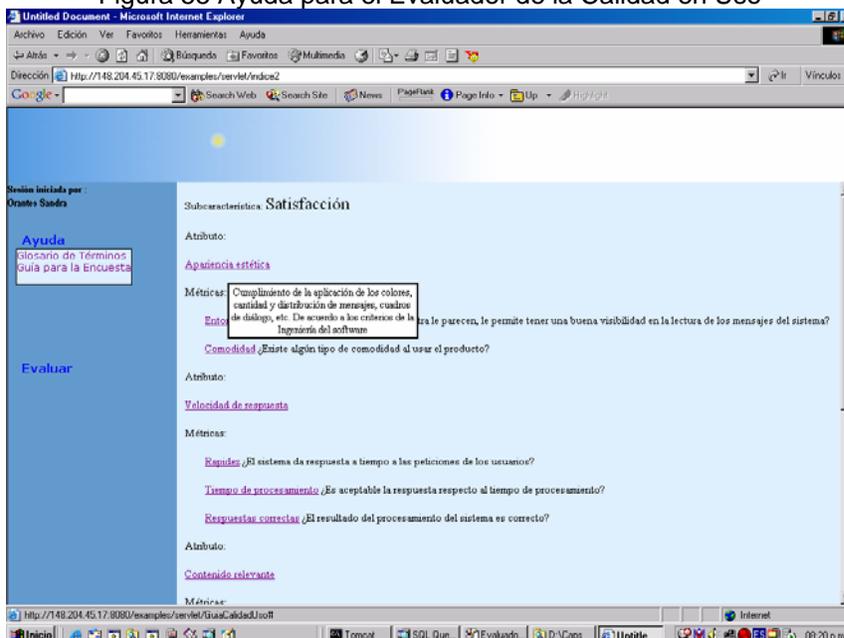
Ayuda.

Este módulo esta integrado por el *Glosario de Términos* y la *Guía para la encuesta*. Dado que el *Glosario de términos* es el mismo en cada tipo de evaluador, se explicará la *Guía para la encuesta*.

Guía para la encuesta.

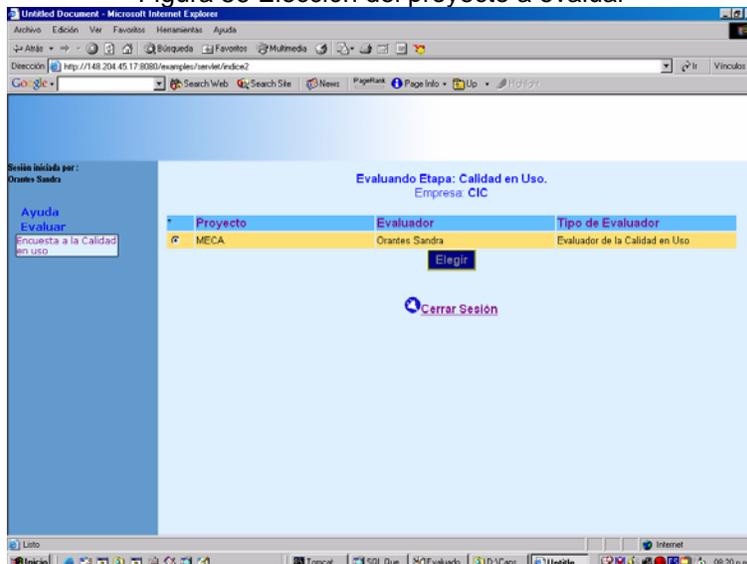
Este módulo proporciona una explicación de cada una de las preguntas de la encuesta, sirve de guía para el Evaluador (véase la Figura 55).

Figura 55 Ayuda para el Evaluador de la Calidad en Uso



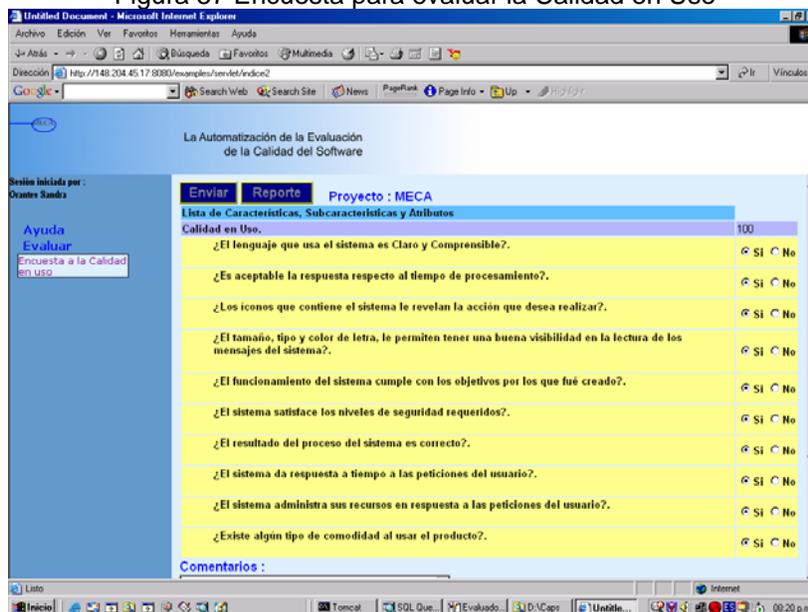
Al igual que el Evaluador Formal e Informal, el de la Calidad en Uso debe elegir primeramente el proyecto a evaluar, en este proceso se observarán todos los proyectos o software que le haya asignado su Administrador (véase la Figura 56).

Figura 56 Elección del proyecto a evaluar



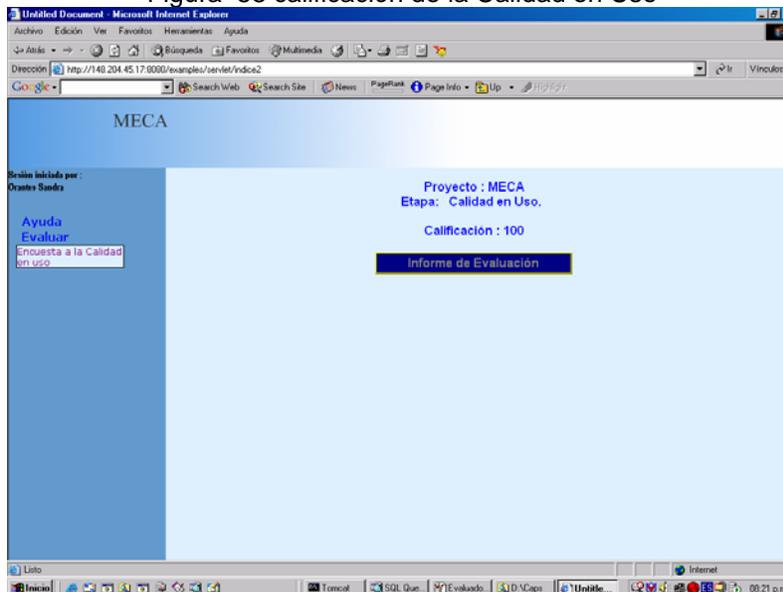
Posteriormente, se da inicio a la resolución de la encuesta, la pantalla que se observa en la Figura 57 presenta estas preguntas.

Figura 57 Encuesta para evaluar la Calidad en Uso



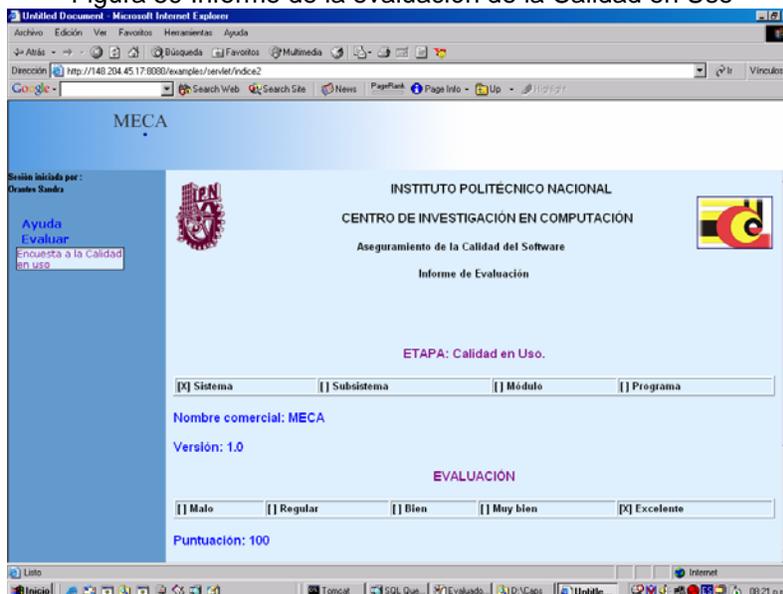
Al finalizar la encuesta es preciso indicar el grabado de la información accionando el botón *Enviar*, la respuesta del sistema es el de presentar la calificación de la Calidad en Uso (véase la Figura 58).

Figura 58 calificación de la Calidad en Uso



El sistema presenta un informe de la evaluación, en el cual se observa, entre otros datos, la clasificación del producto evaluado (véase la Figura 59).

Figura 59 Informe de la evaluación de la Calidad en Uso



Apéndice A

Glosario

Analizador de Consultas. Herramienta de SQL Server para realizar las transacciones de negocio

Master. Base de Datos por Defecto de SQL Server

Navegante de Internet. Persona que usa la red Internet para viajar entre varios servidores de información.

Script. Código o Lenguaje SQL ejecutado regularmente en el Analizador de Consultas.

Apéndice B. Relación de todas las abreviaturas

MECA. Modelo Cualimétrico para la Evaluación de la Calidad del Software

ISO/IEC 9126. (the International Organization for Standardization Organización Internacional para la Estandarización)

UML (Unified Modeling Language, Lenguaje de Modelado Unificado)

SQL Server. Servidor de Lenguaje de Consulta Estructurado (Structured Query Language)

J2SDK. Java segunda versión, Paquete de Desarrollo de Software (Software Development Kit)

Tomcat. servidor de aplicaciones de Apache

CIC. Centro de Investigación en Computación

IPN. Instituto Politécnico Nacional

IE Explorer 4.0. Internet Explorer Navegador de Microsoft

F5. Tecla que ejecuta un Script en el Analizador de Consultas

DNS. System Data Sources (Fuentes de Datos del Sistema)

ODBC. Conectividad Abierta de Bases de Datos (Open Data Base Connectivity)

Apéndice C. Relación de todos los mensajes de error

Acceso negado.

Este mensaje surge cuando no se ha tecleado correctamente la clave de acceso del Presidente al consultar el promedio del comité.

Botón de Errores.

En caso de ocurrir errores durante la evaluación, el sistema presenta un botón de errores en el Reporte de la evaluación. El Evaluador puede pulsar este botón para conocer los errores cometidos durante el proceso, este botón no aparecerá si se ha desarrollado un proceso correcto de evaluación.

Error: Ausencia de atributo crítico.

Otra de las causas de error en el sistema es cuando se asigna un valor cero a un parámetro que es crítico. Entre el conjunto de atributos que conforman algunas subcaracterísticas de calidad del software existen algunos que son críticos por su influencia determinante en la calidad aceptable, ya que su ausencia o no cumplimiento ocasiona una pérdida de la calidad mínima requerida por todo software desarrollado profesionalmente, lo cual invalida automáticamente la aceptación de cualquier representación del software objeto de evaluación. Por tanto estos atributos críticos son obligatorios, es decir, tienen que cumplirse como condición indispensable para la aceptación del software. Los atributos críticos son evaluados solamente con dos únicos valores:

0 significa NO cumplimiento (ausencia)

3 significa SI cumplimiento (presencia)

Si durante el proceso de evaluación el sistema detecta la ausencia de al menos un atributo crítico asociado a una determinada subcaracterística de calidad del software, el sistema enviará este mensaje como un error en el botón de errores, por lo que promediará como cero la característica relacionada a dicha subcaracterística. Tal situación provocará una calificación baja en el software objeto de evaluación mientras no sean corregidos los defectos asociados a la falta de los atributos críticos.

Error al Insertar, El usuario no puede ser asignado como Evaluador Formal o Presidente.

El error ocurre cuando se quiere asignar como Evaluador Formal a un Evaluador de la Calidad en Uso, dado que el Administrador elige al usuario de la Calidad en Uso al cliente final, este regularmente no tiene las bases para ser Evaluador Formal (Presidente o Miembro de un Comité Evaluador).

Error al Insertar, ya existe un Presidente asignado en el Proyecto.

Este error ocurre al asignar dos Presidentes al mismo Proyecto, en la misma empresa y registrados con el mismo Administrador. Se recomienda verificar la asignación.

Error: El primer valor debe ser menor o igual al segundo.

Cuando el Evaluador no analiza adecuadamente el valor de los parámetros, (véase la Figura C.1) y los ingresa de acuerdo al ejemplo, el sistema identifica el error, en este caso se recomienda un mejor análisis de los valores del elemento evaluado.

Figura C.1 Error en parámetros

Contenido Total. (Atributo crítico)	
	3
Error: El primer valor debe ser menor o igual al segundo	
Número de requerimientos que disponen de la información necesaria para su especificación.	8
Número total de requerimientos.	4

Error: Parámetros no numéricos, ingrese sólo valores numéricos.

Si el Evaluador no introduce un valor en el formato, o introduce una letra en lugar de un valor numérico, el sistema presentará el siguiente mensaje de Error: "Error: Parámetros no numéricos, ingrese sólo valores numéricos".

No existe el usuario.

Este mensaje surge al ingresar al sistema con *Intro* y elegir ser Usuario Informal; acto seguido, el sistema presenta un formato de registro del usuario.

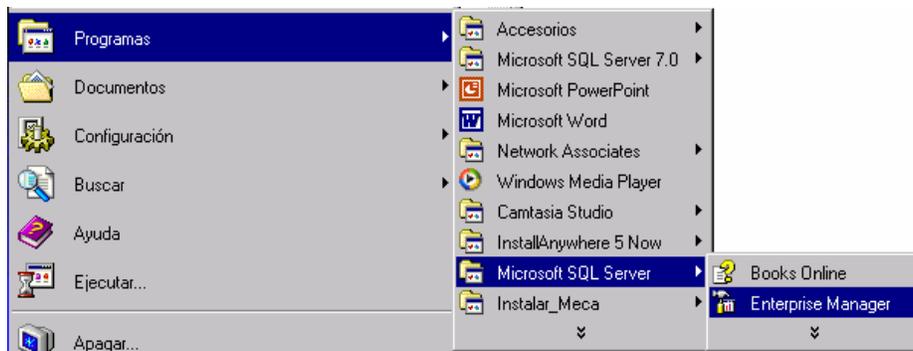
Sistema MECA

Manual para restaurar la Base de Datos

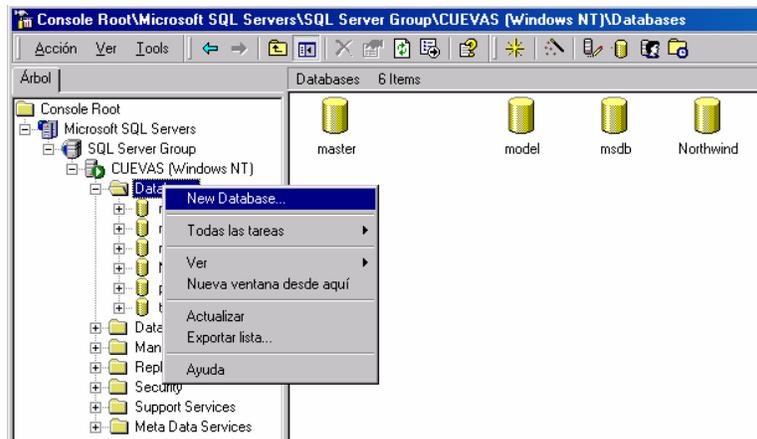
La base de datos MECA se encuentra en la carpeta Base de datos, el archivo de respaldo se llama "bd_meca_18jun2003". Antes de realizar este proceso, es necesario tener definido el ODBC llamado odbc_meca.

A continuación se siguen los pasos:

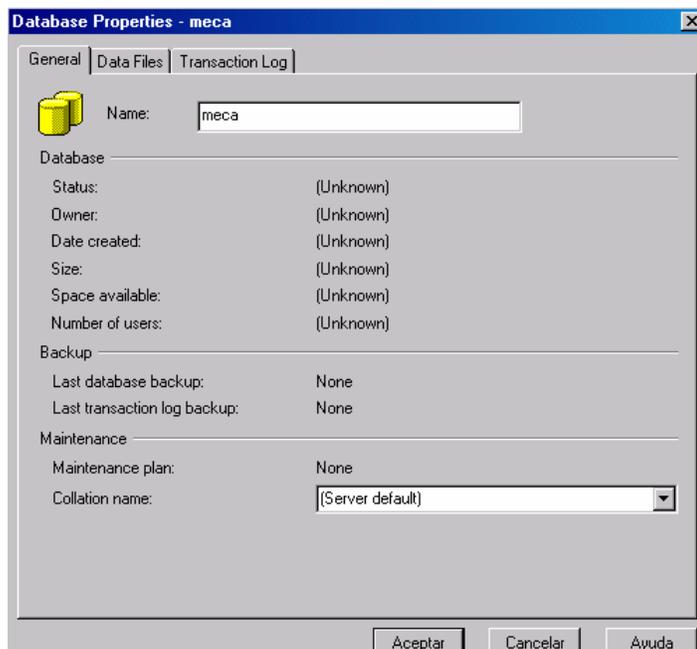
1. Abrir el Enterprise Manager del SQL Server 2000. Este modulo se instala automáticamente durante la instalación del SQL Server 2000.



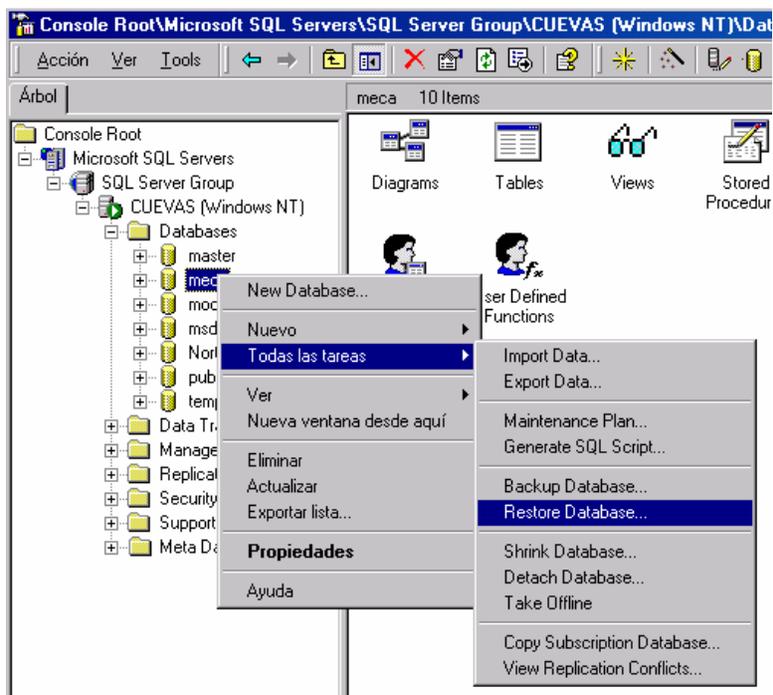
2. Haciendo clic derecho con el ratón en databases se elige crear una nueva Base de datos, llamada meca



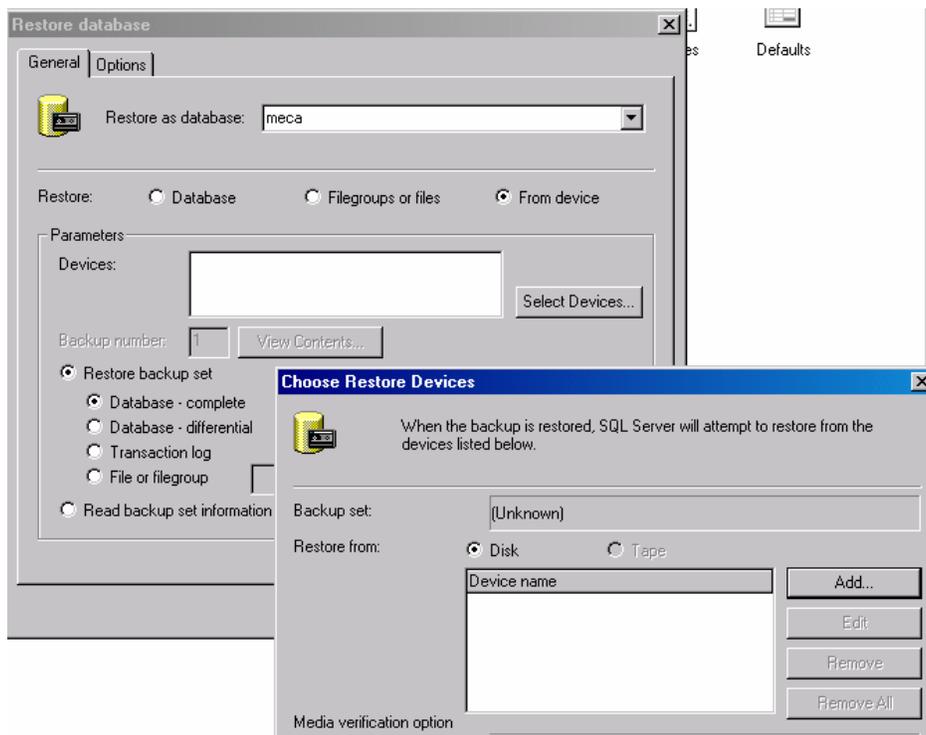
3. llamado meca



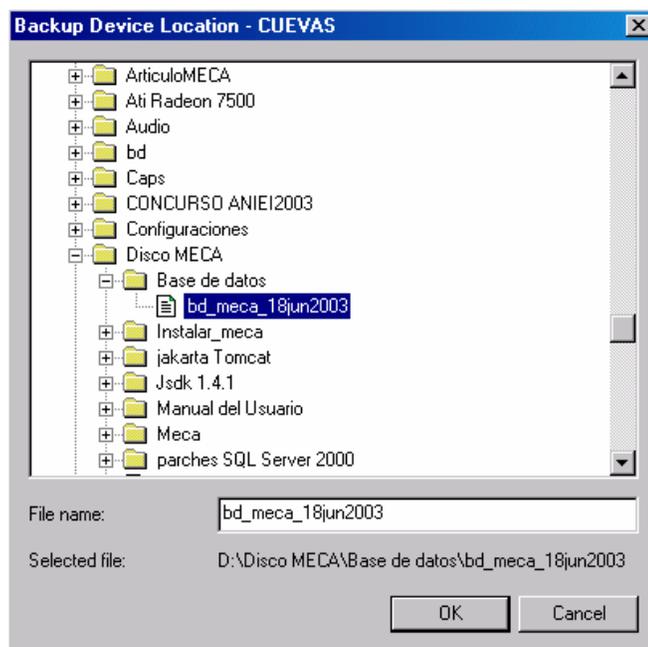
4. Una vez creada meca , clic derecho en meca y elegir Todas las tareas, Restore Database.



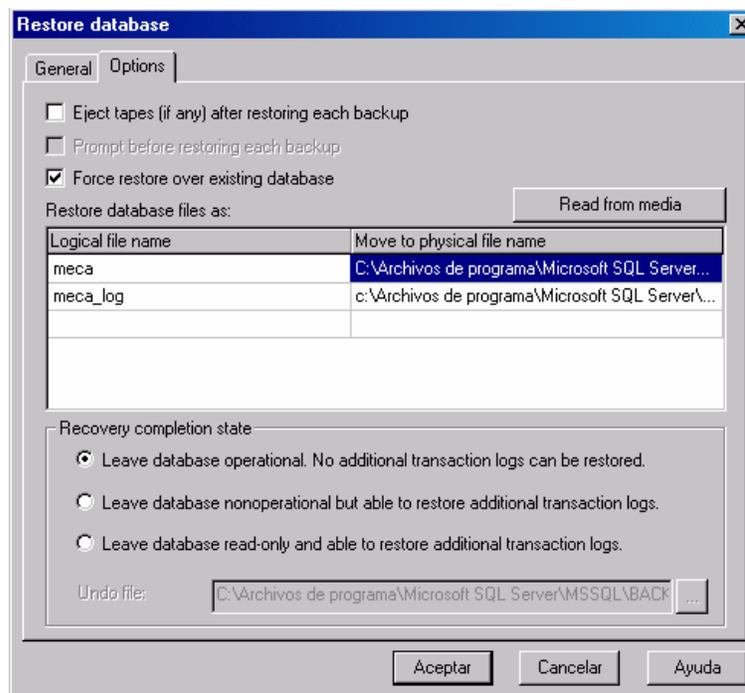
5. indicar el nombre de la Base de datos a restaurar, elegir Select Devices y teclear Add.



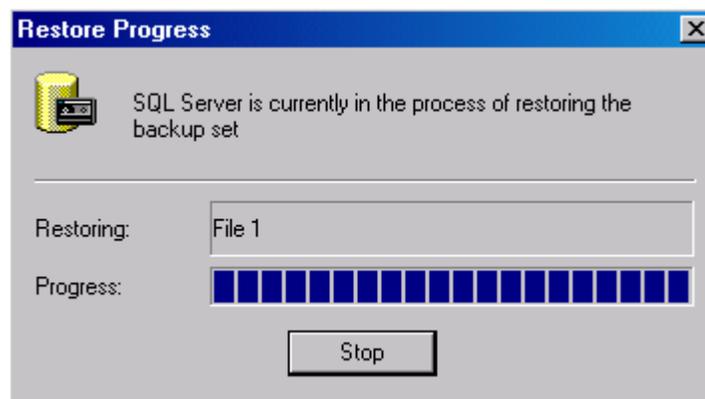
6. Indicar el lugar donde se encuentra el archivo a restaurar, en este caso se llama "bd_meca_18jun2003".



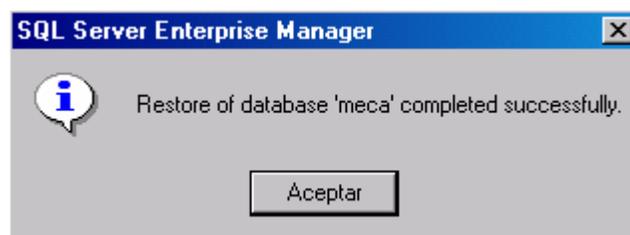
7. teclear OK, OK, no olvidarse de elegir la pestaña Option de Restore database y checar la casilla "Force restore over existing database"



8. Automáticamente inicia el proceso de restauración.



9. el resultado es el siguiente si se realizó correctamente los pasos de este manual.



10. De otra manera verificar el problema, o bien borra la base de datos (desde el Enterprise Manager) y volver al paso 1.

11. Ahora podrá corroborar el contenido de la Base de datos MECA desde el Procesador de consultas del SQL Server 2000.