

Universidad de las Ciencias Informáticas

“ERP”



Título: Desarrollo de un componente para la configuración de aspectos arquitectónicos e integración de componentes de la arquitectura de Cedrux.

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores:

Michael Hernández Martínez

David Vargas Hernández

Tutores:

Ing. Yoandry Morejón Borbón

Ing. René Lazo Ochoa

Ciudad de la Habana

Junio, 2009

Agradecimientos

Antes que nada a la persona que ha sido mi faro guía durante todos estos largos años, con su amor y dedicación, mi madre.

A mi padre, viejo gracias por enseñarme a ser un hombre de bien, este resultado también es tuyo.

A Grisel (Nena) y Guillermo, ustedes han sido uno de los pilares fundamentales durante toda mi vida, Nena a ti te debo la vida, gracias por los consejos y regaños.

A tío Alberto y tía Maritza, quienes han sido mis segundos padres, brindándome su cariño y apoyo en todo momento.

A tía Martha, gracias de corazón por todo.

A mis primos Leydis, Alberyuri y Albertico, gracias por sus consejos.

A mi hermano Duni, por ser siempre esa mano derecha junto a la mía.

A mi querida Dayi por tanto sacrificio, gracias por estar a mi lado todo este tiempo, este logro es tanto mío como tuyo.

A mi amigo y hermano David, sin tí hubiese sido imposible obtener este resultado, espero no haberte defraudado.

A mi tutor Rene, gracias por enseñarme a ser un profesional, y como se desarrolla verdaderamente junto a un equipo, tus regaños sirvieron de mucho.

A mi tutor Borbon (Bb), gracias por más que un tutor ser un amigo.

A Iliana, suegra gracias por tus consejos y por los que aún faltan.

A Ana María, gracias por todo el apoyo y la grandeza que te cubre, eres incomparable.

A mis amigos y compañeros del proyecto, por el apoyo de cada uno de vosotros.

A mis familiares a los que nunca defraudaré jamás.

A todas las personas que me han brindado su apoyo y han creído en mí durante todos estos largos años, de corazón un millón de gracias.

Como hubiera podido llegar a donde estoy sin saber que llevo conmigo el orgullo de los dos grandes ingenieros que son mis padres. Se que la meta es alta, pero confío llegar a alcanzarla algún día.

A mi familia, gracias por haber allanado el camino para hacer mi carrera sobre el. El ingeniero perfecto podría formarse tomando las virtudes que me enseñaron y son parte de Uds.

A Uds. que más que tutores han sido amigos y más que amigos hermanos, y no solo por la edad si no por todas las horas de trabajo empeñadas juntos, gracias por sus consejos, gracias por estar ahí.

A todos los amigos que durante todos estos años de estudios han hecho más largas las horas de azueto y mostrado su mano en los momentos difíciles. Este triunfo también es de Uds.

A mis profes, por haberme enseñado a aprender. Muchos podrán ser los conocimientos olvidados después de los años pero la metodología del aprendizaje siempre quedará, y eso se lo debo a Uds.

A todo el colectivo del ERP, de arquitectura, mi nueva familia, los pínos nuevos. Nunca se verán otros tan unidos por el esfuerzo y la amistad como los que aquí se formaron.

Muy difícil sería pensar en estos 5 años sin acordarme de tí Daisy. Que clase de aguante mi vida.

A mi hermano Michael, por haberme enseñado las virtudes del colectivo. Nunca hubiera escogido un compañero mejor.

Dedicatoria

*A Uds. que en sus hombros nos han llevado y han sido parte de cada uno
de nuestros logros y sacrificios, a Uds...
Nuestros Padres.*

Resumen

El uso de los marcos de trabajo se ha convertido en una de las principales herramientas utilizadas en la informática para el desarrollo de software en sus diversas variantes, debido al dinamismo y extensibilidad que aportan a la producción. El núcleo de la arquitectura del ¹ERP cubano Cedrux descansa sobre las bases de un marco de trabajo propio y que responde a las necesidades del proyecto, sin embargo, la configuración de los aspectos arquitectónicos que gestionan dicha arquitectura se realiza de forma manual.

El presente trabajo de diploma centra su contenido en la creación de un componente que permita una configuración dinámica y amigable a los desarrolladores de los aspectos anteriormente mencionados, para lo cual se llevó a cabo una investigación sobre los principales frameworks que configuran los marcos de trabajo con herramientas propias, se realizaron análisis de algoritmos y patrones que tienen como base la gestión de archivos ²XML como parte del diseño de la aplicación. Se efectuó un análisis para la selección de la plataforma de desarrollo que satisface los requerimientos impuestos por los componentes, se llevó a cabo la captura de requisitos y pruebas basadas en escenarios definidos por los requisitos funcionales y finalmente se valoró el modelo propuesto mediante el uso de métricas dirigidas a garantizar la calidad del modelo de diseño, obteniendo resultados positivos.

Palabras claves: marco de trabajo, software, aspectos arquitectónicos, entorno de desarrollo.

¹ Enterprise resources planning

² eXtensive markage language

INTRODUCCIÓN	1
CAPÍTULO 1: MARCO TEÓRICO Y ESTUDIO REFERENCIAL DE LA SOLUCIÓN.	5
INTRODUCCIÓN.....	5
ANÁLISIS DE LOS PRINCIPALES PROBLEMAS QUE AFECTAN LA PRODUCCIÓN EN LAS LÍNEAS DE TRABAJO DEL ERP.....	6
ESTUDIO REFERENCIAL	8
1.1 Estado actual de diseño de software.	8
1.2 Inversión de control.....	10
1.2.1 Herramientas que garantizan la inversión de control. Contenedores ligeros. Los contenedores Windsor y MicroKernel.....	11
1.2.2 Inyección de dependencias.	13
1.2.3 Estado actual de la configuración de los componentes basados en la inversión de control en las líneas de desarrollo del ERP.	14
1.3 Aspectos arquitectónicamente significativos.....	15
1.3.1 Estado actual de las herramientas y paradigmas que garantizan la configuración de aspectos arquitectónicos en el mundo, nuestra universidad y el ERP Cedrux.	19
1.4 Vista tecnológica de los componentes del IDE para el marco de trabajo de Cedrux	20
1.4.1 Selección de la plataforma:	20
1.4.2 Selección de las características a evaluar en la selección de la plataforma tecnológica.	23
1.4.3 Calificación y cómputo de totales de cada plataforma.	23
1.4.4 Comparación entre plataformas.	24
1.4.5 Selección de la plataforma apropiada.	26
1.4.6 Selección del framework y las librerías	26
1.6 Arquitectura taxonómica:	32
1.7 Esquema de persistencia.....	33
ELEMENTOS DE REUTILIZACIÓN.....	34
RESULTADOS ESPERADOS.	36
CONCLUSIONES:.....	37
CAPÍTULO 2: DESARROLLO DE LA SOLUCIÓN.....	38
INTRODUCCIÓN.....	38
REQUISITOS.....	38
<i>Levantamiento de requisitos de la solución propuesta.....</i>	<i>39</i>
Requisitos funcionales.....	39
Requisitos no funcionales.....	42
PROTOTIPO DE INTERFAZ.....	44

DISEÑO DE CLASES	52
ARTEFACTOS DE IMPLEMENTACIÓN	53
1. <i>Esquemas de la configuración de los XML</i>	53
2. <i>Diagrama de componentes</i>	56
3. <i>Modelo de Datos</i>	57
CASOS DE PRUEBAS.....	58
<i>Características de los casos de pruebas dirigidos a escenarios de casos de uso de un sistema</i>	59
<i>Casos de prueba realizados a la solución propuesta</i>	60
CONCLUSIONES.....	68
CAPÍTULO 3: EVALUACIÓN DE LA SOLUCIÓN.....	69
INTRODUCCIÓN.....	69
VALORACIÓN DE LA SOLUCIÓN.	69
1.1 <i>Valoraciones aplicadas a la solución propuesta</i>	70
MÉTRICAS QUE EVALÚAN EL DISEÑO, ANÁLISIS DE LOS RESULTADOS.....	71
2.1 <i>Métricas de software</i>	71
2.2 <i>Métricas orientadas a objetos</i>	72
Características de las métricas orientadas a objetos.....	72
2.3 <i>Métricas orientadas a clases</i>	73
2.4 <i>Métricas aplicadas a la solución propuesta</i>	74
2.4.1 Resultados del instrumento de evaluación de la métrica Tamaño de Clases (TC).....	78
2.4.2 Resultado del instrumento de evaluación de la métrica Respuesta para una clase (RPC).....	80
2.4.3 Resultados del instrumento de evaluación de la métrica Árbol de profundidad de herencia (APH).....	83
2.4.4 Resultados del instrumento de evaluación de la métrica Número de Descendientes (NDD).....	84
2.4.5 Resultados del instrumento de evaluación de la métrica Número de Operaciones Redefinidas para una Sub-Clase (NOR).....	87
MATRIZ DE CUBRIMIENTO O MATRIZ DE INFERENCIA DE INDICADORES DE CALIDAD.....	89
CONCLUSIONES.....	92
CONCLUSIONES GENERALES	93
RECOMENDACIONES.....	94
BIBLIOGRAFÍA.....	95
ANEXOS	99

ANEXO 1: PLANTILLA DE ENCUESTA REALIZADA A LOS DISTINTOS ROLES QUE INTEGRAN EL PROYECTO CEDRUX.99

ANEXO 2: REQUISITOS FUNCIONALES DEL COMPONENTE GENERADOR DE EXCEPCIONES.....101

ANEXO 3: REQUISITOS FUNCIONALES DEL COMPONENTE GENERADOR DE ASPECTOS.112

ANEXO 4: REQUISITOS FUNCIONALES DEL COMPONENTE GENERADOR DE INVERSIÓN DE CONTROL.....124

ANEXO 5: REQUISITOS FUNCIONALES DEL COMPONENTE AUDITOR DE CÓDIGO FUENTE.....130

ANEXO 6: PROTOTIPOS DE INTERFAZ QUE AMERITAN A LOS REQUISITOS FUNCIONALES DEL COMPONENTE AUDITOR DE CÓDIGO FUENTE.
.....133

ANEXO 7: PROTOTIPOS DE INTERFAZ QUE AMERITAN A LOS REQUISITOS FUNCIONALES DEL COMPONENTE GENERADOR DE INVERSIÓN DE CONTROL.136

ANEXO 8: DISEÑO DE CLASES DEL COMPONENTE GENERADOR DE ASPECTOS.138

ANEXO 9: DISEÑO DE CLASES DEL COMPONENTE GENERADOR DE EXCEPCIONES.....139

ANEXO 10: DISEÑO DE CLASES DEL COMPONENTE AUDITOR DE CÓDIGO FUENTE.141

ANEXO 11: DISEÑO DE CLASES DEL COMPONENTE GENERADOR DE INVERSIÓN DE CONTROL.143

ANEXO 12: DISEÑO DE CASOS DE PRUEBAS DEL COMPONENTE GENERADOR DE ASPECTOS.144

ANEXO 13: DISEÑO DE CASOS DE PRUEBAS DEL COMPONENTE GENERADOR DE EXCEPCIONES.....174

ANEXO 14: DISEÑO DE CASOS DE PRUEBAS DEL GENERADOR DE INVERSIÓN DE CONTROL.211

ANEXO 15: DISEÑO DE CASOS DE PRUEBAS DEL COMPONENTE AUDITOR DE CÓDIGO FUENTE.225

ANEXO 16: INSTRUMENTO DE MEDICIÓN DE LA MÉTRICA DE TAMAÑO DE CLASES (TC).231

ANEXO 17: INSTRUMENTO DE MEDICIÓN DE LA MÉTRICA RESPUESTA PARA UNA CLASE (RPC).235

ANEXO 18: INSTRUMENTO DE MEDICIÓN DE LA MÉTRICA ÁRBOL DE PROFUNDIDAD DE HERENCIA (APH).237

ANEXO 19: INSTRUMENTO DE MEDICIÓN DE LA MÉTRICA NÚMERO DE DESCENDIENTES (NDD).....238

ANEXO 20: INSTRUMENTO DE MEDICIÓN DE LA MÉTRICA NÚMERO DE OPERACIONES REDEFINIDAS PARA UNA SUB-CLASE (NOR) ..240

TABLA 1: PERSONAL ENCUESTADO PARA CONOCER LOS CRITERIOS SOBRE LA GESTIÓN MANUAL DE LOS ARCHIVOS DE CONFIGURACIÓN.	6
TABLA 2: RESULTADOS DE LA ENCUESTA PARA CONOCER LOS CRITERIOS SOBRE LA GESTIÓN MANUAL DE LOS ARCHIVOS DE CONFIGURACIÓN.	7
TABLA 3: PARÁMETROS DE IMPORTANCIA PARA LA INVESTIGACIÓN EN LA COMPARACIÓN ENTRE PLATAFORMAS.	23
TABLA 4: PRINCIPALES ELEMENTOS DE COMPARACIÓN ENTRE LAS PLATAFORMAS.	25
TABLA 5: ELEMENTOS DE REUTILIZACIÓN.	35
TABLA 6: ESTRUCTURA DE LOS REQUISITOS FUNCIONALES DEL COMPONENTE GENERADOR DE EXCEPCIONES.	39
TABLA 7: ESTRUCTURA DE LOS REQUISITOS FUNCIONALES DEL COMPONENTE GENERADOR DE ASPECTOS.	40
TABLA 8: ESTRUCTURA DE LOS REQUISITOS FUNCIONALES DEL COMPONENTE GENERADOR DE INVERSIÓN DE CONTROL.	41
TABLA 9: ESTRUCTURA DE LOS REQUISITOS FUNCIONALES DEL COMPONENTE AUDITOR DE CÓDIGO FUENTE.	42
TABLA 10: PAQUETE DE DCP DEL ESCENARIO GESTIONAR FICHEROS PERTENECIENTE AL COMPONENTE GENERADOR DE ASPECTOS.	60
TABLA 11: TAMAÑO DE CLASES (TC).	76
TABLA 12: RESPUESTA PARA UNA CLASE (RPC).	76
TABLA 13: ÁRBOL DE PROFUNDIDAD DE HERENCIA (APH).	76
TABLA 14: NÚMERO DE DESCENDIENTES (NDD).	77
TABLA 15: NÚMERO DE OPERACIONES REDEFINIDAS PARA UNA SUB-CLASE (NOR).	77
TABLA 16: RESULTADOS EVALUADOS DE LA RELACIÓN ATRIBUTOS/MÉTRICAS POR CADA COMPONENTE QUE INTEGRAN LA SOLUCIÓN.	90
TABLA 17: RANGO DE VALORES PARA LA EVALUACIÓN TÉCNICA DE LOS ATRIBUTOS DE CALIDAD EVALUADOS POR CADA MÉTRICA.	90
TABLA 18: PAQUETE DE R.F. DE LA GESTIÓN DE FICHEROS DEL COMPONENTE GENERADOR DE EXCEPCIONES.	101
TABLA 19: CRUD. DEL R1.1.1.1 GESTIONAR EXCEPCIONES.	103
TABLA 20: CRUD. DEL R1.1.1.2 GESTIONAR LENGUAJES.	105
TABLA 21: CRUD. DEL R1.1.2.1 GESTIONAR MÓDULOS.	107
TABLA 22: CRUD. DEL R1.1.2.2 GESTIONAR CONTROLADOR.	108
TABLA 23: CRUD. DEL R1.1.2.3 GESTIONAR ACCIÓN.	110
TABLA 24: PAQUETE DE R.F. DE LA GESTIÓN DE FICHEROS DEL COMPONENTE GENERADOR DE ASPECTOS.	112
TABLA 25: CRUD. DEL R2.1.1 GESTIONAR ASPECTOS ARQUITECTÓNICOS (ASPECT.XML).	114
TABLA 26: CRUD. DEL R2.1.2 GESTIONAR PLANTILLA DE ASPECTOS (ASPECTTEMPLATE.XML).	116
TABLA 27: CRUD. DEL R2.1.3.1 GESTIONAR MÓDULO PERTENECIENTE AL R2.1.3 GESTIONAR TEJEDOR DE ASPECTOS (WEAVER.XML).	118
TABLA 28: CRUD. DEL R2.1.3.2 GESTIONAR CONTROLADOR.	120
TABLA 29: CRUD. DEL R2.1.3.3 GESTIONAR ACCIÓN.	122
TABLA 30: PAQUETE DE R.F. DE LA GESTIÓN DE FICHEROS DEL COMPONENTE GENERADOR DE INVERSIÓN DE CONTROL.	124
TABLA 31: CRUD. DEL R3.2 GESTIONAR SUBSISTEMA.	126
TABLA 32: CRUD. DEL R3.3 GESTIONAR CONTRATO.	127
TABLA 33: CRUD. DEL R3.3.1.1 GESTIONAR PROTOTIPO.	128
TABLA 34: PAQUETE DE R.F. DE LA GESTIÓN DE FICHEROS DEL COMPONENTE AUDITOR DE CÓDIGO FUENTE.	130

TABLA 35: FUNCIONALIDADES DEL R4.2 GESTIONAR FICHEROS.	132
TABLA 36: DCP DE CRUD. DEL ESCENARIO GESTIONAR MÓDULO PERTENECIENTE AL COMPONENTE GENERADOR DE ASPECTOS.....	144
TABLA 37: DCP DE CRUD DEL ESCENARIO GESTIONAR CONTROLADOR DEL COMPONENTE GENERADOR DE ASPECTOS.....	150
TABLA 38: DCP DE CRUD DEL ESCENARIO GESTIONAR ACCIÓN DEL COMPONENTE GENERADOR DE ASPECTOS.....	156
TABLA 39: DCP DE CRUD DEL ESCENARIO GESTIONAR PLANTILLA DE ASPECTOS DEL COMPONENTE GENERADOR DE ASPECTOS.....	161
TABLA 40: DCP DE CRUD DEL ESCENARIO GESTIONAR ASPECTO DEL COMPONENTE GENERADOR DE ASPECTOS.	167
TABLA 41: PAQUETE DE DCP DEL ESCENARIO GESTIONAR FICHEROS PERTENECIENTE AL COMPONENTE GENERADOR DE EXCEPCIONES....	174
TABLA 42: DCP DE CRUD DEL ESCENARIO GESTIONAR MÓDULO DEL COMPONENTE GENERADOR DE EXCEPCIONES.	181
TABLA 43: DCP DE CRUD DEL ESCENARIO GESTIONAR CONTROLADOR DEL COMPONENTE GENERADOR DE EXCEPCIONES.	187
TABLA 44: DCP DE CRUD DEL ESCENARIO GESTIONAR ACCIÓN DEL COMPONENTE GENERADOR DE EXCEPCIONES.	193
TABLA 45: DCP DE CRUD DEL ESCENARIO GESTIONAR EXCEPCIÓN DEL COMPONENTE GENERADOR DE EXCEPCIONES.....	198
TABLA 46: DCP DE CRUD DEL ESCENARIO GESTIONAR LENGUAJE DEL COMPONENTE GENERADOR DE EXCEPCIONES.....	205
TABLA 47: PAQUETE DE DCP DEL ESCENARIO GESTIONAR FICHEROS PERTENECIENTE AL COMPONENTE GENERADOR DE INVERSIÓN DE CONTROL.....	211
TABLA 48: DCP DE CRUD DEL ESCENARIO GESTIONAR SUBSISTEMA DEL COMPONENTE GENERADOR DE INVERSIÓN DE CONTROL.....	216
TABLA 49: DCP DE CRUD DEL ESCENARIO GESTIONAR CONTRATO DEL COMPONENTE GENERADOR DE INVERSIÓN DE CONTROL.....	220
TABLA 50: PAQUETE DE DCP DEL ESCENARIO GESTIONAR FICHEROS PERTENECIENTE AL COMPONENTE AUDITOR DE CÓDIGO FUENTE....	225
TABLA 51: RANGO DE VALORES PARA LA EVALUACIÓN TÉCNICA DE LOS ATRIBUTOS DE CALIDAD (RESPONSABILIDAD, COMPLEJIDAD DE IMPLEMENTACIÓN Y REUTILIZACIÓN) RELACIONADOS CON LA MÉTRICA TC.....	231
TABLA 52: RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TC Y SU INFLUENCIA EN LOS ATRIBUTOS DE CALIDAD (RESPONSABILIDAD, COMPLEJIDAD DE IMPLEMENTACIÓN Y REUTILIZACIÓN).	231
TABLA 53: RANGO DE VALORES DE PARA LA EVALUACIÓN TÉCNICA DE LOS ATRIBUTOS DE CALIDAD (ACOPLAMIENTO, COMPLEJIDAD DE MANTENIMIENTO, REUTILIZACIÓN Y CANTIDAD DE PRUEBAS) RELACIONADOS CON LA MÉTRICA RPC.....	235
TABLA 54: RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RPC Y SU INFLUENCIA EN LOS ATRIBUTOS DE CALIDAD (ACOPLAMIENTO, COMPLEJIDAD DE MANTENIMIENTO, REUTILIZACIÓN Y CANTIDAD DE PRUEBAS).....	235
TABLA 55: RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA APH.	237
TABLA 56: RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA NDD Y SU INFLUENCIA EN LOS ATRIBUTOS DE CALIDAD (REUTILIZACIÓN, ABSTRACCIÓN DEL DISEÑO, NIVEL DE COHESIÓN Y CANTIDAD DE PRUEBAS).....	239
TABLA 57: RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA NOR Y SU INFLUENCIA EN LOS ATRIBUTOS DE CALIDAD (ABSTRACCIÓN DEL DISEÑO, CANTIDAD DE PRUEBAS, COMPLEJIDAD DE MANTENIMIENTO).....	240

FIGURA 1: ARQUITECTURA BASE DEL FRAMEWORK .NET.....	27
FIGURA 2: COMPILADOR DE LA PLATAFORMA .NET.....	28
FIGURA 3: BIBLIOTECA DE CLASES .NET.	30
FIGURA 4: ESTRUCTURA DEL COMPILADOR DE LA PLATAFORMA .NET.....	32
FIGURA 5: INTERFAZ QUE AMERITA AL PAQUETE DE R. F. DE LA GESTIÓN DE FICHEROS DEL COMPONENTE GENERADOR DE EXCEPCIONES. .45	
FIGURA 6: INTERFAZ QUE AMERITA AL CRUD R1.1.2.1 GESTIONAR MÓDULO.	45
FIGURA 7: INTERFAZ QUE AMERITA AL CRUD. DEL R1.1.2.2 GESTIONAR CONTROLADOR.....	46
FIGURA 8: INTERFAZ QUE AMERITA AL CRUD. DEL R1.1.2.3 GESTIONAR ACCIÓN.....	46
FIGURA 9: INTERFAZ QUE AMERITA AL CRUD. DEL R1.1.1.1 GESTIONAR EXCEPCIÓN.	47
FIGURA 10: INTERFAZ QUE AMERITA AL CRUD. DEL R1.1.1.2 GESTIONAR LENGUAJES.	47
FIGURA 11: INTERFAZ QUE AMERITA AL PAQUETE DE R.F. DE LA GESTIÓN DE FICHEROS DEL COMPONENTE GENERADOR DE ASPECTOS.	48
FIGURA 12: INTERFAZ QUE AMERITA AL CRUD. DEL R2.1.3.1 GESTIONAR MÓDULOS.	49
FIGURA 13: INTERFAZ QUE AMERITA AL CRUD. DEL R2.1.3.2 GESTIONAR CONTROLADORES.....	49
FIGURA 14: INTERFAZ QUE AMERITA AL CRUD. DEL R2.1.3.3 GESTIONAR ACCIONES.....	50
FIGURA 15: INTERFAZ QUE AMERITA AL CRUD. DEL R2.1.2. GESTIONAR PLANTILLA DE ASPECTOS.....	50
FIGURA 16: INTERFAZ QUE AMERITA AL CRUD. DEL R2.1.1 GESTIONAR ASPECTOS.....	51
FIGURA 17: DISEÑO DE CLASES DE LA INTEGRACIÓN DEL COMPONENTE GENERAL.	53
FIGURA 18: ESQUEMA DEL XML DE ASPECTOS (ASPET.XML).....	54
FIGURA 19: ESQUEMA DEL XML DE LA PLANTILLA DE ASPECTOS (ASPECTTEMPLATE.XML).....	54
FIGURA 20: ESQUEMA DEL XML TEJEDOR (WEAVER.XML).....	54
FIGURA 21: ESQUEMA DEL XML DE EXCEPCIÓN (EXCEPTION.XML).....	55
FIGURA 22: ESQUEMA DEL XML MANEJADOR DE EXCEPCIONES (MANAGEREXCEPTION.XML).....	55
FIGURA 23: ESQUEMA DEL XML DE INVERSIÓN DE CONTROL (IOC.XML).....	55
FIGURA 24: DIAGRAMA DE COMPONENTES DEL SISTEMA (PARTE I).....	56
FIGURA 25: DIAGRAMA DE COMPONENTES DEL SISTEMA (PARTE II).....	57
FIGURA 26: MODELO DE DATOS DEL COMPONENTE AUDITOR DE CÓDIGO FUENTE.	58
FIGURA 27: REPRESENTACIÓN DE LOS RESULTADOS OBTENIDOS EN EL INSTRUMENTO AGRUPADOS EN LOS INTERVALOS DEFINIDOS.....	78
FIGURA 28: REPRESENTACIÓN EN % DE LOS RESULTADOS OBTENIDOS EN EL INSTRUMENTO AGRUPADOS EN LOS INTERVALOS DEFINIDOS..	78
FIGURA 29: REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TC EN EL ATRIBUTO RESPONSABILIDAD.....	79
FIGURA 30: REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TC EN EL ATRIBUTO COMPLEJIDAD DE IMPLEMENTACIÓN.	79
FIGURA 31: REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TC EN EL ATRIBUTO REUTILIZACIÓN.....	79
FIGURA 32: GRÁFICA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RPC AGRUPADOS POR LA TENDENCIA DE LOS VALORES. .	80

FIGURA 33: REPRESENTACIÓN EN % DE LOS RESULTADOS OBTENIDOS EN EL INSTRUMENTO AGRUPADOS EN LOS INTERVALOS DEFINIDOS..	81
FIGURA 34: REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RPC EN EL ATRIBUTO ACOPLAMIENTO.....	81
FIGURA 35: REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RPC EN EL ATRIBUTO COMPLEJIDAD DE MANTENIMIENTO.	82
FIGURA 36: REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RPC EN EL ATRIBUTO CANTIDAD DE PRUEBAS.....	82
FIGURA 37: REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RPC EN EL ATRIBUTO REUTILIZACIÓN.	83
FIGURA 38: REPRESENTACIÓN EN % DE LOS RESULTADOS OBTENIDOS EN EL INSTRUMENTO AGRUPADOS POR NIVEL.	84
FIGURA 39: REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA NND EN EL ATRIBUTO REUTILIZACIÓN.	85
FIGURA 40: REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA NDD EN EL ATRIBUTO ABSTRACCIÓN DE LA CLASE BASE.	85
FIGURA 41: REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA NDD EN EL ATRIBUTO COHESIÓN DE LA JERARQUÍA DE CLASES.	86
FIGURA 42: REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA NDD EN EL ATRIBUTO CANTIDAD DE PRUEBAS.....	86
FIGURA 43: REPRESENTACIÓN EN % DE LOS RESULTADOS OBTENIDOS EN EL INSTRUMENTO AGRUPADOS EN LOS VALORES EXISTENTES.	87
FIGURA 44: REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA NOR EN EL ATRIBUTO COMPLEJIDAD DEL MANTENIMIENTO.	88
FIGURA 45: REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA NOR EN EL ATRIBUTO CANTIDAD DE PRUEBAS.....	88
FIGURA 46: REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA NOR EN EL ATRIBUTO VIOLACIÓN DE LA ABSTRACCIÓN REPRESENTADA POR LA SUPERCLASE.	89
FIGURA 47: GRÁFICA DE LOS RESULTADOS OBTENIDOS DE LOS ATRIBUTOS DE CALIDAD EVALUADOS EN LAS MÉTRICAS.	91
FIGURA 48: INTERFAZ QUE AMERITA AL PAQUETE R4.1 GESTIONAR PROYECTOS.	134
FIGURA 49: INTERFAZ QUE AMERITA AL R4.2 GESTIONAR FICHEROS.	134
FIGURA 50: INTERFAZ QUE AMERITA AL R4.2.1 INSTANCIAR.	135
FIGURA 51: INTERFAZ QUE AMERITA A LOS R4.2.2 AUDITAR Y R4.2.3 CHEQUEAR.....	135
FIGURA 52: INTERFAZ QUE AMERITA AL PAQUETE DE R. F. GESTIONAR FICHERO.....	136
FIGURA 53: INTERFAZ QUE AMERITA AL CRUD DEL R3.2 GESTIONAR SUBSISTEMA.	136
FIGURA 54: INTERFAZ QUE AMERITA AL CRUD R3.3 GESTIONAR CONTRATO.	137
FIGURA 55: INTERFAZ QUE AMERITA AL CRUD 3.3.1.1 GESTIONAR PROTOTIPO.....	137

FIGURA 56: DISEÑO DE CLASES DEL PAQUETE DE GESTORES QUE PERTENECE AL PAQUETE DE LÓGICA DEL COMPONENTE GENERADOR DE ASPECTOS.	138
FIGURA 57: DISEÑO DE CLASES DEL PAQUETE DE ENTIDADES QUE PERTENECE AL PAQUETE DE LÓGICA DEL COMPONENTE GENERADOR DE ASPECTOS.	139
FIGURA 58: CONFIGURACIÓN DE LOS PAQUETES DE CLASES DEL COMPONENTE GENERADOR DE ASPECTOS.	139
FIGURA 59: DISEÑO DE CLASES DEL PAQUETE DE GESTORES QUE PERTENECE AL PAQUETE DE LÓGICA DEL COMPONENTE GENERADOR DE EXCEPCIONES.	140
FIGURA 60: DISEÑO DE CLASES DEL PAQUETE DE ENTIDADES QUE PERTENECE AL PAQUETE DE LÓGICA DEL COMPONENTE GENERADOR DE EXCEPCIONES.	140
FIGURA 61: CONFIGURACIÓN DE LOS PAQUETES DE CLASES DEL COMPONENTE GENERADOR DE EXCEPCIONES.	140
FIGURA 62: DISEÑO DE CLASES DEL PAQUETE DE ENTIDADES QUE PERTENECE AL PAQUETE DE LÓGICA DEL COMPONENTE AUDITOR DE CÓDIGO FUENTE.	141
FIGURA 63: DISEÑO DE CLASES DE LOS PAQUETES DE GESTORES QUE PERTENECE AL PAQUETE DE LÓGICA, CONSULTAS Y CRUD QUE PERTENECEN AL DE CAD (CAPA DE ACCESO A DATOS) DEL COMPONENTE AUDITOR DE CÓDIGO FUENTE.	142
FIGURA 64: DISEÑO DE CLASES DEL PAQUETE ÚTILES QUE PERTENECE AL COMPONENTE AUDITOR DE CÓDIGO FUENTE.	142
FIGURA 65: CONFIGURACIÓN DE LOS PAQUETES DE CLASES DEL COMPONENTE AUDITOR DE CÓDIGO FUENTE.	143
FIGURA 66: DISEÑO DE CLASES DEL PAQUETE PERSISTIDOR Y EL PAQUETE CLASES QUE PERTENECE AL COMPONENTE GENERADOR DE INVERSIÓN DE CONTROL.	143
FIGURA 67: CONFIGURACIÓN DE LOS PAQUETES DE CLASES DEL COMPONENTE GENERADOR DE INVERSIÓN DE CONTROL.	144
FIGURA 68: GRÁFICA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TC Y SU INFLUENCIA EN LOS ATRIBUTOS DE CALIDAD (RESPONSABILIDAD, COMPLEJIDAD DE IMPLEMENTACIÓN Y REUTILIZACIÓN), PARTE 1.	233
FIGURA 69: GRÁFICA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TC Y SU INFLUENCIA EN LOS ATRIBUTOS DE CALIDAD (RESPONSABILIDAD, COMPLEJIDAD DE IMPLEMENTACIÓN Y REUTILIZACIÓN), PARTE 2.	234
FIGURA 70: GRÁFICA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TC Y SU INFLUENCIA EN LOS ATRIBUTOS DE CALIDAD (RESPONSABILIDAD, COMPLEJIDAD DE IMPLEMENTACIÓN Y REUTILIZACIÓN), PARTE 3.	234
FIGURA 71: GRÁFICA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA APH.	238
FIGURA 72: RANGO DE VALORES PARA LA EVALUACIÓN TÉCNICA DE LOS ATRIBUTOS DE CALIDAD (REUTILIZACIÓN, ABSTRACCIÓN DEL DISEÑO, NIVEL DE COHESIÓN Y CANTIDAD DE PRUEBAS) RELACIONADOS CON LA MÉTRICA NDD.	239
FIGURA 73: REPRESENTACIÓN DE LOS RESULTADOS OBTENIDOS AL APLICAR LOS INSTRUMENTOS QUE EVALÚAN LA MÉTRICA NDD.	240
FIGURA 74: RANGO DE VALORES PARA LA EVALUACIÓN TÉCNICA DE LOS ATRIBUTOS DE CALIDAD (ABSTRACCIÓN DEL DISEÑO, CANTIDAD DE PRUEBAS, COMPLEJIDAD DE MANTENIMIENTO) RELACIONADOS CON LA MÉTRICA NOR.	240
FIGURA 75: REPRESENTACIÓN DE LOS RESULTADOS OBTENIDOS AL APLICAR LOS INSTRUMENTOS QUE EVALÚAN LA MÉTRICA NOR.	241

Introducción

Los productos informáticos evolucionan dinámicamente para poder adaptarse a las necesidades tecnológicas del momento. Las aplicaciones dirigidas a la plataforma web no están exentas de estos requerimientos, por lo que reflejan las contradicciones y cambios que ocurren en la sociedad. Las características y potencialidades de las aplicaciones o productos web; el gran poder de difusión y alcance que posee, así como la heterogeneidad de usuarios que hacen uso de ellas convierte a este tipo de software en herramientas de alta capacidad, y uso ilimitado en los más diversos campos; resaltando entre ellos la planificación económica, el comercio electrónico, la gestión de datos y otras importantes ramas de la ciencia y la técnica.

La mayor parte de las aplicaciones web en los países capitalistas están orientadas a ser parte del sistema consumista que impera en estas naciones, las mismas son fundamentalmente utilizadas para el marketing y venta de sus productos mediante el desarrollo web, provocando consigo un aumento de la publicidad, abarrotamiento de los mercados internacionales y subida del puntaje empresarial en internet. Sin embargo en otros países menos desarrollados se utiliza con propósitos más humanitarios para la sociedad, en vista de una mejora económica de la misma y una mejor resolución de los problemas inherentes al tercer mundo, como la pobreza entre otros.

Entre los lenguajes de programación web más utilizados en el mundo debido a las características que poseen y potencian están: ¹PHP, ²ASP y ASP.NET.

En Cuba el área de la programación web se especializa en plataformas ³LAMP y ⁴LAPP debido a que esta tecnología brinda un bajo costo de soporte y mantenimiento y cumple con el principio de plataforma libre. Nuestro país se inclina por el uso del lenguaje PHP, programación del lado del servidor que posee código abierto y puede ser modificado por los desarrolladores a su conveniencia, introduciendo nuevas funciones para crear versiones mejoradas del mismo. Siendo un caso muy distinto en otros lenguajes como ASP y la plataforma .NET, que realizan la

¹ *PHP Hypertext Pre-processor*

² *Active Server Pages*

³ *Linux/Apache/PHP/MySQL*

⁴ *Linux/Apache/PHP/PostgreSQL*

totalidad de las tareas o al menos algunas de ellas apoyándose en productos que deben ser comprados a empresas especializadas.

Como parte de las soluciones informáticas orientadas a la plataforma web que se elaboran actualmente en nuestro país y debido a la necesidad de incrementar la eficiencia económica en cuanto a la planificación y el control de los recursos empresariales, nace el nuevo proyecto ¹ERP Cedrux.

Los ERP pueden ser vistos como soluciones para tecnologías de la información que permiten integrar los procesos de competencias de las empresas. Estos productos son modulares en estructura y comprenden diversas áreas en su logística, como planificación financiera, ventas, procesamiento de órdenes, producción y planificación de los recursos humanos y materiales; las organizaciones pueden implementar uno o varios módulos al mismo tiempo, y realizar seguimiento para la implementación de otros módulos proyectados a futuro. (Deeplnder, 2004)

El interés por utilizar los sistemas ERP está determinado por el éxito y la planificación de los negocios, teniendo numerosos beneficios en su implementación ya sea de forma operacional o estratégica. La evidencia de estudios con los que se cuenta en la actualidad indica que contar con un sistema ERP mejora la información en las entidades y la integración de operaciones en general. Las organizaciones tienden a tener un alto desarrollo financiero con respecto al valor del mercado que aquellas que no han invertido en este tipo de sistemas. (Shang, 2000)

“Cedrux”, es una novedosa solución informática, un complejo y extenso sistema de software, que se construye especialmente adaptado al control económico en las condiciones de nuestro país, de ahí su importancia, ya que en el mundo existen otras soluciones ERP, como son Open Bravo, TinyERP y otras, pero ya sea por ser software de tipo privativo o por las características de nuestra economía no son viables de aplicar.

El correcto funcionamiento de un ERP depende en gran medida del núcleo de procesamiento que este contenga es decir la arquitectura y diseño arquitectónico con que cuente el mismo. La arquitectura de software engloba las diferentes formas de presentar el sistema mediante componentes con puntos de vistas o perspectivas distintas. Sin embargo no solo está influenciada por la estructura y el comportamiento, sino también por el uso, la funcionalidad, el rendimiento, la flexibilidad, la reutilización, la facilidad de comprensión, las restricciones y

¹ Enterprise Resource Planning

compromisos económicos, tecnológicos y la estética. (Jacobson, Booch, & Rumbaugh, 2000) Para lograr la construcción de un sistema con estas características y de esta complejidad, es necesario crear una arquitectura de software eficiente y capaz, moldeada a las necesidades del diseño de la aplicación.

Es la estructura jerárquica de los componentes del programa (módulos), la manera de interactuar de estos componentes y la estructura de los datos usados por los mismos. (Pressman, 2000). Aunque en un sentido más amplio estos componentes pueden generalizarse para representar los componentes del sistema y sus interacciones.

Por la importancia y el alcance que tienen la rama de la arquitectura y la configuración de los aspectos arquitectónicos (caché, excepciones, transacciones y otros), dentro de la construcción del ERP cubano, se realizó un análisis detallado de las actividades llevadas a cabo en cada una de las líneas de desarrollo y módulos del software, el estudio de los resultados reveló un grupo de deficiencias en la gestión de los archivos que configuran la arquitectura del marco de trabajo, lo cual permitió definir como **situación problemática** de esta investigación: *La producción en las líneas y módulos donde se construye actualmente el software "Cedrux" se ve afectada por deficiencias debidas a la gestión manual de los archivos de configuración de los aspectos arquitectónicos e integración de componentes, la cual trae como consecuencias baja eficiencia en la utilización del tiempo y los recursos humanos, así como la introducción de errores.*

Dentro de esta situación, el presente Trabajo de Diploma se plantea como **problema a resolver**.

La necesidad de eliminar la configuración manual de los aspectos arquitectónicos e integración de componentes en las líneas de desarrollo y módulos de Cedrux.

El mismo tiene como **objeto de estudio** *los frameworks de inversión de control y de aspectos* y su **campo de acción** está determinado por *los componentes de inversión de control y aspectos de Cedrux.*

Proponiendo como **objetivo general de la investigación**:

Desarrollar un componente para la configuración de los aspectos arquitectónicos e integración de componentes en el marco de trabajo de Cedrux.

Con los siguientes **objetivos específicos**:

- Investigar el estado actual de los frameworks y componentes que gestionan la integración basada en inversión de control y configuración de los aspectos arquitectónicos en el mundo, nuestro país y la universidad.
- *Diseñar* un componente para la integración, y la configuración del marco de trabajo de la arquitectura de software del proyecto ERP Cuba.
- Implementar el componente para la integración y la configuración del marco de trabajo de la arquitectura de software del proyecto ERP Cuba.
- Realizar pruebas al componente para la integración y la configuración del marco de trabajo de la arquitectura de software del proyecto ERP Cuba.
- Validar el diseño mediante la utilización de instrumentos de medición.

Por tanto, luego de llevar a cabo la construcción de esta herramienta y su utilización en las diferentes líneas de desarrollo del ERP se puede enmarcar como ***Idea a Defender***:

La existencia de un componente para la configuración de los aspectos arquitectónicos y la integración de componentes minimiza los errores y disminuye el tiempo de desarrollo de Cedrux.

Este Trabajo de Diploma consta de tres capítulos, en el primero se aborda la fundamentación teórica y marco contextual de la investigación, el estado referencial, estado de reutilización y resultados esperados u objetivos de la solución propuesta. El segundo capítulo trata sobre los requisitos, prototipos de interfaces, diseño de clases, artefacto de implementación y casos de pruebas. El tercer capítulo realiza una valoración de la solución, matriz de cubrimiento o matriz de inferencia de indicadores de calidad y las métricas que evalúan el diseño de la solución propuesta.

Capítulo 1: Marco teórico y estudio referencial de la solución.

Introducción

En este capítulo se hace un análisis de los principales problemas relacionados con la configuración manual de la arquitectura del marco de trabajo en las distintas líneas de desarrollo y módulos del ERP cubano CedruX, que afectan el rendimiento de la producción. Se lleva a cabo un estudio de las herramientas que existen a nivel mundial, nacional y en la Universidad de las Ciencias Informáticas, que permiten la configuración de estos aspectos arquitectónicos y la integración mediante inversión de control de los componentes. Se incluye además la vista tecnológica para la creación del marco de configuración de la arquitectura de CedruX.

Análisis de los principales problemas que afectan la producción en las líneas de trabajo del ERP.

El proyecto de software ERP Cedrux se encuentra en pleno desarrollo por lo que carece de una herramienta que garantice la configuración del marco de trabajo del mismo, por tanto, los ficheros de datos encargados de gestionar los aspectos arquitectónicos son tratados de forma manual e insegura.

Como parte de la presente investigación, se aplicó una encuesta, (Anexo1: Plantilla de encuesta realizada a los distintos roles que integran el proyecto Cedrux.), a un grupo de 66 personas del equipo de trabajo del software relacionado (muestra representativa), con el fin de conocer las características, efectos y consecuencias de la gestión manual de los archivos de configuración. (Vargas, 2005)

Tabla 1: Personal encuestado para conocer los criterios sobre la gestión manual de los archivos de configuración.

Personal encuestado	No.	%
Jefe de Línea	3	4.54
Analista	10	15.15
Planificador	2	3.03
Arquitecto	6	9.09
Desarrollador	45	68.18
Total	66	100

Fuente: Elaboración Vargas D y Hernández M 2009

Tabla 2: Resultados de la encuesta para conocer los criterios sobre la gestión manual de los archivos de configuración.

Resultados de la encuesta:		
	No.	%
3. Insatisfechos con la configuración manual	61	92.42
4. Principales dificultades:		
Pérdida de tiempo	56	84.85
Introducción de errores.	48	72.73
Incremento del tiempo de máquina.	30	45.45
Baja productividad del equipo de trabajo.	49	74.24
Disminución de la moral del equipo de trabajo.	44	66.67
5. Consideran necesario desarrollar componentes de configuración	61	92.42

Fuente: Elaboración Vargas D y Hernández M 2009

El análisis de los resultados obtenidos por la encuesta aplicada al personal del ERP, refleja el impacto negativo, efecto de la configuración manual de los archivos de configuración de la arquitectura del marco de trabajo, de donde se puede concluir que esta práctica causa pérdida de tiempo, que se traduce en retrasos innecesarios en el cumplimiento de los hitos y metas programadas para la liberación de componentes y funcionalidades importantes, disminuyendo la eficiencia del trabajo en Cedrux e incrementando la introducción de errores en los ficheros de configuración.

En general se impone como criterio general (92.42%), *la necesidad de desarrollar un componente para la configuración de aspectos arquitectónicos e integración de componentes de la arquitectura de Cedrux.*

Estudio Referencial

1.1 Estado actual de diseño de software.

Actualmente el concepto de diseño es orientado a diferentes aristas como la que plantea Grady Booch” el diseño es el proceso de determinar una implementación efectiva y eficiente que realice las funciones y tenga la información del análisis de dominio” (Booch, 1991), o la planteada por R.S.Pressman, “el diseño del software se encuentra en el núcleo técnico de la ingeniería de software y se aplica independientemente del modelo de diseño que se utilice. Una vez que se analizan y especifican los requisitos, el diseño del software es la primera de las tres actividades técnicas-diseño, generación de código y pruebas que se requieren para construir y verificar la aplicación. Cada actividad transforma la información de manera que dé lugar por último a un software de computadora validado” (PRESSMAN 1998). El diseño es el lugar en donde se fomenta la calidad en la ingeniería de software. El diseño es la única forma de convertir exactamente los requisitos de un cliente en un producto o sistema finalizado. El diseño sirve como fundamento para todos los pasos siguientes del soporte y de la ingeniería del software. Sin el diseño, corremos el riesgo de construir un sistema inestable, un sistema cuya realidad no puede evaluarse hasta muy avanzado el proceso, sin tiempo suficiente y con mucho dinero gastado (PRESSMAN 1998). (Rumbaugh, Jacobson, & Booch, 2002)

En el caso de las metodologías, como por ejemplo RUP, los criterios sobre el diseño de sistemas plantean que es el flujo de trabajo donde el arquitecto y los diseñadores de sistema, describen de una forma más concreta y acorde al tipo de tecnología, cómo serán implementadas las funcionalidades del sistema en construcción, basándose en los tipos de colaboración, la relación que existe entre las clases, y las responsabilidades de cada unidad de código (Rumbaugh, Jacobson, & Booch, 1998).

Otra metodología muy utilizada es ¹XP, la misma describe el diseño con un enfoque práctico y dinámico para el cual el diseño de sistema es una actividad que construye un proceso de diseño evolutivo que se basa en refactorizar un sistema simple en cada iteración. Todo el diseño se centra en la iteración actual y no se hace nada anticipadamente para necesidades futuras.

¹ Extreme Programming

Sin duda alguna uno de los principios de la programación extrema es la simplicidad, de aquí que el diseño no escape a esta filosofía.

El diseño debe ser lo más simple posible. El paradigma ¹KISS se lleva hasta las últimas consecuencias; por ejemplo, se hace énfasis en no añadir funcionalidad al diseño del sistema nunca antes de lo necesario, por las sencillas razones de que probablemente ahora mismo no sea lo más prioritario o porque quizás nunca llegue a ser necesaria (FOWLER 2003).

Algunas empresas, como DEADELUS, plantean que el diseño de sistemas se ocupa de desarrollar las directrices propuestas durante el análisis en términos de aquella configuración que tenga más posibilidades de satisfacer los objetivos planteados tanto desde el punto de vista funcional como del no funcional. El proceso de diseño de un sistema complejo se suele realizar de forma descendente: Diseño de alto nivel(o descomposición del sistema a diseñar en subsistemas menos complejos), diseño e implementación de cada uno de los subsistemas, especificación consistente y completa del subsistema de acuerdo con los objetivos establecidos en el análisis, desarrollo según la especificación, prueba, integración de todos los subsistemas, (Daedalus, 2006) validación del diseño.

Microsoft por su parte plantea en su modelo ²MSF para el desarrollo de sistemas de software, que el diseño es una actividad que debe ser realizada con el fin de establecer el comportamiento dinámico del sistema, es decir, cómo debe reaccionar ante los acontecimientos. Para su desarrollo también se utilizan los lenguajes de modelado. El resultado obtenido en la etapa de diseño facilita enormemente la implementación posterior de los sistemas, pues proporciona la estructura básica del sistema y cómo los diferentes componentes actúan y se relacionan entre ellos. (Microsoft, 2007)

Partiendo del estado del arte de los conceptos que hoy se manejan en el mundo sobre la importancia y la responsabilidad del “Diseño del Sistema”, planteados tanto por personalidades de la teoría del desarrollo de software, como por metodologías de desarrollo de software, o algunas empresas de desarrollo de software y teniendo en cuenta las características de la solución propuesta, se conceptualiza la actividad de diseño como un proceso y un modelado a la vez, de mucha importancia en la vida de un proyecto de software. Debido a que el proceso de diseño es además un conjunto de pasos repetitivos que permiten al diseñador describir

¹ “Keep It Small and Simple” para unos o “Keep It Simple, Stupid” para otros

² Microsoft Solution Framework

todos los aspectos del sistema a construir, partiendo de las especificaciones que llegan del flujo de análisis; es el flujo que describe de una forma más concreta y teniendo en cuenta la tecnología y el lenguaje de programación que se utiliza, la representación estructural del sistema que se desea construir.

A lo largo del diseño se evalúa la calidad del desarrollo del proyecto apoyándonos en un conjunto de revisiones técnicas. El diseño debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acumular todos los requisitos implícitos que desea el cliente. (Estrada & Estévez, 2003). Debe ser una guía que puedan leer y entender los que construyan el código y los que prueban y mantienen el software. El diseño debe proporcionar una completa idea de lo que es el Software, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la implementación. En el diseño se deben concretar las representaciones funcionales de interacción entre los siguientes aspectos, Cliente-Sistema, Sistema-Dispositivos, Sistema- Soportes de almacenamiento, y es una base para garantizar un producto acorde con las necesidades del cliente. (Lazo & Yzquierdo, 2007)

1.2 Inversión de control.

La inversión de control es un principio abstracto que describe el diseño de una arquitectura de software en la cual el flujo de control está invertido, en comparación con la arquitectura tradicional de “Librerías de software”.

Tradicionalmente el programador especifica la secuencia de decisiones y procedimientos que pueden ocurrir durante el ciclo de vida de un programa mediante llamadas a funciones. En su lugar, en la inversión de control se especifican respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que algún tipo de entidad o arquitectura externa lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir. En general la ¹IOC es un estilo de construcción de software donde, componentes reusables y genéricos controlan la ejecución de funciones específicas desde un ámbito exterior a estas. Con la importante connotación de que estas se encuentran diseñadas e implementadas por separado teniendo como resultado mayores posibilidades de mantenimiento y revisión del código. (Fowler, 2005)

¹ Inversion Of Control

Otra forma de definir la inversión de control es la que expone el proyecto de fuente abierta “Castle”: La Inversión de Control es un principio usado por los frameworks como una forma de permitir a los desarrolladores extender de ellos o crear aplicaciones usándolos. La idea es que el framework tiene conocimiento de los objetos del programador y hace invocaciones a los mismos.

Este concepto es el opuesto de usar una ¹API donde el código del programador hace invocaciones al código de la misma. Por tanto el framework invierte el control: no es el código del programador el que está a cargo, en su lugar es el framework quien hace las llamadas a los objetos basado en ciertos estímulos. (Castle, 2009)

1.2.1 Herramientas que garantizan la inversión de control. Contenedores ligeros. Los contenedores Windsor y MicroKernel.

Un contenedor de Inversión de Control usa el principio expuesto anteriormente para la administración de clases, creación y disposición, y para proveer las dependencias y configuraciones requeridas por las mismas. De esta forma las clases no necesitan obtener o configurar las clases de las que ellas dependen. De esta forma se reduce la asociación entre objetos, en fin, las dependencias y como consecuencia de esto la reusabilidad, pruebas y mantenimiento son tareas muy simplificadas, es decir, son herramientas que engloban las funcionalidades del concepto Inversión de Control y permiten configurar las mismas en los marcos de trabajo. Un contenedor es un objeto el cual ensambla los diferentes componentes en el sistema, en orden de producir un servicio productivo y consistente. (Eini, 2006)

MicroKernel es un contenedor de Inversión de Control el cual es diseñado con vistas a lograr extensibilidad. Combina los módulos de las aplicaciones para crecer de forma ortogonal, de esta forma trabajan entre ellos “lado a lado”, sin crear dependencias o interferir en las funcionalidades de otros módulos permitiendo extender las funcionalidades del contenedor. Puede ser usado como un contenedor acoplable en herramientas que soporten la extensión o conexiones (plug-in) como los compiladores. Este contenedor es configurado a través de código y puede ser utilizado por cualquier sistema de configuración que soporte el proyecto en el cual se utiliza. (Castle, 2009)

¹ Application Programming Interface

Contenedores ligeros para la inversión de control e inyección de dependencias

Picocontainer es un contenedor altamente acoplable, orientado a servicios que garantiza la Inversión de control y utiliza el patrón de Inyección de Dependencias. Desde sus inicios posee un código abierto y una licencia ¹BSD lo que hace libre su uso tanto para fines comerciales como para otros proyectos de software libre. Puede ser utilizado como una alternativa a los patrones ²J2EE de Sun para aplicaciones web o soluciones generales. El mismo consta entre sus características principales de un contenedor para código abierto que puede ser leído y modificado y además resuelve la dependencia entre componentes, facilitando la interrelación de componentes y manteniendo un bajo acoplamiento.

El mismo se especializa en el tipo de inyección de dependencia de tipo constructor, mientras que el Spring en el de tipo “setter”. Pero no dispone -con muchísima diferencia- de la funcionalidad y conjunto de librerías de Spring. Es útil para ser empleado en applets y en aplicaciones ³J2ME, pero no para grandes aplicaciones a nivel de empresas. (Figueras, 2006)

Avalon es uno de los contenedores más antiguo, no es una ventaja para los desarrolladores debido a que su diseño es mucho menos flexible y elegante que el que implementa Spring. Su comunidad es realmente pequeña y no se ha encontrado muchas referencias a usos en aplicaciones reales. (Figueras, 2006)

HiveMind resulta ser una de los contenedores más utilizados después de Spring. De la mano de Apache es un completo contenedor de IOC. Sin embargo existen tres áreas donde Spring es superior: conjunto de librerías, apoyo de la comunidad a todos los niveles y elegancia de uso. (Figueras, 2006)

¹ La **licencia BSD** es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

² **Java 2 Enterprise Edition**

³ **Java 2 Micro Edition**

Contenedor de inversión de control del framework Spring: El módulo Core es el más importante de Spring. Es el que provee el contenedor IOC. Este contenedor nos permite aplicar el patrón Inyección de Dependencias en nuestras aplicaciones.

En forma muy resumida, el objetivo del contenedor IOC es encargarse de instanciar los objetos de un sistema, denominados “beans”, y asignarle sus dependencias. Para que el contenedor pueda llevar a cabo esta tarea, debemos, mediante información de configuración, indicarle dónde se encuentran dichos “beans”. (Greca, 2007)

Suponiendo que dentro de un sistema necesitamos representar una computadora. Dicha computadora poseerá un micro, una motherboard, memorias, disco rígidos, etc. Al crear un objeto de tipo computadora tendríamos que asignarle, mediante código, estos componentes. Utilizando el contenedor de Spring, en lugar de hacer todas estas creaciones y asignaciones de dependencias que posee la computadora con sus componentes, le pediremos al contenedor que nos instancie un objeto de tipo computadora y él será el encargado de crear y asignar las dependencias de la computadora con sus componentes.

El principal concepto es la interfaz BeanFactory. Es una sofisticada implementación del patrón “Factory”, nos permite instanciar objetos de forma muy rápida y fácil, liberándonos de la necesidad de especificar las dependencias en la lógica de nuestro programa y, por lo tanto, desacoplar esta dependencia en tiempo de codificación, quedando ligada al tiempo de “inicialización / start-up” o incluso al de “ejecución / run-time” ya que pueden cambiarse las implementaciones en tiempo de ejecución. (Greca, 2007)

1.2.2 Inyección de dependencias.

Generalmente cuando utilizamos un objeto, este no trabaja solo sino que lo hace comunicándose con otros objetos. Estos objetos (dependencias) son pasados como argumentos en constructores o sesteados como propiedades (inyectados) al objeto por el contenedor IOC.

Cabe destacar la importancia de la inyección de dependencias: El código se vuelve más claro y menos acoplado cuando los objetos no crean sus dependencias. (No necesitan saber donde están localizados ni a qué clase pertenecen). (Greca, 2007)

Para el marco de trabajo de Spring, Inversión de Control e Inyección de Dependencias, si bien están relacionados, no son equivalentes. IOC es visto como un concepto más amplio, como un criterio de buen diseño, que nos permite programar contra interfaces, para desacoplarnos de la lógica específica. Por otro lado la Inyección de Dependencias es un patrón de diseño desarrollado bajo el criterio de inversión de control (Greca, 2007). La idea de Inyección de Dependencias es proveer a los objetos de nuestro sistema de otros denominados colaboradores que les permitan llevar a cabo su finalidad. El patrón es parecido al ¹Builder, pero con un objetivo diferente, proveer desacoplamiento entre los objetos de nuestro sistema.

Existen tres tipos de inyección de dependencia que son:

- Inyección de interfaz.
- Inyección de setter.
- Inyección de constructor.

1.2.3 Estado actual de la configuración de los componentes basados en la inversión de control en las líneas de desarrollo del ERP.

Unos de los problemas existentes para la realización del ERP, estaba dado por la falta de operaciones o interoperabilidad entre los componentes. A raíz de ello se comenzó a estudiar y buscar patrones desde la base conceptual del desarrollo basado en componentes. Entre las posibles soluciones encontradas para satisfacer los problemas existentes, se llevó a cabo un estudio de los Patrones de Integración, realizando un mayor énfasis dentro de estos en el Patrón de IoC y ServiceLocator. Permitiendo definir el trabajo para dar solución a las polémicas, mediante la utilización de Inyección de Dependencia, el cual es un Patrón de Diseño elaborado bajo el concepto de Inversión de Control.

Funcionalidad del IoC: Se formaliza en el IoC en el fichero ioc.xml, el contrato (servicios) que brinda un determinado componente (o varios), permitiendo esto la comprensión de los servicios que pueda brindar el componente a realizarle la petición o encargado de responderla. La descripción del XML tiene un inyector: Que no es más que la función que permite que la clase interface que responde ha dicho servicio y describe el método que se va a ejecutar con sus parámetros y resultados.

¹ Es un patrón de diseño de software. La intención es abstraer los pasos de construcción de los objetos de manera que las distintas implementaciones de estos pasos puedan instanciar representaciones diferentes de los objetos.

Dentro de cada componente a realizar la petición existe una carpeta donde se guardan las interfaces (que vienen definidas y tienen dentro operaciones y funciones que se van a ofrecer en cada servicio) (Equipo Arquitectura del ERP Cedrux, 2008).

¿Cómo interactúan los componentes?

Cuando un componente desea obtener información referenciada en otro componente, este invoca un contrato descrito en el XML (la clase IOC que es el contenedor del objeto), el mismo invoca la operación de la clase interfaz, recibe un resultado (objetos, atributos), y lo devuelve (inyecta) hacia el componente que solicitó la independencia.

En caso de que el componente desee obtener una simulación de los valores referenciados de otro componente, entonces el IoC simula el resultado cumpliendo según la descripción guardada en el fichero XML sin conectarse al otro componente.

1.3 Aspectos arquitectónicamente significativos.

Los métodos actuales del diseño arquitectónico coinciden en la importancia de tomar en cuenta los aspectos de calidad para dirigir la selección de la solución arquitectónica, sin embargo aún no hay acuerdo sobre métodos precisos que puedan ser utilizados en la práctica común. Distintas propuestas de métodos de diseño arquitectónico se han presentado hasta el momento (Jacobson, Booch, & Rumbaugh, 2000), los mismos son fundamentados en conceptos que bien son equivalentes, complementarios o alternativos. Por tanto debido a la carencia de herramientas, métodos, o acuerdos que cumplan con estas características se hace necesario crear herramientas capaces de gestionar los aspectos que van a dar solución a la arquitectura tipo de Cedrux. A continuación se hace un análisis de los principales aspectos arquitectónicamente significativos con los que trata directamente el presente trabajo investigativo. (Buschmann, 2006)

Caché:

La caché no es más que un conjunto de información que generalmente se almacena en ficheros, tanto en clientes como en servidores web en un formato determinado, de forma tal que pueda ser recuperada rápida y eficientemente, permitiendo almacenar información de cualquier tipo, textos, clases, funciones, objetos o instancias de clases, estructuras complejas como

listas, pilas, arboles, arreglos o tablas hash, páginas, ficheros, etc., por lo general esta información no debe cambiar en un período de tiempo determinado, en caso de que alguna información almacenada en la caché cambie o expire esta debe ser actualizada instantáneamente, para evitar problemas de inconsistencia. La mayoría de los gestores de caché almacenan esta información en forma serializada para ganar en rendimiento y espacio de almacenamiento, un ejemplo de ello es el trabajo con las sesiones en PHP, que es una caché con estas características pero que se almacena en un servidor web para un cliente web específico. (Marinescu, 2004)

La ventaja principal de usar un mecanismo para el almacenamiento de la información que cambia poco como la configuración es que puede mejorar considerablemente el rendimiento de la aplicación, evitando tener que construir por ejemplo estructuras complejas, configuraciones, e incluso realizar peticiones innecesarias a bases de datos y aumentando el uso concurrente de recursos que varían poco en el tiempo, además puede ser usada con otros fines como la comunicación entre sesiones. Actualmente el marco de trabajo del proyecto Cedrux y de la UCID no cuenta con un componente para la gestión de la caché y debido a ello se decidió desarrollar, adaptar o integrar un componente al marco de trabajo con este objetivo (Equipo Arquitectura del ERP Cedrux, 2008).

Trazas:

Las trazas permiten en el desarrollo de software crear un mecanismo de registro oficial de eventos durante un período de tiempo en particular, además de registrar datos o información sobre quién, qué, cuándo, dónde y por qué un evento ocurre para un dispositivo en particular o aplicación. Todo esto permite monitorear las actividades de la aplicación o dispositivo, donde se puede obtener una buena oportunidad para determinar eventos y tomar la acción necesaria para corregir el problema o iniciar una investigación en caso de un incidente de seguridad (Equipo Arquitectura del ERP Cedrux, 2008).

Excepciones:

Los programadores de cualquier lenguaje se esfuerzan por escribir programas libres de errores, sin embargo, es muy difícil que los programas reales se vean libres de ellos. Aún cuando una sentencia o expresión sea sintácticamente correcta, puede causar un error al intentar

ejecutarla. Los errores que se detectan en la ejecución se llaman excepciones. Las excepciones están destinadas para la detección y corrección de errores. Si hay un error, la aplicación no debería morir y generar un core (o un crash en caso del DOS). Se debería lanzar (throw) una excepción que se debería capturar (catch) y resolver la situación del error. Utilizadas en forma adecuada, las excepciones aumentan en gran medida la robustez de las aplicaciones (Equipo Arquitectura del ERP Cedrux, 2008).

El sistema de gestión CEDRUX presenta estos mismos problemas pero no basta con el mero hecho del lanzamiento y captura de excepciones sino también se imponen algunos requerimientos descritos a continuación:

- La internacionalización de las excepciones.
- Manejo declarativo del comportamiento de las excepciones.
- Mecanismo extensible de creación de nuevos tipos de excepciones.
- Codificación de las excepciones.
- Tratamiento igualitario de las excepciones propias o de los marcos de desarrollo subyacentes (Doctrine, Zend y EzComponent) que ya poseen su propia forma de gestionarlas.

Validaciones:

La seguridad de las aplicaciones web es un tema muy importante ya que en esta se manejan informaciones de muchas instituciones, y si personal no autorizado tuviese acceso a ellas podría causar daños irreparables. Dada la importancia del ERP y toda la información que maneja se decidió prestar el máximo de interés a las validaciones de cada una de las aplicaciones que conforman este software. En la mayoría de las aplicaciones sólo se valida en la capa de presentación, esto trae como consecuencia que si el desarrollador hizo una mala validación o el atacante logra violar esta capa entonces tiene acceso total a la información almacenada. Como la seguridad se basa en la cantidad de barreras que un desarrollador implementa, desde la vista que se le muestra al usuario hasta los datos, se decidió desarrollar un componente para validar en el servidor todas las acciones que requiera de ellas.

Este componente debe validar todos los datos y acciones que se activan ya sean por un usuario o por un sistema externo, que serán ejecutadas en un controlador de una aplicación en

específico. La acción será validada antes de llegar al controlador para no hacer peticiones innecesarias a este. (Equipo Arquitectura del ERP Cedrux, 2008)

Conceptos Globales:

Los contenedores de variables globales son aspectos muy importantes en los frameworks ya que agrupan información común conceptualizada que se maneja en todo el sistema, o sea en todos los módulos y sesiones de una aplicación. Por lo general todo framework que se implementa contiene un componente de este tipo para que cada parte del sistema pueda acceder de forma fácil y sencilla a datos que oferte otro sin incurrir en códigos engorrosos que implican gastos de tiempo y esfuerzos para los programadores. Por lo tanto la implementación de estos contenedores de información común permite que la misma sea almacenada una sola vez y actualizada en el momento conveniente, así cuando un componente necesite de esta información la toma del contenedor sin tener que estar realizando peticiones constantes a los servicios que oferten los demás componentes.

Por lo general los frameworks permiten que en el desarrollo de una aplicación quede implementado que en un determinado momento se guarde la información común en el contenedor. Posteriormente un módulo accede a la variable que almacena la información en el mismo tantas veces le haga falta, pero internamente se invoca una sola vez al servicio de la parte que oferta la misma.

Un ejemplo de esto es el manejador de sesiones que implementa CodeIgniter, que al inicio de la aplicación permite implementar de manera fácil que sean guardados entre otros datos los del usuario que ejecuta el sistema y los coloca en las cookies del navegador, de esta forma es fácil para los módulos acceder a estos datos cuantas veces lo necesiten sin derrochar tiempo durante la ejecución de la aplicación.

En el framework que se desarrolla en el seno del equipo de arquitectura del ERP (Cedrux) para que un módulo pueda acceder a información que oferta otro, debe pedírselo a través de un servicio cada vez que lo necesite, esto trae consigo muchas peticiones entre los módulos y por lo tanto genera una lentitud mayor cuando se ejecuta cada componente del sistema, ya que se generan un sin número de peticiones concurrentes desde cada módulo a determinadas partes de la aplicación que ofertan información común a cada fragmento del sistema.

Es por ello que se decide implementar un componente capaz de obtener toda la información común conceptualizada en variables para que cada módulo pueda acceder a la misma de

forma rápida y sin generar lentitudes al sistema. Este es un componente de conceptos globales cuyo nombre es GlobalConcept. (Equipo Arquitectura del ERP Cedrux, 2008)

1.3.1 Estado actual de las herramientas y paradigmas que garantizan la configuración de aspectos arquitectónicos en el mundo, nuestra universidad y el ERP Cedrux.

La configuración dinámica y extensible de los Aspectos Arquitectónicos es una metodología en discusión actualmente en los marcos de trabajo e ¹IDE del lenguaje PHP en el mundo actual. Lo cual trae consigo que estas herramientas no hayan incluido módulos y/o componentes en su desarrollo para lidiar con esta nueva tecnología. Debido a esto las soluciones informáticas que siguen esta línea deben implementar sus propios gestores para configurar la arquitectura base de sus aplicaciones, o de lo contrario usar librerías y herramientas desarrolladas por las comunidades internacionales de software, las cuales aportan grandes ventajas pero aún no alcanzan un estado maduro de servicios.

Entre los principales marcos de trabajo que tratan el lenguaje PHP podemos encontrar:

- Zend Studio
- Eclipse
- PDT (PHP Development Tools, Eclipse)
- Aptana (Eclipse)
- Php Designer
- PhpEdit

Ninguno de ellos actualmente tiene incluido dentro de su IDE un gestor para el tema de la configuración de aspectos arquitectónicos. Por lo que la productividad del desarrollo web en este sentido se observa muy disminuida.

Como consecuencia nuevos proyectos e inversiones de software en el ámbito del software libre como lo es el naciente ERP-Cuba se ven obligados a crear sus propias herramientas para la gestión de este concepto, precisamente en nuestro proyecto se llevó a cabo la confección de un componte llamado ZendExt-Aspect.

¹ Integrated Development Enviroment

1.4 Vista tecnológica de los componentes del IDE para el marco de trabajo de Cedrux

1.4.1 Selección de la plataforma:

En esta sección se realiza la valoración y evaluación de las plataformas tecnológicas y la selección de la más óptima teniendo en cuenta las características propias del proyecto, los recursos materiales y financieros y el nivel de conocimiento de los recursos humanos disponibles en esta plataforma.

Debido a las características de la enseñanza y la producción en nuestra universidad, las aplicaciones de escritorio construidas son generadas a partir de dos plataformas principales:

- J2EE
- .NET

Por tanto, debido al estrecho marco de tiempo que caracteriza la entrega del proyecto y a la preparación de los recursos humanos, serán tomadas estas dos plataformas en cuenta para el desarrollo del IDE.



Plataforma J2EE:

Una plataforma es el entorno de hardware o software en el cual se ejecuta un programa.

A diferencia de las plataformas tradicionales como Linux, Windows y Solaris, la plataforma Java está basada solo en software que se ejecuta sobre otras plataformas basadas en hardware. Por esto la plataforma Java y su software pueden ser utilizados sobre variados sistemas operativos y hardware. La plataforma Java está constituida de tres componentes: el lenguaje, la máquina virtual y las bibliotecas. (Sun Microsystems, Inc., 2007)

Java 2 Platform, Enterprise Edition es una solución dada por *Sun Microsystems* para la creación de servidores de aplicación basado en el lenguaje Java. J2EE es una especificación,

un concepto, que define una arquitectura para el desarrollo de aplicaciones basada en componentes que los servidores de aplicación J2EE- compatibles deben cumplir.

J2EE es una buena solución para el desarrollo de aplicaciones empresariales porque permite reducir el tiempo de desarrollo y mejorar la calidad del producto. Sin embargo, J2EE no es simple porque consta de variados componentes y unifica una gran cantidad de conceptos. Entre las nuevas componentes destacan servlets y ¹JSP para generar contenidos dinámicos, y ²EJB por sus posibilidades de encapsular el acceso a una fuente de datos permitiendo el uso de objetos persistentes en el tiempo, centralizar la lógica del negocio y permitir utilizar programación distribuida. Sin embargo, J2EE y sus componentes son relativamente nuevos, con tres años desde la publicación de su primera versión, por lo que no son muy conocidas sus cualidades y objetivos lo que hace necesario un estudio profundo de este. (Alur, Crupi, & Malks, 2004)



Plataforma .NET 2.0

.NET es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.

.NET podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Sun Microsystems y a los diversos framework de desarrollo web basados en PHP. Su propuesta es ofrecer una manera rápida y económica, y a su vez segura y robusta, de desarrollar aplicaciones o como la misma plataforma las denomina “soluciones”, permitiendo una integración más rápida y ágil entre

¹ Java Server Pages

² Enterprise JavaBeans

empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo.

A fines del 2005 Microsoft lanzó la nueva versión de su plataforma de desarrollo .NET 2.0. Dentro de este paquete de actualizaciones existen cambios para todo el sistema de desarrollo comprendiendo entorno de desarrollo, aplicaciones de base de datos, lenguajes, frameworks y otras novedades del sistema.

Nuevos entornos de desarrollo:

- Visual Studio 2005.
- Visual Web Developer 2005 Express

Nuevas Base de Datos:

- SQL Server 2005
- SQL Server 2005 Express

Nuevos Lenguajes:

- Visual Basic .NET 7.0
- C# 2.0

Nuevos FrameWorks:

- .NET Framework 2.0
- .NET Compact Framework 2.0

Al igual que la anterior plataforma, la versión 2.0 mantiene todas las prestaciones que la convirtieron en un producto de excelente calidad para el desarrollo de sistemas empresariales, orientados tanto a aplicaciones de escritorio como a servicios web.

Esta tecnología ofrece soluciones a los actuales problemas de programación, como son la administración de código o la programación para Internet. Para aprovechar al máximo las características de .NET es necesario entender la arquitectura básica en la que está implementada esta tecnología y así beneficiarse de todas las características que ofrece esta nueva plataforma.

1.4.2 Selección de las características a evaluar en la selección de la plataforma tecnológica.

- Par 1: Conocimiento de los recursos humanos sobre las características y el manejo de las aplicaciones en la plataforma.
- Par 2: Seguridad de los distintos sistemas. Vista desde todos los niveles de aplicación, datos, lógica e interfaz de usuario.
- Par 3: Gestión de Datos, almacenamiento y servicios.
- Par 4: Framework de trabajo con una interfaz amigable, que presente una curva ligera de aprendizaje en vistas de la rapidez de las entregas del producto.
- Par 5: Accesibilidad (De código abierto o cerrado).
- Par 6: Popularidad y desarrollo.
- Par 7: Portabilidad (Sistema operativo con el cual se ejecuta).

1.4.3 Calificación y cómputo de totales de cada plataforma.

Se lleva a cabo la calificación y cómputo de las plataformas mediante la realización de la tabla de cómputo de los parámetros en la que se asigna una calificación a las características que brinda cada plataforma, permitiendo realizar una comparación entre ellas. Los parámetros están valorados en una escala de valores del 1 al 5 de forma ascendente.

Tabla 3: Parámetros de importancia para la investigación en la comparación entre plataformas.

CARACTERÍSTICAS A EVALUAR	.NET	J2EE
Parámetro 1	10*	3
Parámetro 2	5	5
Parámetro 3	5	4
Parámetro 4	10*	4
Parámetro 5	5	4
Parámetro 6	5	4
Parámetro 7	3	5

* Los elementos marcados poseen características que destacan el nivel por las necesidades del proyecto.

Fuente: Vargas D. y Hernández M. 2009

1.4.4 Comparación entre plataformas.

Existen pocas alternativas que compitan directamente con J2EE y la tecnología .NET, debido a que muchos proveedores en vez de crear productos similares a estos, escogen implementar plataformas compatibles. (Barrios, 2005)

Microsoft es el único proveedor que ofrece una plataforma para desarrollo de aplicaciones empresariales que compite directamente con J2EE. Su producto es conocido como .NET, anteriormente llamado DNA, que es un modelo de desarrollo para aplicaciones empresariales para la plataforma Windows. Existen analogías directas entre .NET y J2EE como ¹ADO, ²ActiveX, ASP, ³COM+. Sin embargo, existen diferencias entre .NET y J2EE, siendo la diferencia principal la independencia del lenguaje de .NET v/s independencia del proveedor de J2EE. J2EE es una plataforma independiente de un proveedor, es una aplicación que cumple con los estándares de J2EE puede ser ejecutada en cualquier servidor de aplicaciones J2EE compatible, sin embargo, tiene una limitación, las aplicaciones J2EE deben ser implementadas en Java. (Barrios, 2005)

Microsoft .NET es una plataforma independiente del lenguaje, así los desarrolladores pueden elegir el lenguaje y las herramientas a utilizar para implementar aplicaciones .NET.

Entre los lenguajes a elegir se encuentran Visual Basic, Visual C++, Visual J++ y Visual C#, esto se logra transformándolos a un lenguaje intermedio e independiente del lenguaje de programación (IL). Sin embargo, para ser ejecutado se debe utilizar la plataforma Windows y su software como IIS, DTC, OLE DB e Internet Explorer si se utiliza ActiveX.

J2EE es una plataforma más madura, se creó primero y su avance es muy rápido dado los aportes de muchas personas a través del mundo que apoyan el software Java. Por otra parte Microsoft .NET ofrece una solución integrada en la plataforma Windows, donde Microsoft hace

¹ Es un API que permite ser usado por programadores para acceder a datos y servicios de datos.

² Es un marco de trabajo para definir componentes de software reusables.

³ Component Object Model

el esfuerzo en asegurar que sus productos pueden trabajar juntos sin mayores problemas de configuración.

Dada las distintas tecnologías para desarrollar aplicaciones, es que se han creado métodos para poder relacionarlas. Estas son conocidas como tecnologías de integración de aplicaciones, y están compuestas principalmente por ¹CORBA y Servicios Web. CORBA permite la comunicación de aplicaciones independiente de la arquitectura, el sistema operativo, el lenguaje de programación y el tipo de red utilizada (ver sección 5.3), si bien su ofrecimiento es bueno, no ha logrado posicionarse como la mejor solución existente ya sea por su complejidad o por las pocas implementaciones existentes. Los Servicios Web constituyen un conjunto de tecnologías que utilizan XML para comunicarse e intercambiar información, usando un protocolo de comunicación estándar basado en XML conocido como ²SOAP.

Los Servicios Web son muy recientes, pero han logrado un buen éxito inicial por comunicarse a través de HTTP. Por tanto una integración de los servicios brindados por el proyecto IDE para el ERP cubano podría comunicarse con cualquier otra plataforma de configuración mediante servicios CORBA y simplemente SOAP.

Tabla 4: Principales elementos de comparación entre las plataformas.

Aspectos	.NET	J2EE
Lenguaje de programación	C#...	Java
Plataforma	Windows	...
Máquina virtual	CLR [Common Language Runtime]	JVM [Java Virtual Machine] / JRE [Java Runtime Environment]
Lenguaje interpretado	MSIL	Java bytecodes
Cientes Windows	.NET Forms	Swing
Cientes Web	ASP.NET	JSP / Servlets
Servidores de	...	EJB [Enterprise Java Beans]

¹ Common Object Requesting Broker Architecture es un estandar definido por Object Management Group (OMG) el cual permite componentes de software escritos en múltiples lenguajes de máquina y trabajen juntos en distintos ordenadores.

² Simple Object Access Protocol

aplicaciones		
Acceso a bases de datos	ADO.NET	JDBC / EJB-SQL
Paso de mensajes	MSMQ	JMS [Java Messaging Service] / Msg EJBs
Integración con sistemas previos	COM TI	JCA [Java Connector Architecture]

Fuente: Vargas D. y Hernández M. 2009

1.4.5 Selección de la plataforma apropiada.

Luego de analizar los resultados de la comparación en base a los parámetros definidos como sobresalientes, se determina como plataforma a utilizar, la plataforma .NET. En la comparación marcan pauta principalmente los parámetros 1 y 4 debido a las características del desarrollo del ERP cubano. (Campos & Martínez, 2006)

1.4.6 Selección del framework y las librerías

En esta sección se realiza la valoración y evaluación de los frameworks y librerías para la plataforma tecnológica seleccionada y la selección de los frameworks y librerías más óptimas según las características propias del proyecto, los atributos de calidad y las categorías o aspectos arquitectónicamente significativos de la aplicación, los recursos materiales y financieros y el nivel de conocimiento de los recursos humanos disponibles. (Canos & Penades, 2006)

Framework .NET.

El framework .NET es la propuesta de Microsoft para el desarrollo de aplicaciones completamente orientadas a objetos, seguras, sencillas de instalar y multiplataforma; no es un lenguaje de programación: es un conjunto de tecnologías de software que permite el desarrollo de aplicaciones de escritorio (Windows), Web(ASP.Net), bibliotecas de código (componentes reutilizables: DLL), dispositivos móviles, controles de usuario, acceso a datos, reportes y muchos más que se ejecutan bajo el .NET Framework. Es completamente extensible: podemos heredar de sus clases base y modificar su comportamiento o implementar sus interfaces para desarrollar nuevas ideas totalmente originales. (Avila, 2005)

El entorno de programación de la plataforma .NET, *Visual Studio*, es uno de los más potentes en cuanto a facilidad de uso. Además proporciona una gran cantidad de herramientas para mejorar la productividad del programador. Es independiente del lenguaje de programación, por lo que se pueden realizar componentes en un lenguaje y utilizarlo en otro, sin complicaciones ni pérdida de rendimiento, debido a su sistema de tipos comunes y a su arquitectura de lenguaje intermedio (*MSIL*).

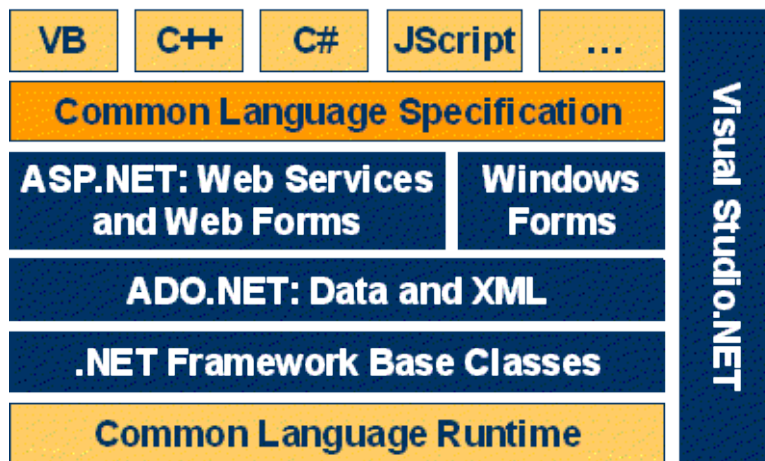


Figura 1: *Arquitectura base del framework .NET.*

Fuente: Vargas D. y Hernández M. 2009

CLR

El **Common Language Runtime (CLR)** es el núcleo de la plataforma .NET ya que es el encargado de gestionar la ejecución de código compilado para la plataforma .NET. Puede asimilarse a la *máquina virtual de Java*. (Microsoft Corporation. MSDN Community, 2009)

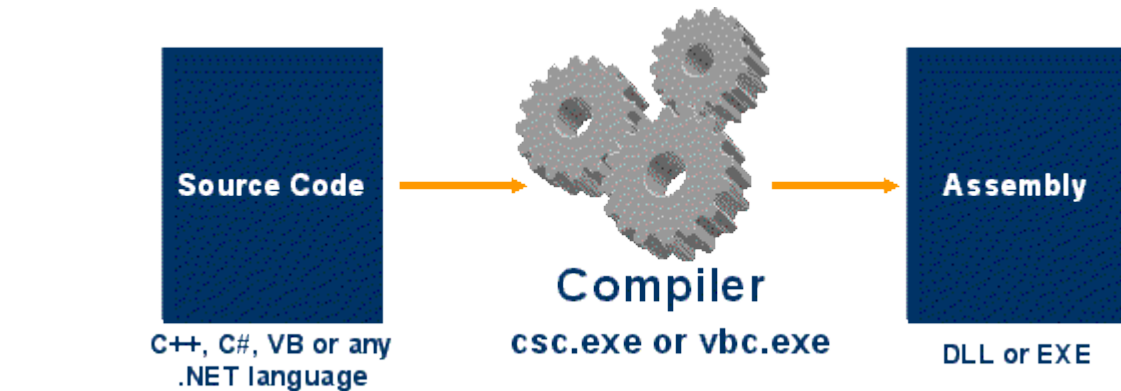


Figura 2: Compilador de la plataforma .NET.

Fuente: Vargas D. y Hernández M. 2009

Las dos principales características del CLR son:

- **Ejecución multiplataforma:** El CLR actúa como una máquina virtual, encargándose de ejecutar las aplicaciones diseñadas para la plataforma .NET. Su especificación está abierta, por lo que cualquier plataforma para la que exista una versión del CLR podrá ejecutar cualquier aplicación .NET.
- **Integración de lenguajes:** El CLR asocia cualquier lenguaje para el que exista un compilador que genere código para la plataforma .NET. Es decir el CLR actúa como intérprete entre los lenguajes integrados en la plataforma.

Otras características destacables son:

- **Modelo de programación consistente:** A todos los servicios y facilidades ofrecidos por el CLR se accede de la misma forma: a través de un modelo de programación orientado a objetos.
- **Gestión de memoria:** El CLR incluye un **recolector de basura** que evita que el programador tenga que tener en cuenta cuándo ha de destruir los objetos que dejen de serle útiles. Gracias a este recolector se evitan errores de programación muy comunes como intentos de borrado de objetos inexistentes, agotamiento de memoria por olvido de eliminación de elementos inutilizables o solicitud de acceso a miembros de objetos ya destruidos.
- **Seguridad de tipos:** El CLR facilita la detección de errores de programación difíciles de localizar comprobando que toda conversión de tipos que se realice durante la ejecución

de una aplicación .NET se haga de modo que los tipos origen y destino sean compatibles.

- **Aislamiento de procesos:** El CLR asegura que desde el código perteneciente a un determinado proceso no se pueda acceder al código o datos pertenecientes a otro, ni se permite acceder a posiciones arbitrarias de memoria.
- **Tratamiento de excepciones:** En el CLR todos los errores que se puedan producir durante la ejecución de una aplicación se propagan de igual manera: mediante excepciones.
- **Soporte multihilo:** El CLR es capaz de trabajar con aplicaciones divididas en múltiples hilos de ejecución que pueden ir evolucionando por separado en paralelo o intercalándose, según el número de procesadores de la máquina sobre la que se ejecuten. Las aplicaciones pueden lanzar nuevos hilos, destruirlos, suspenderlos por un tiempo o hasta que les llegue una notificación, enviarles notificaciones, sincronizarlos, etc.
- **Seguridad avanzada:** El CLR proporciona mecanismos para restringir la ejecución de ciertos códigos o los permisos asignados a los mismos según su procedencia o el usuario que los ejecute. (Microsoft Corporation, 2006)

Ensamblados

Constituyen la unidad lógica de despliegue en la plataforma .NET. Un assembly incluye metadatos acerca de los componentes incluidos en el assembly (versiones, tipos, dependencias...), metadatos acerca de los tipos incluidos (propiedades, atributos, métodos, firmas, clases base...), el código intermedio MSIL (Microsoft Intermediate Language, similar a los bytecodes de Java) y los recursos adicionales que sean necesarios (imágenes, textos...). En fin, es muy parecido a los ficheros ¹JAR de Java.

En la plataforma .NET, una aplicación está formada por uno o varios assemblies. Al poder coexistir distintas versiones de un assembly, se eliminan muchos de los problemas que caracterizan a las aplicaciones típicas de Windows, facilitando el despliegue, actualización y eliminación de aplicaciones. De hecho, una aplicación concreta podría utilizar simultáneamente varias versiones de un assembly.

¹ Java Archives

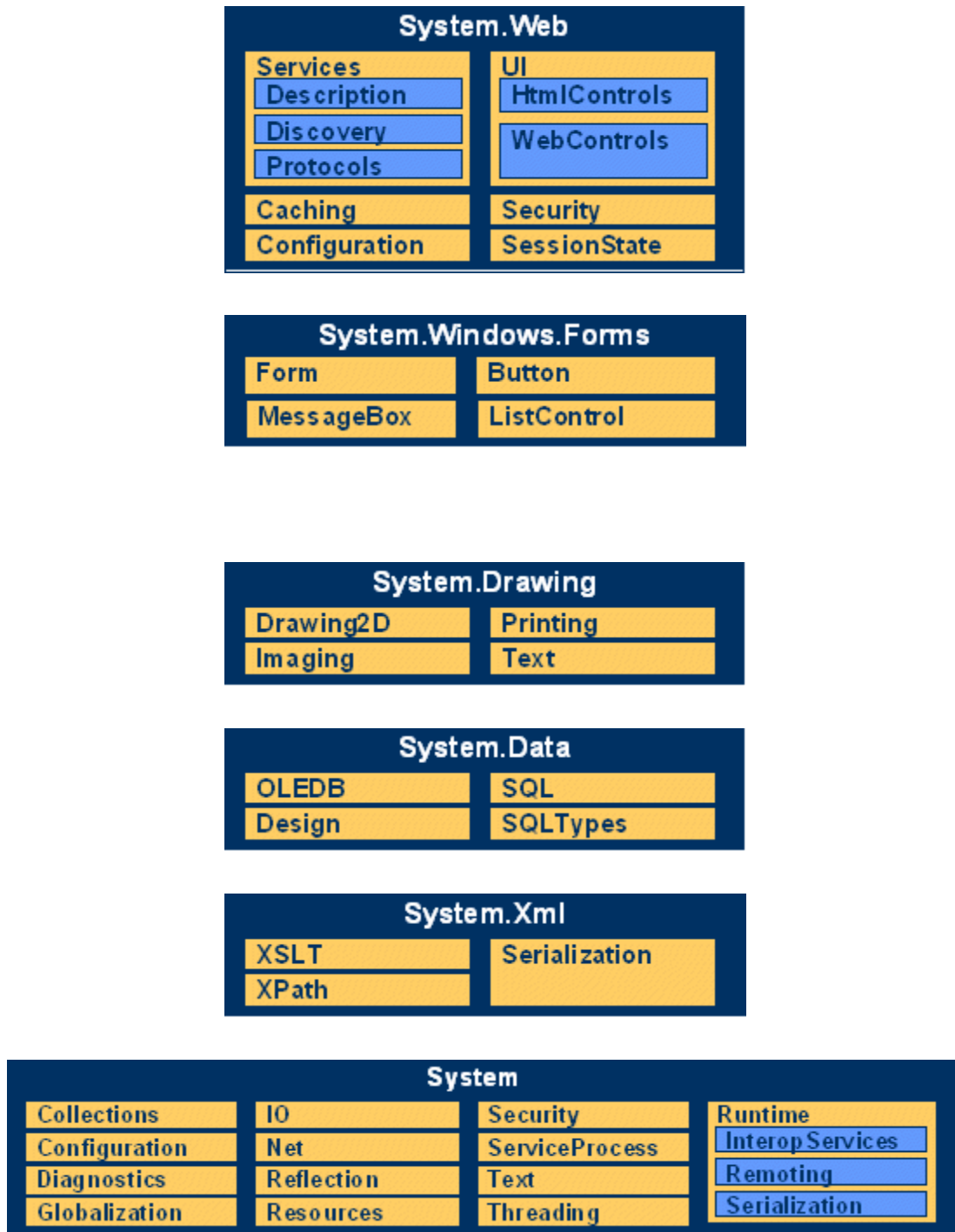


Figura 3: Biblioteca de clases .NET.

Fuente: Vargas D. y Hernández M. 2009.



ADO.NET

Similar a ADO, proporciona acceso a datos de forma independiente al lenguaje de programación que utilizemos. Los datos se pueden ver y procesar de forma relacional (tablas) o jerárquica (XML).

Desarrollo de aplicaciones. Formularios Windows

Los formularios Windows están contruidos sobre la base de la plataforma .NET y permiten construir complejas aplicaciones Windows en un entorno de desarrollo visual de aplicaciones ¹RAD, tal como hasta ahora se venía haciendo con lenguajes del estilo de Visual Basic o Delphi.

Soporte para múltiples lenguajes

Para permitir el desarrollo de componentes utilizando múltiples lenguajes de programación, la plataforma .NET establece un sistema de tipo común (CTS: Common Type System) y una especificación que permite que puedan interactuar fragmentos de código escritos en distintos lenguajes (CLS: Common Language Specification).

La plataforma .NET permite utilizar una amplia gama de lenguajes de programación, como es el caso de

- C#: Un nuevo lenguaje creado para la plataforma .NET. Se puede considerar una versión "segura" de C++ y viene a ser un híbrido entre Java (de Sun) y Delphi (de Borland). Es un lenguaje de programación orientado a objetos que pretende facilitar el desarrollo de componentes de software robustos y duraderos que preserven la inversión realizada en su desarrollo.
- Visual Basic .NET: Moderniza y simplifica el lenguaje de programación Visual Basic, con algunas novedades sintácticas, herencia simple, tratamiento de hebras y manejo de excepciones.

¹ Rapid Application Development

Ejecución de código

Para que un lenguaje de programación sea soportado por la plataforma .NET, debe existir un compilador que traduzca de este lenguaje a MSIL ("managed code"). A la hora de ejecutar el código intermedio este es siempre compilado a código nativo.

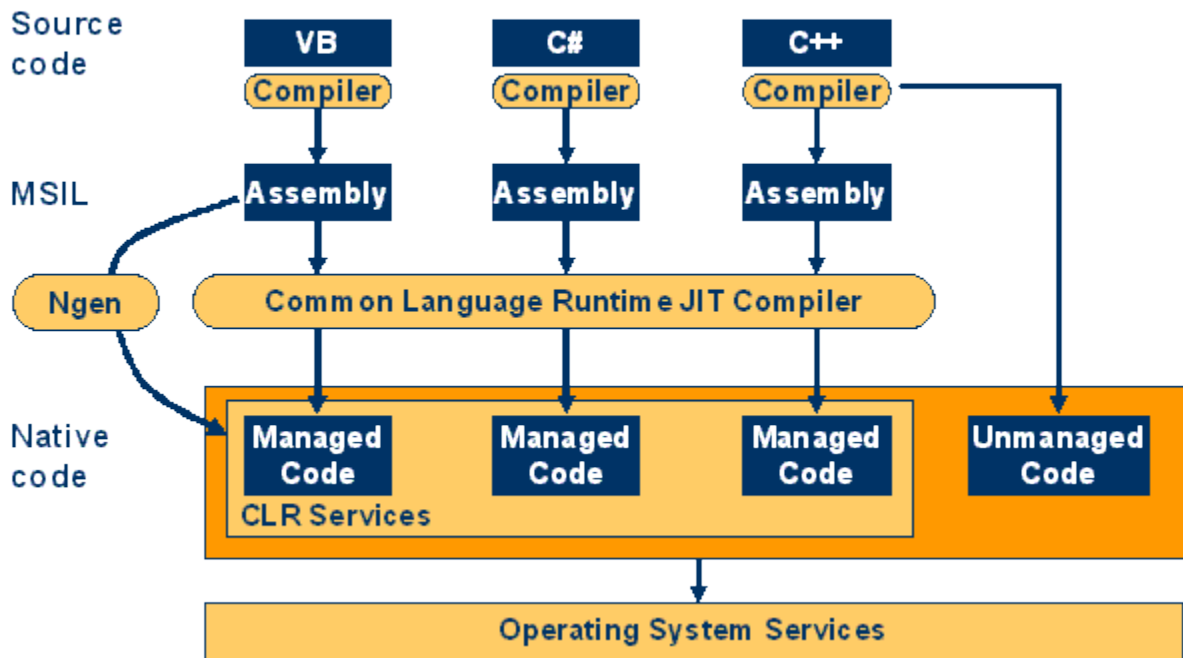


Figura 4: Estructura del compilador de la plataforma .NET.

Fuente: Vargas D. y Hernández M. 2009.

1.6 Arquitectura taxonómica:

La arquitectura taxonómica de cada componente ha sido modelada en una variante basada en el modelo de tres capas. La capa de presentación constituida por los formularios Web ve la capa de negocio y se comunica con esta empleando objetos de la capa de negocio y parámetros de consulta representados en tipos de datos del lenguaje C#. La capa de negocio esta compuesta de las entidades y los gestores de lógica, no tiene acceso a la capa de presentación, y se comunica con esta capa a través de las entidades del negocio y los tipos de datos abstractos implementados en la tecnología de desarrollo .NET para el transporte de datos entre capas arquitectónicas como Dataset o Datatable. Por otra parte la capa de negocio tiene visibilidad sobre la capa de acceso a datos y se comunica con ella a través de las

entidades del negocio, mediante objetos Datasets, Datatables o mediante atributos. La capa de acceso a datos solo tiene visibilidad a las entidades de la capa de negocio y es quien implementa las especificaciones del tipo de fuente de persistencia de estas entidades, responsabilizando a clases que modelan función de fábricas de cada una de las clases del negocio que necesitan de persistencia.

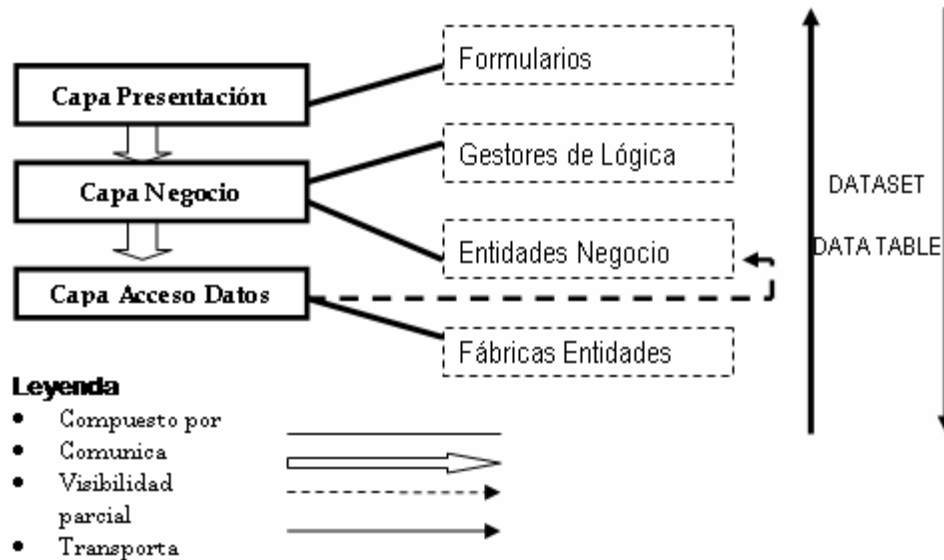


Figura 5: Arquitectura en tres capas

Fuente: (Lazo & Yzquierdo, 2007)

1.7 Esquema de persistencia

La implementación del esquema de persistencia de los componentes se realizó basada en las herramientas para la gestión de archivos xml que brinda la plataforma .NET. Las cuales brindan facilidades de lectura y escritura mediante las clases xmlreader¹ y xmlwriter¹.

¹ Reader: Representa un lector que proporciona acceso rápido a datos XML, sin almacenamiento en caché y con desplazamiento sólo hacia delante.

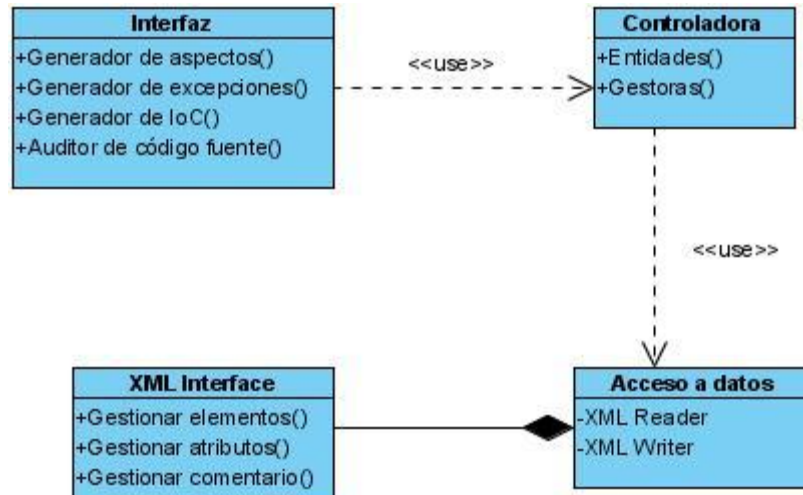


Figura 6: Esquema de persistencia

Fuente: Vargas D. y Hernández M. 2009.

Elementos de reutilización

Reutilizar, ya sean elementos de diseño o código, disminuye el tiempo de desarrollo y abarata los costos de producción, a lo que se le podría añadir que permite concentrarse en el dominio del problema y no en problemas que ya se encuentran evaluados, analizados e incluso solucionados por otras personas. Un elemento importante para lograr efectividad y buen impacto organizativo en el diseño, consiste en institucionalizar la reutilización. Una idea que podría apoyarse es la de incentivar la búsqueda de oportunidades para reutilizar desde una perspectiva coordinada. (López, Laguna, & Marqués, 2005)

La reutilización efectiva del diseño se alcanza mejor asignándoles esta tarea a personas en específico, los que serían los responsables de garantizar la actividad concretamente además de extender el análisis hasta el impacto de dicha reutilización en la fase de implementación del sistema. (García & López, 2002).

¹ Writer: Representa un sistema de escritura que constituye un medio rápido, no almacenado en caché y de sólo avance para generar secuencias o archivos con datos XML.

Para nuestra investigación la planificación de la reutilización se realizó mediante las estrategias que propone RUP en sus fases y flujos de trabajo, específicamente diseño e implementación. Aunque si bien no es el único elemento que debe tener un buen diseño, la reutilización y más aún, la reutilización institucionalizada, es uno de los más importantes atractivos del proceso de diseño a nivel industrial. (Valdés, 2008)

Tabla 5: Elementos de reutilización.

Proceso	Código, Contexto, Ideas	Objetivo	Por Ciento
Construcción de los componentes.	Framework 2.0	Utilizar los artefactos que brinda para una mejor implementación.	100%
Construcción de Interfaz Visual.	SharpLibrary.dll y MagicLibrary.dll	Mantener una Configuración estandarizada o modular de los componentes del framework.	65%
Cálculo de CRC	MD5	Realizar un cálculo seguro, invariable y eficaz de los ficheros del ERP.	85%
Configuración de los XML	System.XML	Leer y Persistir en los XML.	80%
Gestionar datos	PostgreSQL 8.3	Gestionar información del Auditor de Código Fuente.	35%
Mostrar XML	WindowsControlLibrary	Presentar una vista previa de los XML.	15%

Fuente: Elaboración Vargas D y Hernández M. 2009

Resultados esperados.

Se obtuvo un sistema, el cual posibilita un mejor aprovechamiento del tiempo de desarrollo. Resultado que responde a las problemáticas existentes y expuestas en nuestro trabajo acerca de Cedrux. El mismo consta de varios componentes integrados dentro de la aplicación permitiendo al usuario realizar uso de sus diversas funcionalidades. Los componentes que brinda la aplicación son: Generador de Aspectos que consta de 7977 líneas de código y que se encuentra en su versión 1.0, un Generador de Excepciones que consta de 7484 líneas de código y que se encuentra en su versión 1.0, un Generador de Inversión de Control que consta de 8667 líneas de código y que se encuentra en su versión 1.0, y un componente Auditor de Código fuente de 5092 líneas de código y que se encuentra en su versión 1.0, constituyendo un total de 93826 de líneas de código que conforman a la aplicación.

Además brinda un Documento que permite conocer los requisitos funcionales y no funcionales más importantes que brinda dicha aplicación, el cual consta de 33 páginas para su explicación, también un Manual de Usuarios para brindar un mejor entendimiento y operatividad de la aplicación el mismo está compuesto por 50 páginas.

Constituyendo todos estos elementos una gama de facilidades para la confección, seguridad y control a través de todo el proceso de desarrollo del ERP.

Conclusiones:

En este capítulo se hizo un análisis de los principales problemas relacionados con la configuración manual de la arquitectura del marco de trabajo en las distintas líneas de desarrollo y módulos del ERP cubano Cedrux, que afectan el rendimiento de la producción. Se realizó un cuestionario a una muestra representativa de los trabajadores del proyecto, la cual reflejó la necesidad de desarrollar el componente para configuración e integración de la arquitectura. Se efectuó un estudio del estado actual del diseño de software. Así como también se realizó un estudio sobre el tema de la Inversión de Control y el estado actual del mismo en el mundo, nuestro país y la Universidad. Análogamente se efectuó un análisis de los aspectos arquitectónicamente significativos y el estado actual de los mismos en el mundo, nuestro país y la Universidad. Además se incluye la vista tecnológica de los componentes del IDE para el marco de trabajo de Cedrux a través de la cual se toma como decisión desarrollar el componente usando la plataforma .NET y el framework que propone la misma. Finalmente se muestran los elementos que fueron reutilizados en el desarrollo del componente.

Capítulo 2: Desarrollo de la solución.

Introducción

En este capítulo se lleva a cabo el análisis de la solución propuesta. Se realiza un levantamiento de los requisitos funcionales y no funcionales desarrollados. Se muestran los prototipos de interfaz relacionados con los escenarios de los requisitos funcionales descritos y la estructura de los tipos de relaciones que existen entre las clases, mediante el diseño de clases. Se presentan además los artefactos de implementación que se obtienen a través de la construcción de los componentes que integran el sistema y los diversos casos de prueba realizados a los mismos para lograr una mejor comprensión de cada escenario presente ante el usuario.

Requisitos

Los requerimientos son definidos por RUP (**R**ational **U**nified **P**rocess) como condiciones o capacidades que debe alcanzar un componente o un sistema. Establecen lo que se debe hacer mediante la identificación de las funcionalidades requeridas y las restricciones trazadas. (IBM Corporation, 2003) La validación de los requerimientos tiene como objetivo evitar resultados inesperados como: la pérdida de tiempo o costos excesivos para la entidad o empresa para la cual se desarrolla el mismo. La técnica de validación de los requerimientos enmarca importantes funcionalidades a tener en cuenta durante todo el proceso de validación, como: demostrar los principales conceptos del sistema, probar las opciones de diseño y entender mejor el problema y su solución. (IBM Corporation, 2003)

Una vez realizadas las técnicas de validación se obtendrá una primera visión del producto, dando paso a la identificación de los requisitos funcionales y no funcionales.

Requisitos funcionales: son capacidades o condiciones que el sistema debe cumplir.

Requisitos no funcionales: son propiedades o cualidades que el producto debe tener. Son las que hacen al producto atractivo, usable, rápido o confiable.

Levantamiento de requisitos de la solución propuesta.

Requisitos funcionales

En el (Anexo 2: Requisitos funcionales del componente Generador de excepciones.) se muestra una descripción detallada de los requisitos funcionales del componente Generador de excepciones. Los cuales están organizados primeramente por los que gestionan los ficheros XML físicamente y seguido por aquellos que gestionan los conceptos definidos para el componente. A continuación se representa la estructura de los mismos:

- R1 Gestionar excepciones.

Tabla 6: Estructura de los requisitos funcionales del componente Generador de Excepciones.

No. R	Estructura de los RF del componente Generador de Excepciones
R1.1	Crear fichero.
R1.2	Cargar fichero.
R1.3	Guardar fichero.
R1.4	Buscar.
R1.1.1	Gestionar excepciones (exception.xml).
R1.1.1.1	Gestionar excepciones.
R1.1.1.1.1	Adicionar.
R1.1.1.1.2	Modificar.
R1.1.1.1.3	Eliminar.
R1.1.1.2	Gestionar lenguajes.
R1.1.1.2.1	Adicionar.
R1.1.1.2.2	Modificar.
R1.1.1.2.3	Eliminar.
R1.1.2	Gestionar administrador de excepciones (managerexception.xml).
R1.1.2.1	Gestionar módulos.
R1.1.2.1.1	Adicionar.
R1.1.2.1.2	Modificar.
R1.1.2.1.3	Eliminar.

R1.1.2.2	Gestionar controlador.
R1.1.2.2.1	Adicionar.
R1.1.2.2.2	Modificar.
R1.1.2.2.3	Eliminar.
R1.1.2.3	Gestionar acción.
R1.1.2.3.1	Adicionar.
R1.1.2.3.2	Modificar.
R1.1.2.3.3	Eliminar.

Fuente: Vargas D. y Hernández M. 2009.

En el (Anexo 3: Requisitos funcionales del componente Generador de aspectos.) se muestra una descripción detallada de los requisitos funcionales del componente Generador de aspectos. Los cuales están organizados primeramente por los que gestionan los ficheros XML físicamente y seguido por aquellos que gestionan los conceptos definidos para el componente. A continuación se representa la estructura de los mismos:

- Gestionar aspectos.

Tabla 7: Estructura de los requisitos funcionales del componente Generador de Aspectos.

No. R	Estructura de los RF del componente Generador de Aspectos
R2.1	Crear fichero.
R2.2	Cargar fichero.
R2.3	Guardar fichero.
R2.4	Buscar.
R2.1.1	Gestionar aspectos arquitectónicos (aspects.xml).
R2.1.1.1	Adicionar.
R2.1.1.2	Modificar.
R2.1.1.3	Eliminar.
R2.1.2	Gestionar plantilla de aspectos (aspecttemplate.xml).
R2.1.2.1	Adicionar.
R2.1.2.2	Modificar.
R2.1.2.3	Eliminar.
R2.1.3	Gestionar tejedor de aspectos (weaver.xml).
R2.1.3.1	Gestionar módulos.
R2.1.3.1.2	Adicionar.
R2.1.3.1.3	Modificar.
R2.1.3.1.4	Eliminar.
R2.1.3.2	Gestionar controlador.

R2.1.3.2.1	Adicionar.
R2.1.3.2.2	Modificar.
R2.1.3.2.3	Eliminar.
R2.1.3.3	Gestionar acción.
R2.1.3.3.1	Adicionar.
R2.1.3.3.2	Modificar.
R2.1.3.3.3	Eliminar.

Fuente: Vargas D. y Hernández M. 2009.

En el (Anexo 4: Requisitos funcionales del componente Generador de inversión de control.) se muestra una descripción detallada de los requisitos funcionales del componente Generador de inversión de control. Los cuales están organizados primeramente por los que gestionan los ficheros XML físicamente y seguido por aquellos que gestionan los conceptos definidos para el componente. A continuación se representa la estructura de los mismos:

- Gestionar inversión de control.

Tabla 8: Estructura de los requisitos funcionales del componente Generador de Inversión de Control.

No. R	Estructura de los RF del componente Generador de Inversión de Control
R3.1	Gestionar fichero (ioc.xml).
R3.1.1	Crear fichero.
R3.1.2	Cargar fichero.
R3.1.3	Guardar fichero.
R3.2	Gestionar subsistema.
R3.2.1	Adicionar.
R3.2.2	Eliminar.
R3.3	Gestionar contrato.
R3.3.1	Adicionar.
R3.3.1.1	Gestionar prototipo.
R3.3.1.1.1	Adicionar.
R3.3.1.1.2	Modificar.
R3.3.1.1.3	Eliminar.
R3.3.2	Modificar.
R3.3.3	Eliminar.

Fuente: Vargas D. y Hernández M. 2009.

En el (Anexo 5: Requisitos funcionales del componente Auditor de Código Fuente.) se muestra una descripción detallada de los requisitos funcionales del componente Auditor de Código Fuente. Los cuales están organizados primeramente por los que gestionan los ficheros XML físicamente y seguido por aquellos que gestionan los conceptos definidos para el componente. A continuación se representa la estructura de los mismos:

- Auditor de Código Fuente.

Tabla 9: Estructura de los requisitos funcionales del componente Auditor de Código Fuente.

No. R	Estructura de los RF del componente Auditor de Código Fuente
R4.1	Gestionar proyecto.
R4.1.1	Crear fichero.
R4.1.2	Abrir fichero.
R4.2	Gestionar ficheros.
R4.2.1	Instanciar.
R4.2.2	Auditar.
R4.2.3	Chequear.

Fuente: Vargas D. y Hernández M. 2009.

Requisitos no funcionales

1. Usabilidad (USB)

El componente podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de la estructura de los XML del Sistema Integral de Gestión de Entidades Cedrux.

2. Rendimiento (REN)

Los tiempos de respuesta y velocidad de procesamiento de información serán rápidos, no mayores de 2000 milisegundos para las operaciones. Exceptuando al componente Auditor de Código Fuente debido a la gran cantidad de ficheros que debe cargar del sistema y realizar las operaciones estimadas. El tiempo estimado depende de peso de la aplicación a gestionar por el auditor.

3. Seguridad (SEG)

No presenta un sistema de autenticación, puesto que solo tendrán acceso a él las personas calificadas para interactuar con el manejo y configuración de los XML de “CedruX”.

4. Portabilidad (POR)

El componente solo trabajará sobre Windows. Para una próxima versión el sistema se compilará con el marco de trabajo de MonoProject logrando que sea multiplataforma.

5. Soporte (SOP)

El componente contará antes de su puesta en marcha con un período de pruebas, se le dará mantenimiento, configuración y se brindará el servicio de instalación.

6. Políticos Culturales (CUL)

El sistema solo podrá ser utilizado en territorio cubano y por las entidades autorizadas por el Ministerio de las FAR.

El producto no debe contener palabras en otros idiomas.

8. Software

- Sistema operativo Windows 98 o superior.
- Un servidor de base de datos PostgreSQL 8.3 o superior.

9. Hardware

Para el cliente:

- Requerimientos mínimos: Procesador Pentium IV a 2.4 GHz con 512 Mb de memoria RAM.
- Tarjeta de red. 100 Mb/S

Prototipo de interfaz

Los prototipos de interfaz son aquellos que permiten al cliente relacionar los requisitos funcionales establecidos por cada componente con cada escenario donde se encuentre. Además de propiciarle una mejor manipulación facilitan un entendimiento lógico del orden en que se deben ejecutar las operaciones y de las respuestas ofrecidas por el sistema según la acción realizada.

Debido a que el sistema es una aplicación de escritorio, no presenta la gran cantidad de efectos visuales y estilos que en la actualidad abundan en las aplicaciones web para la facilidad y riqueza del uso de la misma, tendiendo a confundir en la gran mayoría a los usuarios que la emplean. Por tanto las interfaces de la solución propuesta son lo mayormente flexibles para los usuarios según las necesidades y condiciones establecidas para la configuración dinámica de los ficheros que conforman el marco de trabajo del proyecto Cedrux. (Lazo & Yzquierdo, 2007). A continuación se establece una breve descripción de los prototipos de interfaz por los distintos componentes que integran al sistema, así como del orden lógico en que se deberán ejecutar.

1. El componente Generador de excepciones brinda una interfaz que le permite al usuario gestionar los ficheros sobre los que realizará los cambios o modificaciones pertinentes, presentando funcionalidades como: Crear, Cargar o Abrir. Posibilitando a que se gestionen los distintos conceptos presentes en la interfaz de excepciones como: módulos, controladores, acciones, excepciones y lenguajes; todos ejecutándose en el orden establecido según la lógica de la configuración de los ficheros de excepciones. Dichas interfaces brindan la facilidad de realizar las funcionalidades de CRUD (**C**reate, **R**ead, **U**ppdate, **D**elete), tanto sobre el árbol de navegación, como en el panel de edición, estas características le permiten al usuario optar por algunas de las opciones para una mayor manejabilidad de las interfaces según sus necesidades. Así como también presenta opciones de observar la configuración previa de los ficheros XML que se estén gestionando sin tener que acceder al archivo de manera física, y de buscar si existe algunos de los conceptos para la utilidad necesaria que desee darle el mismo.

- Prototipos de interfaz del componente Generador de excepciones.

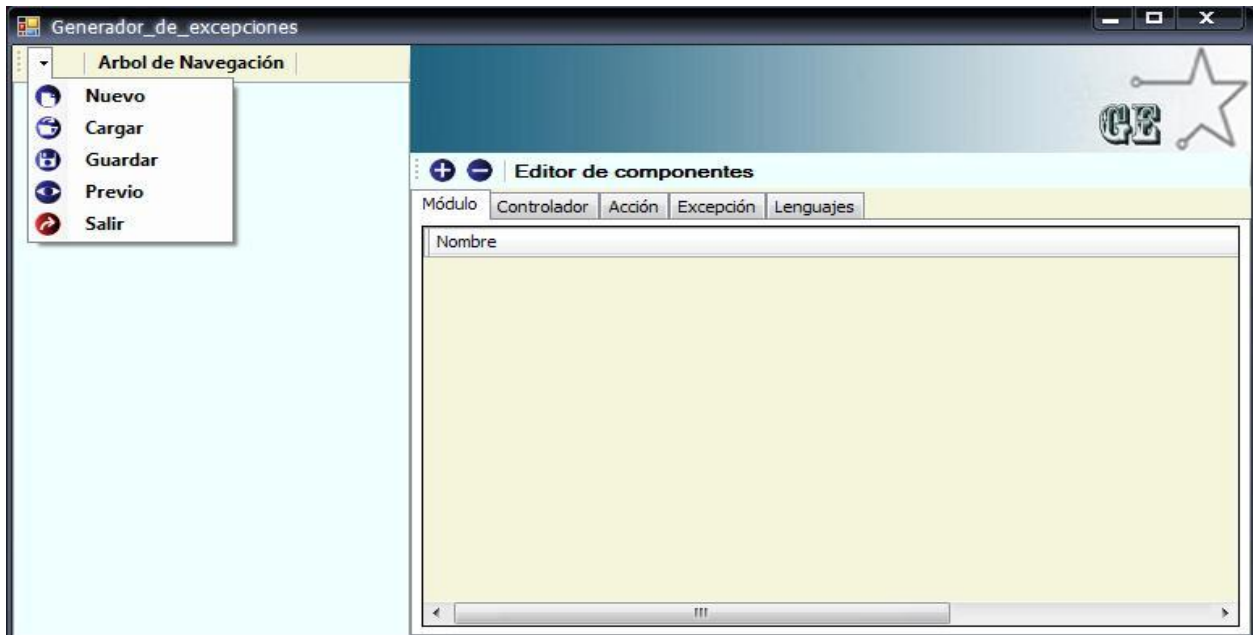


Figura 5: Interfaz que amerita al paquete de R. F. de la gestión de ficheros del componente Generador de excepciones.

Fuente: Vargas D. y Hernández M. 2009.

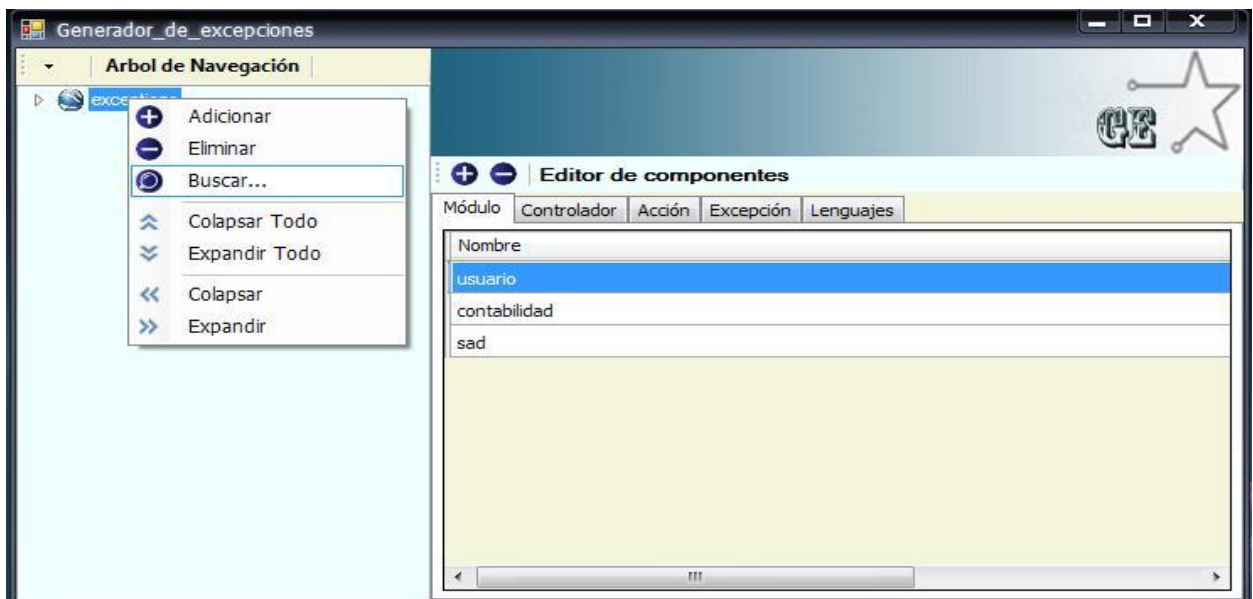


Figura 6: Interfaz que amerita al CRUD R1.1.2.1 Gestionar módulo.

Fuente: Vargas D. y Hernández M. 2009.

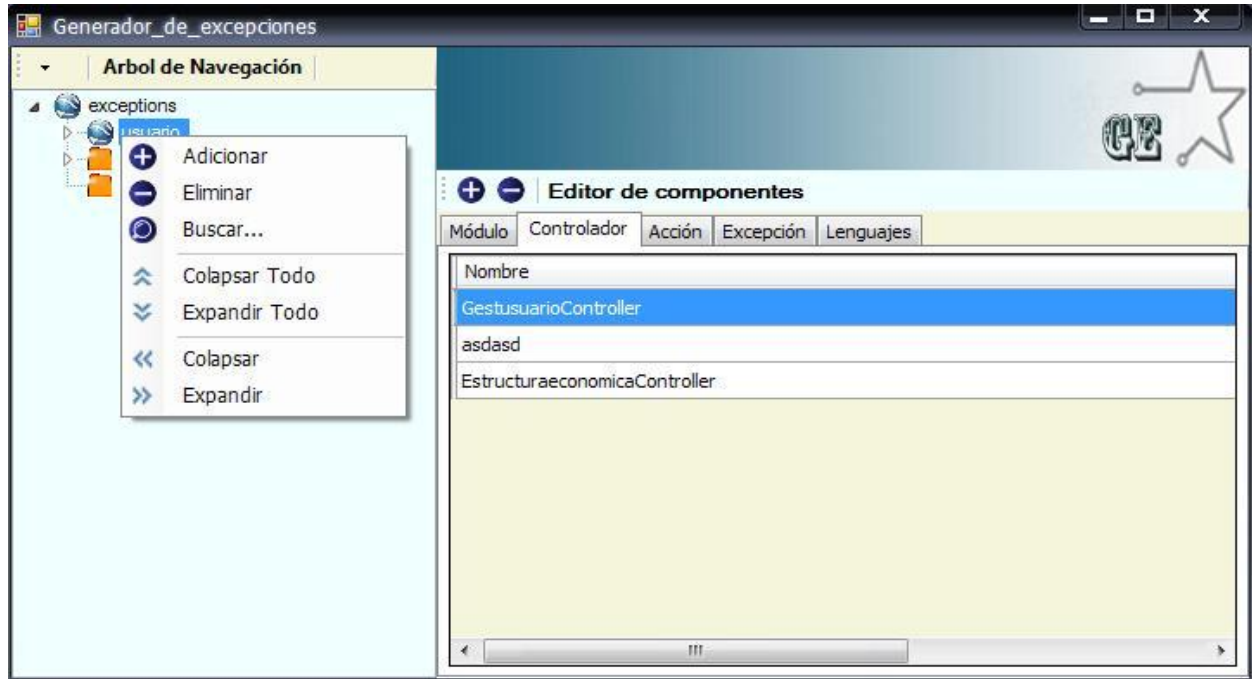


Figura 7: Interfaz que amerita al CRUD. del R1.1.2.2 Gestionar controlador.

Fuente: Vargas D. y Hernández M. 2009.

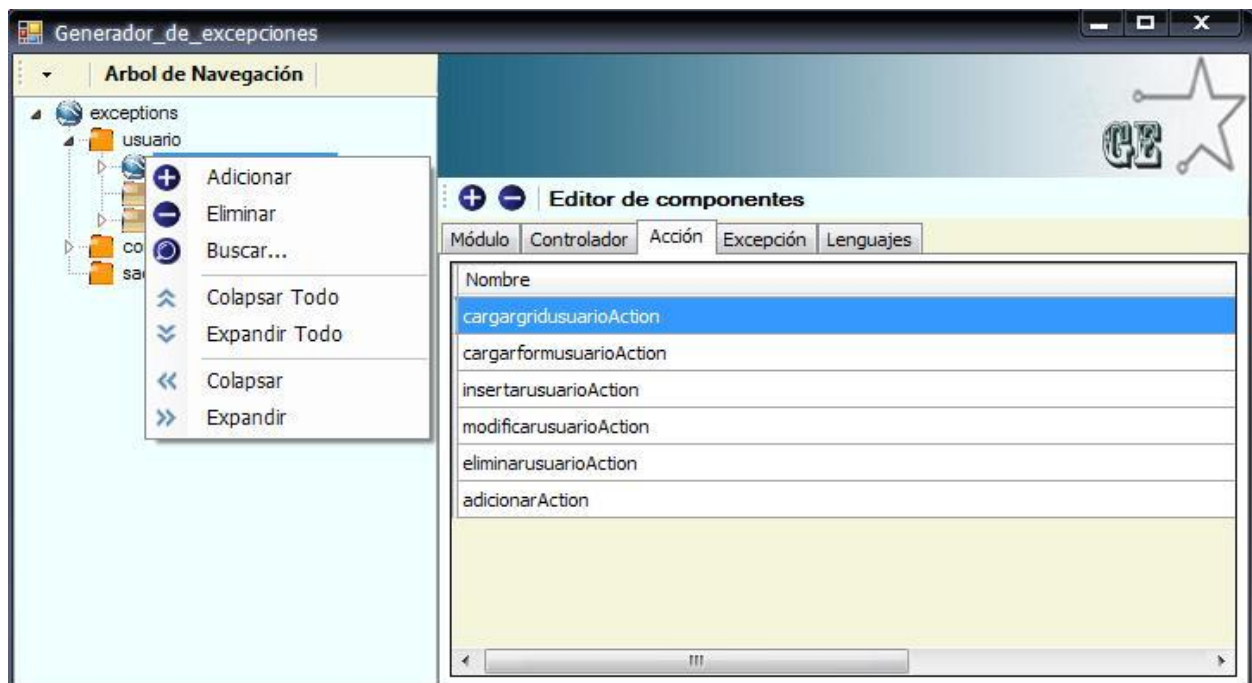


Figura 8: Interfaz que amerita al CRUD. del R1.1.2.3 Gestionar acción.

Fuente: Vargas D. y Hernández M. 2009.

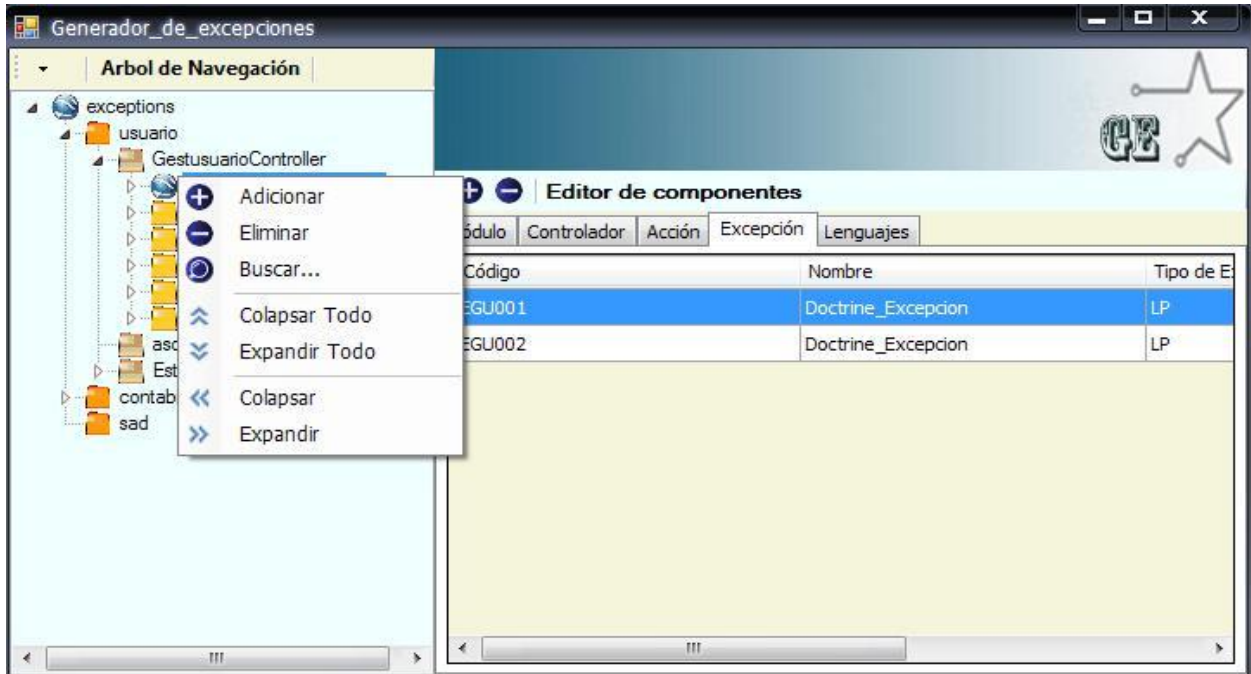


Figura 9: Interfaz que amerita al CRUD. del R1.1.1.1 Gestionar excepción.

Fuente: Vargas D. y Hernández M. 2009.

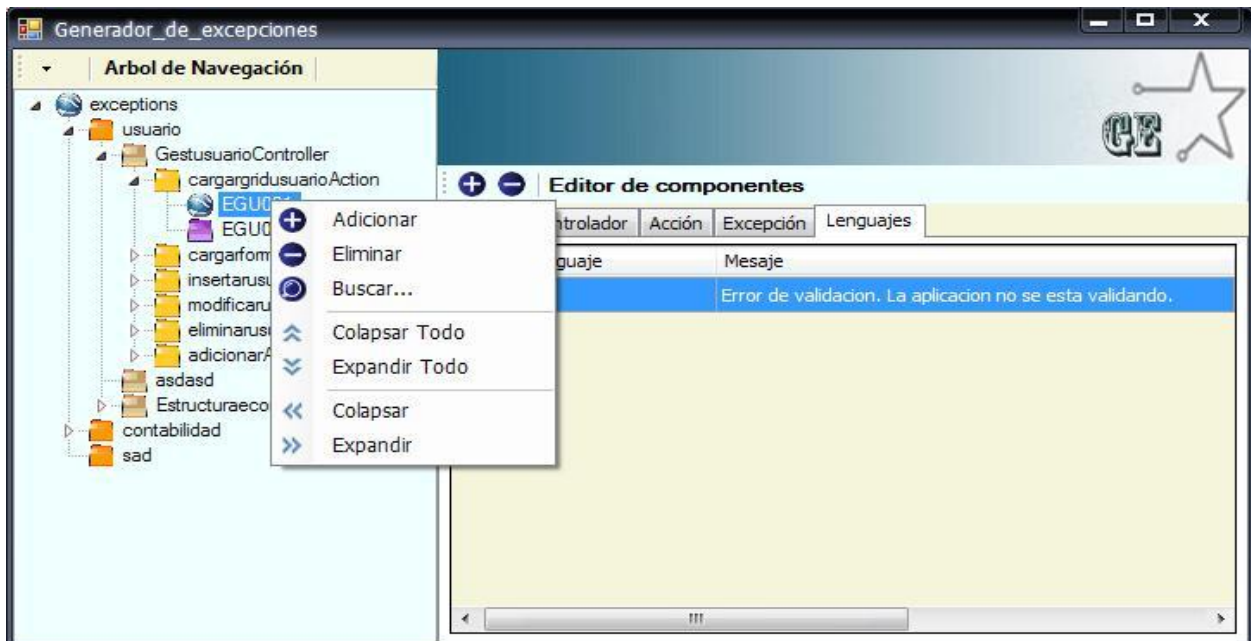


Figura 10: Interfaz que amerita al CRUD. del R1.1.1.2 Gestionar lenguajes.

Fuente: Vargas D. y Hernández M. 2009.

2. El componente Generador de aspectos brinda un interfaz que le permite al usuario gestionar los ficheros sobre los que realizará los cambios o modificaciones pertinentes, presentando funcionalidades como: Crear, Cargar o Abrir. Posibilitando a que se gestionen los distintos conceptos presentes en la interfaz de aspectos como: módulos, controladores, acciones, plantillas de aspectos y aspectos; todos ejecutándose en el orden establecido según la lógica de la configuración de los ficheros de excepciones. Dichas interfaces brindan la facilidad de realizar las funcionalidades de CRUD (**C**reate, **R**ead, **U**ppdate, **D**elte), tanto sobre el árbol de navegación, como en el panel de edición, estas características le permiten al usuario optar por algunas de las opciones para una mayor manejabilidad de las interfaces según sus necesidades. Así como también presenta opciones de observar la configuración previa de los ficheros XML que se estén gestionando sin tener que acceder al archivo de manera física, y de buscar si existe algunos de los conceptos para la utilidad necesaria que desee darle el mismo.

- Prototipo de interfaz del componente Generador de aspectos.

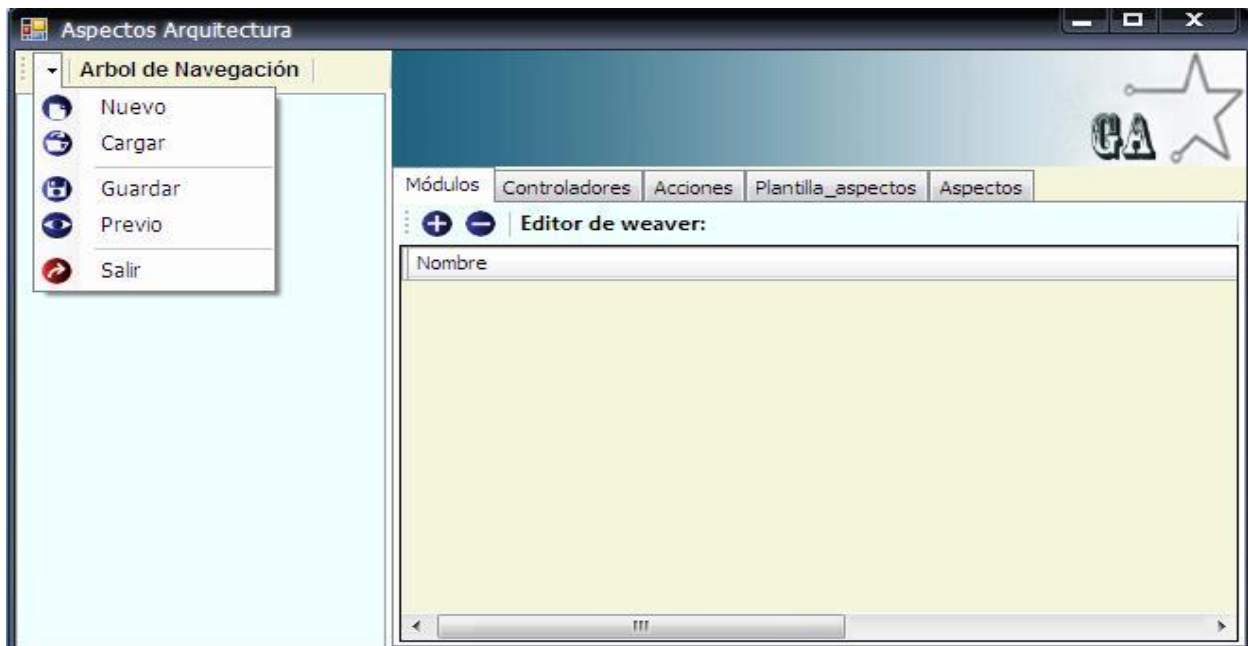


Figura 11: Interfaz que amerita al paquete de R.F. de la gestión de ficheros del componente Generador de aspectos.

Fuente: Vargas D. y Hernández M. 2009.

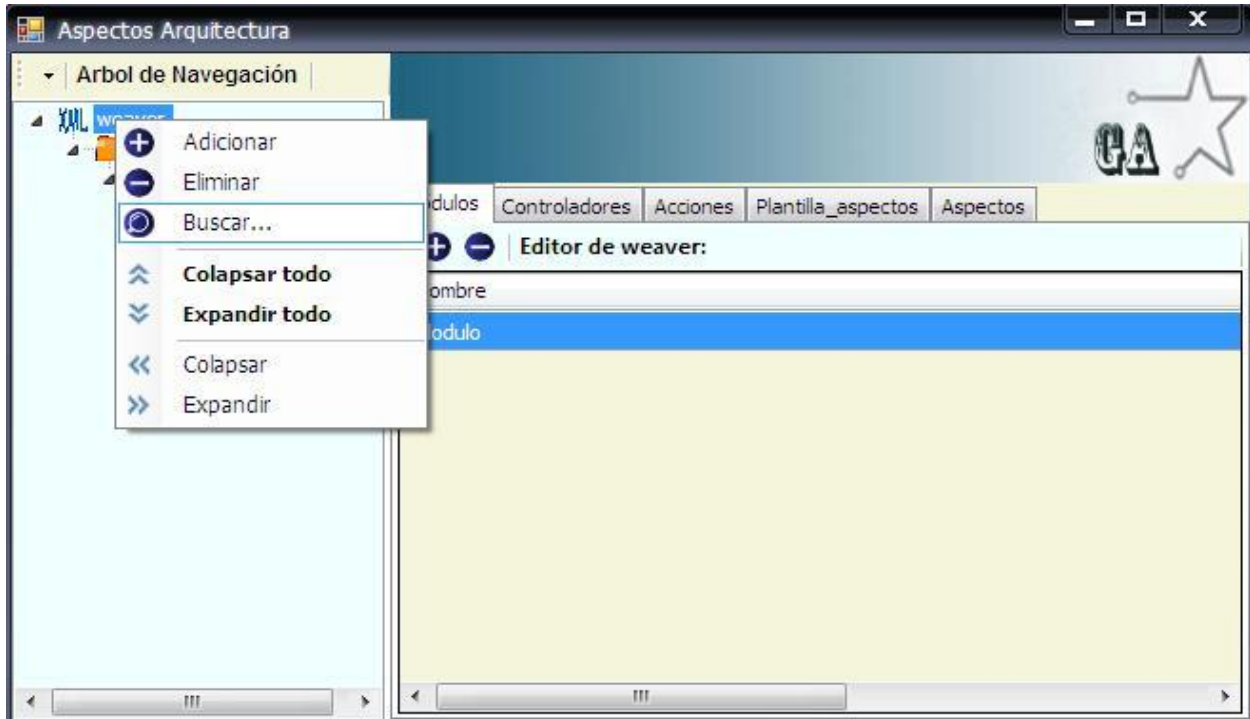


Figura 12: Interfaz que amerita al CRUD. del R2.1.3.1 Gestionar módulos.

Fuente: Vargas D. y Hernández M. 2009.

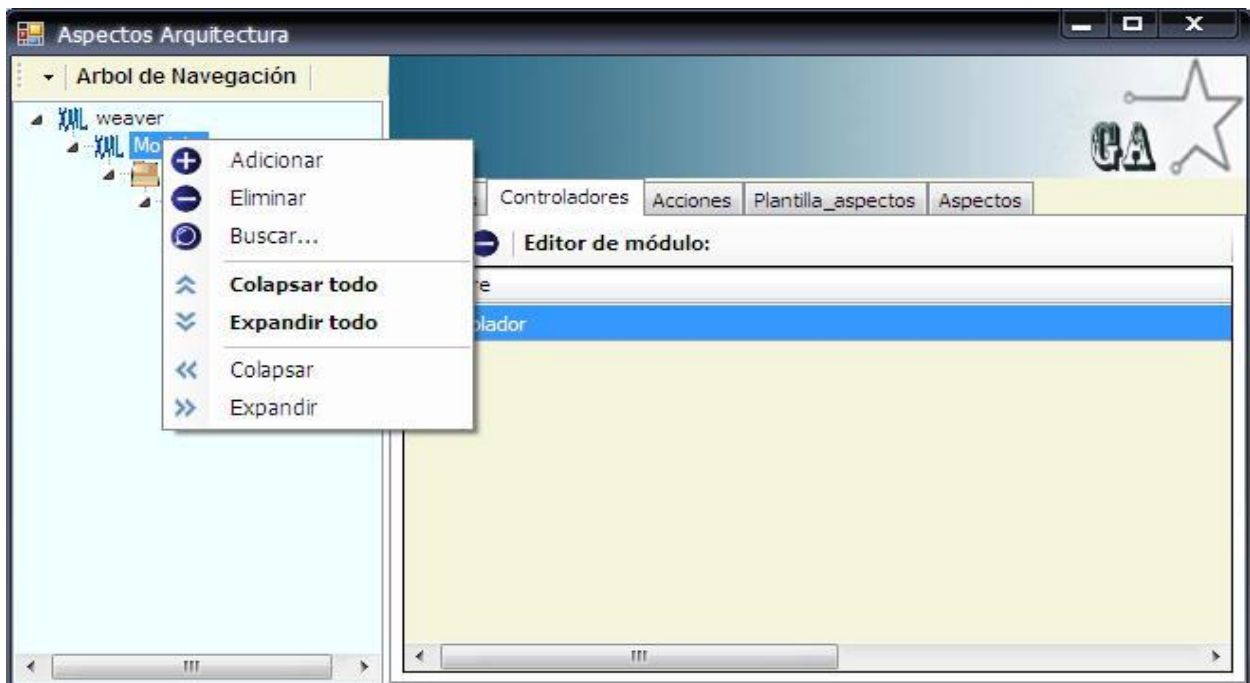


Figura 13: Interfaz que amerita al CRUD. del R2.1.3.2 Gestionar controladores.

Fuente: Vargas D. y Hernández M. 2009.

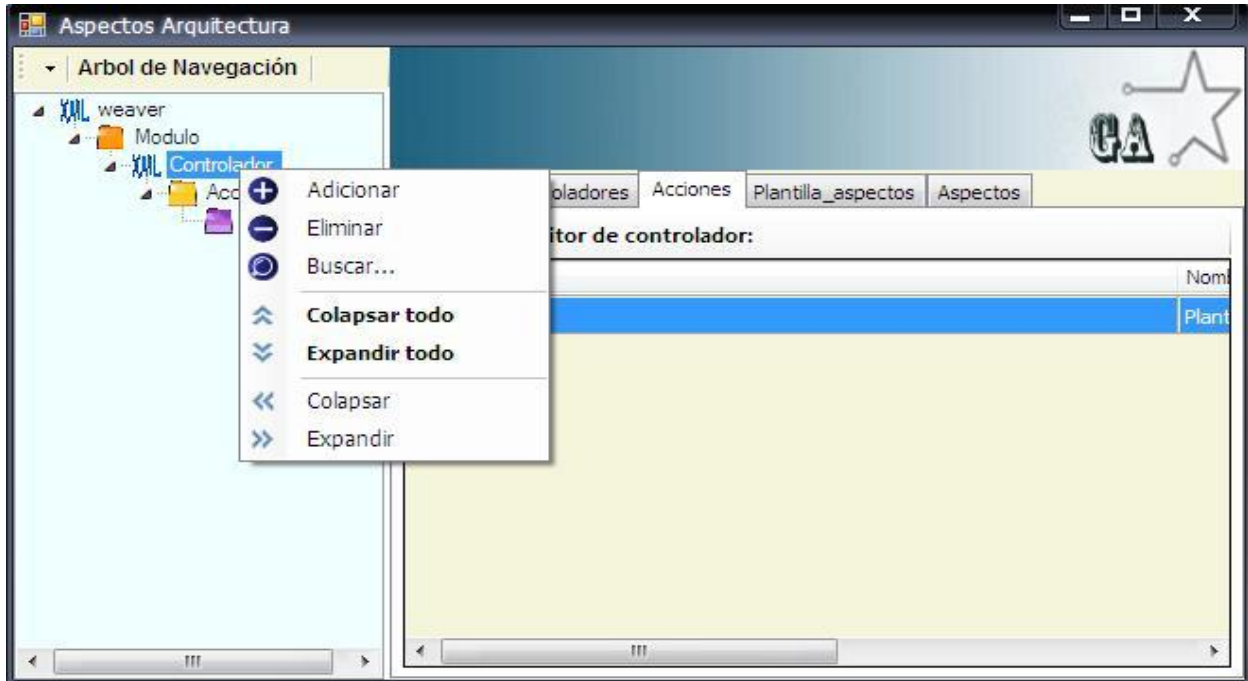


Figura 14: Interfaz que amerita al CRUD. del R2.1.3.3 Gestionar acciones.

Fuente: Vargas D. y Hernández M. 2009.

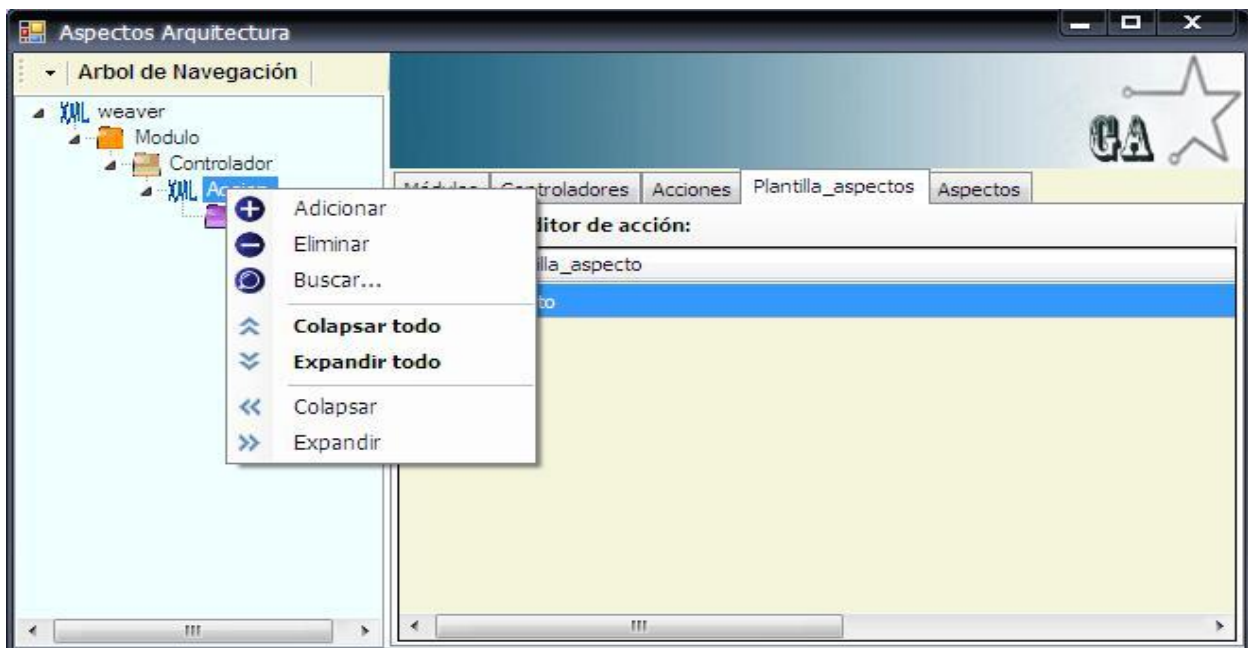


Figura 15: Interfaz que amerita al CRUD. del R2.1.2. Gestionar plantilla de aspectos.

Fuente: Vargas D. y Hernández M. 2009.

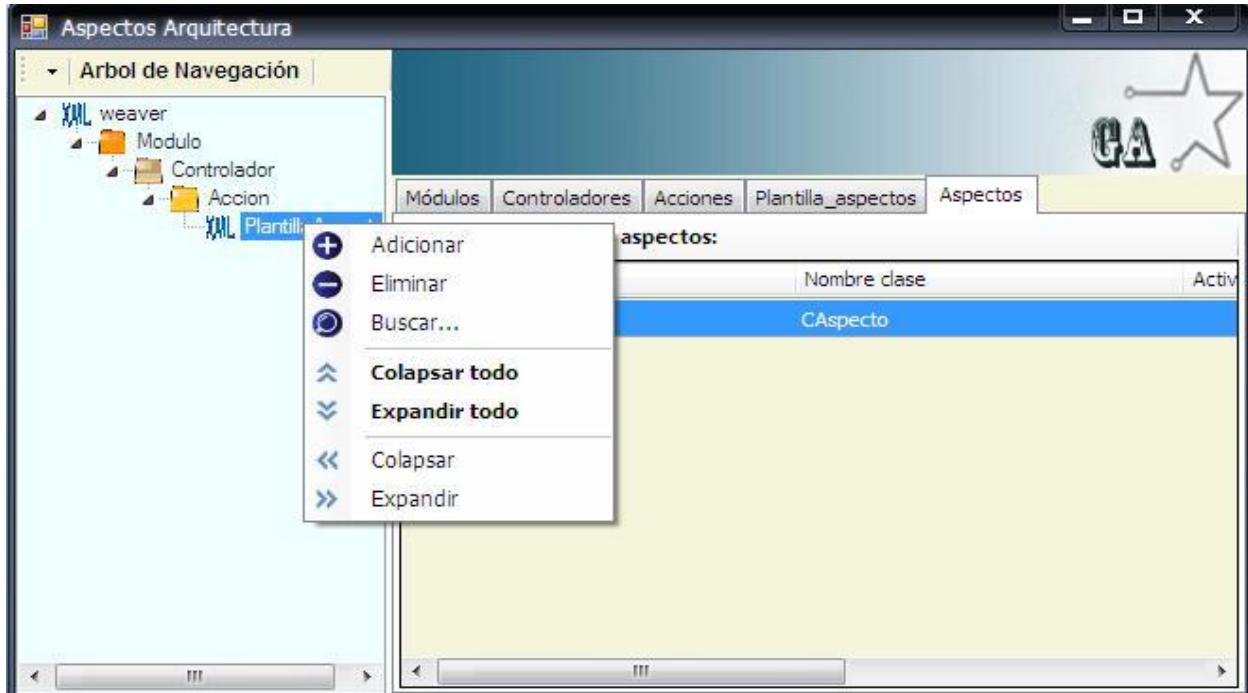


Figura 16: Interfaz que amerita al CRUD. del R2.1.1 Gestionar aspectos.

Fuente: Vargas D. y Hernández M. 2009.

El componente Auditor de código fuente primeramente brinda una interfaz que le permite al usuario la posibilidad de elegir el proyecto sobre el que va a trabajar, presentando funcionalidades como: Crear un nuevo proyecto en caso de que seleccione la opción de Instanciar o cargar un proyecto existente en caso de Auditar. Además en el caso de haber realizado la audición presenta una interfaz que le permite al usuario comprobar la integridad de los ficheros mediante una comparación. La representación de los prototipos de interfaz se pueden observar en (

3. Anexo 6: Prototipos de interfaz que ameritan a los requisitos funcionales del componente Auditor de código fuente.).
4. El componente Generador de inversión de control brinda una interfaz que le permite al usuario gestionar los ficheros sobre los que realizará los cambios o modificaciones

←

pertinentes, presentando funcionalidades como: Crear fichero nuevo o cargar fichero existente que le posibilite la gestión de los distintos conceptos presentes en la interfaz de inversión de control como: subsistema, contrato y parámetro; todos ejecutándose en el orden establecido según la lógica de la configuración de los ficheros de inversión de control. Dichas interfaces brindan la facilidad de realizar las funcionalidades de CRUD (**C**reate, **R**ead, **U**ppdate, **D**elete), en cada panel de edición existente por conceptos, estas características le permiten al usuario una mayor manejabilidad de las interfaces según sus necesidades. Así como también presenta opciones de observar la configuración previa de los ficheros XML que se estén gestionando sin tener que acceder al archivo de manera física, y de buscar si existe algunos de los conceptos para la utilidad necesaria que desee darle el mismo. La representación de los prototipos de interfaz se pueden observar en (Anexo 7: Prototipos de interfaz que ameritan a los requisitos funcionales del componente Generador de inversión de control.).

Diseño de clases

En el (Anexo 8: Diseño de clases del componente Generador de Aspectos.) se muestra el diagrama de clases perteneciente al componente Generador de Excepciones.

En el (Anexo 9: Diseño de clases del componente Generador de Excepciones. Anexo 9: Diseño de clases del componente Generador de Excepciones.) se muestra el diagrama de clases perteneciente al componente Generador de Aspecto.

En el (Anexo 10: Diseño de clases del componente Auditor de Código Fuente.) se muestra el diagrama de clases perteneciente al componente Auditor de Código Fuente.

En el (Anexo 11: Diseño de clases del componente Generador de Inversión de Control.) se muestra el diagrama de clases perteneciente al componente Generador de Inversión de Control.

En el siguiente diseño de clases se muestra la integración final de todos los componentes que tienen comprendido el componente principal (Ver. Figura 17).

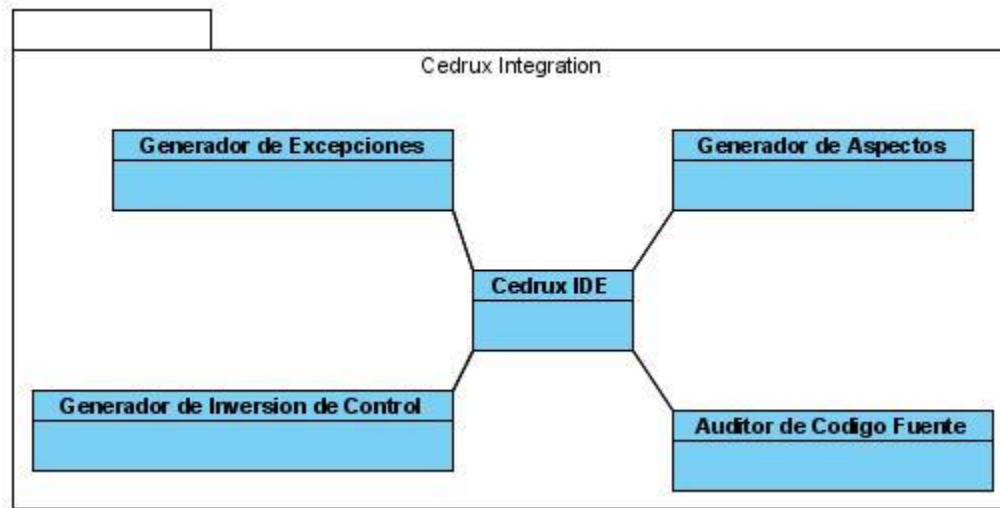


Figura 17: Diseño de clases de la integración del componente general.

Fuente: Vargas D. y Hernández M. 2009.

Artefactos de implementación

Los artefactos de implementación son generados una vez implementados los subsistemas o componentes que integran la solución propuesta como: La estructura de los ficheros XML mediante los cuales se realiza la configuración mediante la lectura y persistencia de los datos que introduce el usuario, también el diagrama de componentes que establece las relaciones entre los componentes y la estructura de los subsistemas que integran al sistema desarrollado, y el modelo de datos el que se genera mediante la funciones implícitas como: Instanciar y Auditar presentes en el subsistema Auditor de código fuente para la seguridad de los ficheros del Sistema Integral de Gestión de Entidades.

1. Esquemas de la configuración de los XML.

Los esquemas de configuración de los XML muestran cómo están estructurados según la estandarización de cada fichero para el marco de trabajo de CedruX. Los mismos permiten la

lectura y persistencia de datos logrando que se mantenga la estructura de los ficheros y la seguridad de los mismos, haciendo el proceso más dinámico.

- Esquema de los XML del Generador de aspectos arquitectónicos.

```
<?xml version="1.0" encoding="utf-8" ?>
- <aspects>
  - <Aspecto active="True">
    - <class name="CApecto" singleton="True">
      <method name="MApecto" static="True" />
    </class>
  </Aspecto>
</aspects>
```

Figura 18: Esquema del XML de aspectos (aspet.xml).

Fuente: Vargas D. y Hernández M. 2009.

```
<?xml version="1.0" encoding="utf-8" ?>
- <aspecttemplates>
  - <PlantillaAspectos>
    <aspect name="Aspecto" execution="Pre_Condicion" />
  </PlantillaAspectos>
</aspecttemplates>
```

Figura 19: Esquema del XML de la plantilla de aspectos (aspecttemplate.xml).

Fuente: Vargas D. y Hernández M. 2009.

```
<?xml version="1.0" encoding="utf-8" ?>
- <weaver>
  - <Modulo>
    - <Controlador>
      <Accion aspecttemplate="PlantillaAspectos" />
    </Controlador>
  </Modulo>
</weaver>
```

Figura 20: Esquema del XML tejedor (weaver.xml).

Fuente: Vargas D. y Hernández M. 2009.

- Esquema de los XML del Generador de Excepciones.

```
<?xml version="1.0" encoding="utf-8" ?>
- <excepciones>
  - <C0001>
    <tipo>B</tipo>
    - <es>
      <mensaje>Ha ocurrido un error grave en el sistema.</mensaje>
      <descripcion>Se ha producido un error en el sistema.</descripcion>
    </es>
  </C0001>
</excepciones>
```

Figura 21: Esquema del XML de excepción (exception.xml).

Fuente: Vargas D. y Hernández M. 2009.

```
<?xml version="1.0" encoding="utf-8" ?>
- <exceptions>
  - <Modulo>
    - <Controlador>
      - <Accion>
        <exception type="Excepcion" code="C0001" filter="FExcepcion" />
      </Accion>
    </Controlador>
  </Modulo>
</exceptions>
```

Figura 22: Esquema del XML manejador de excepciones (managerexception.xml).

Fuente: Vargas D. y Hernández M. 2009.

- Esquema del XML del Generador de inversión de control.

```
<?xml version="1.0" encoding="utf-8" ?>
- <IoC>
  - <ModuloX>
    - <ContratoUno>
      <inyector controlador="ClaseX" accion="RegistrarInformacion" />
    - <prototipo>
      <parametro nombre="Informacion" tipo="integer" />
      <resultado tipo="void" />
    </prototipo>
  </ContratoUno>
</ModuloX>
</IoC>
```

Figura 23: Esquema del XML de inversión de control (IoC.xml).

Fuente: Vargas D. y Hernández M. 2009.

2. Diagrama de componentes.

Los diagramas de componentes se representan como un grafo de componentes de software unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten. Son usados para estructurar el modelo de implementación en conjuntos como relaciones de componentes y subsistemas de implementación. Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, ya sean estos componentes de código fuente, binarios o ejecutables. (IBM Corporation, 2003)

Partiendo de la estructuración del diagrama de componentes se puede destacar la gestión de software, la reutilización, los requisitos relacionados con la facilidad de desarrollo y las restricciones según los lenguajes de programación y las herramientas utilizadas durante el proceso de desarrollo.

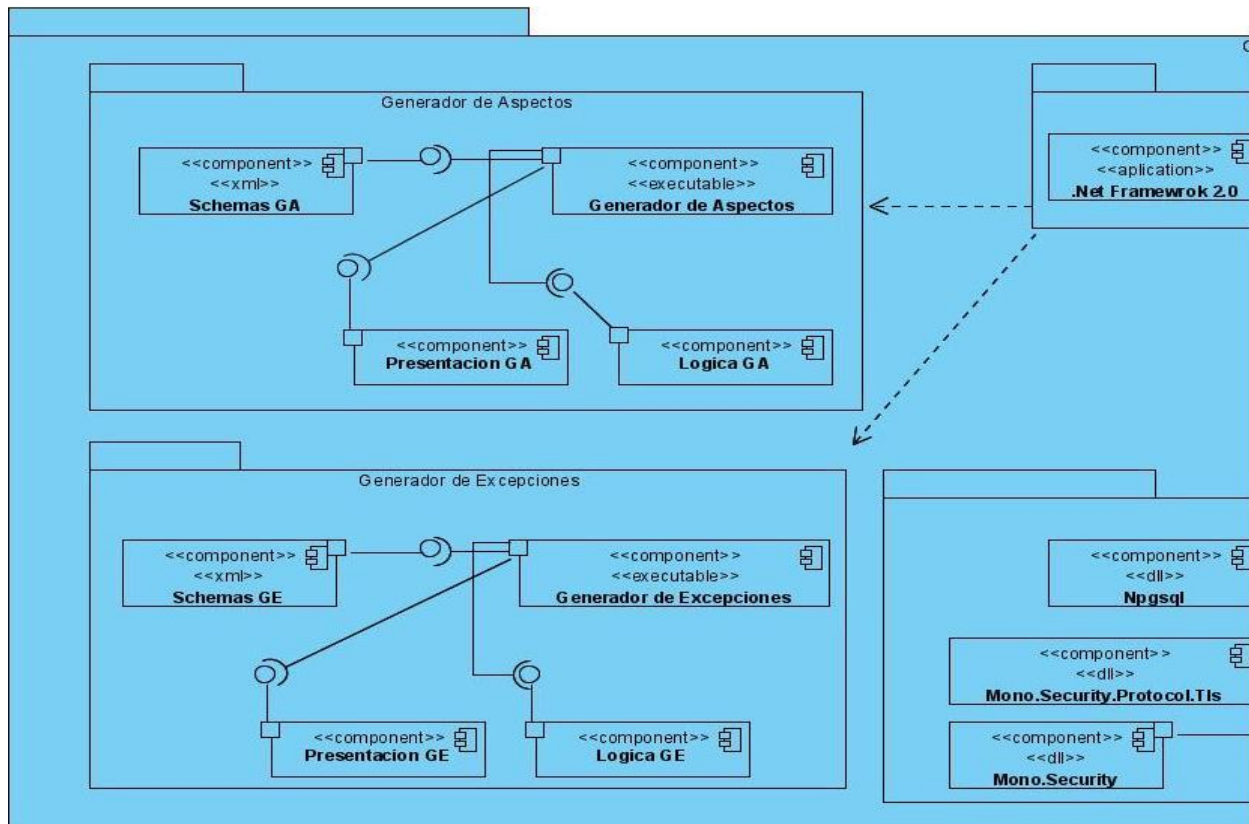


Figura 24: Diagrama de componentes del sistema (parte I).

Fuente: Vargas D. y Hernández M. 2009.

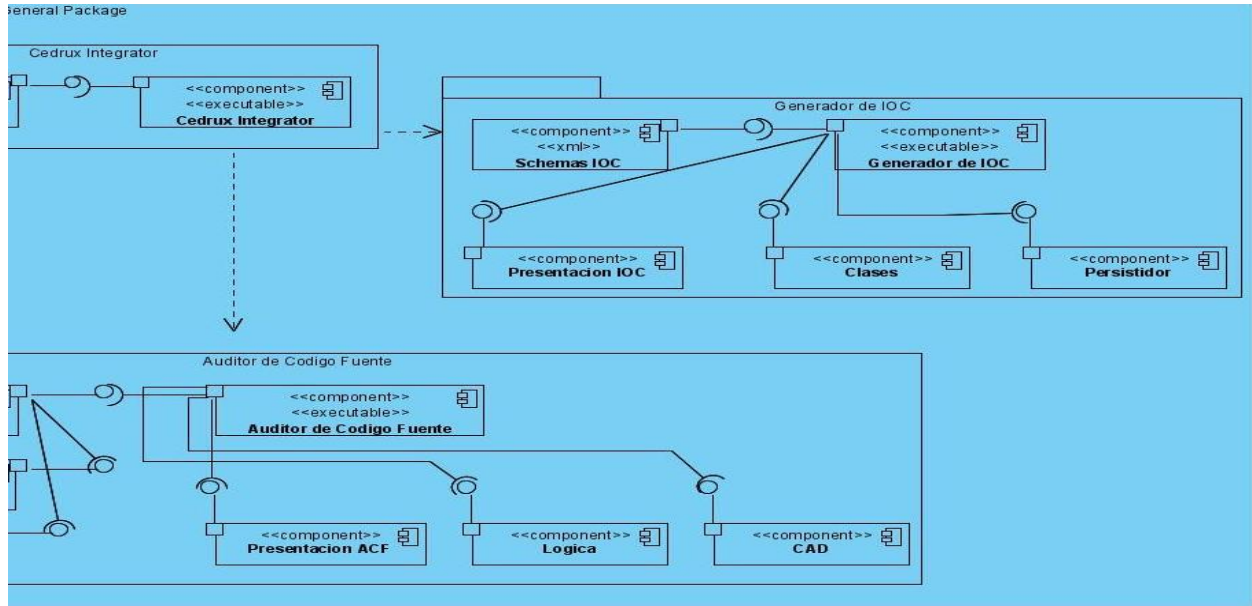
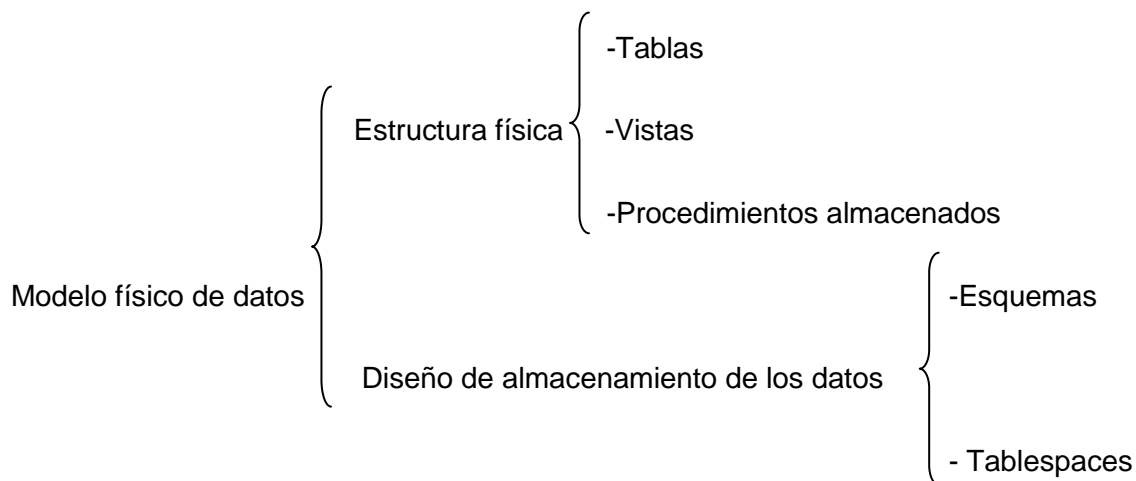


Figura 25: Diagrama de componentes del sistema (parte II).

Fuente: Vargas D. y Hernández M. 2009.

3. Modelo de Datos.

Según RUP el modelo de datos permite proveer de una vista de las entidades lógicas de datos y sus relaciones con independencia de la plataforma de base de datos a utilizar. Durante el proceso de construcción de la misma pasa por distintas formas hasta llegar a la 3 forma normal (IBM Corporation, 2003), donde la cual queda constituida de la siguiente manera:



El modelo de datos ha sido diseñado manteniendo la estructura y el estandarizado trazado por parte del proyecto Cedrux. Partiendo desde el nombre del esquema, de las tablas, las funciones y las secuencias hasta los tipo de datos de cada campo dentro de las tablas. Manteniendo el concepto de los patrones de Alta Cohesión y Bajo Acoplamiento entre las clases.

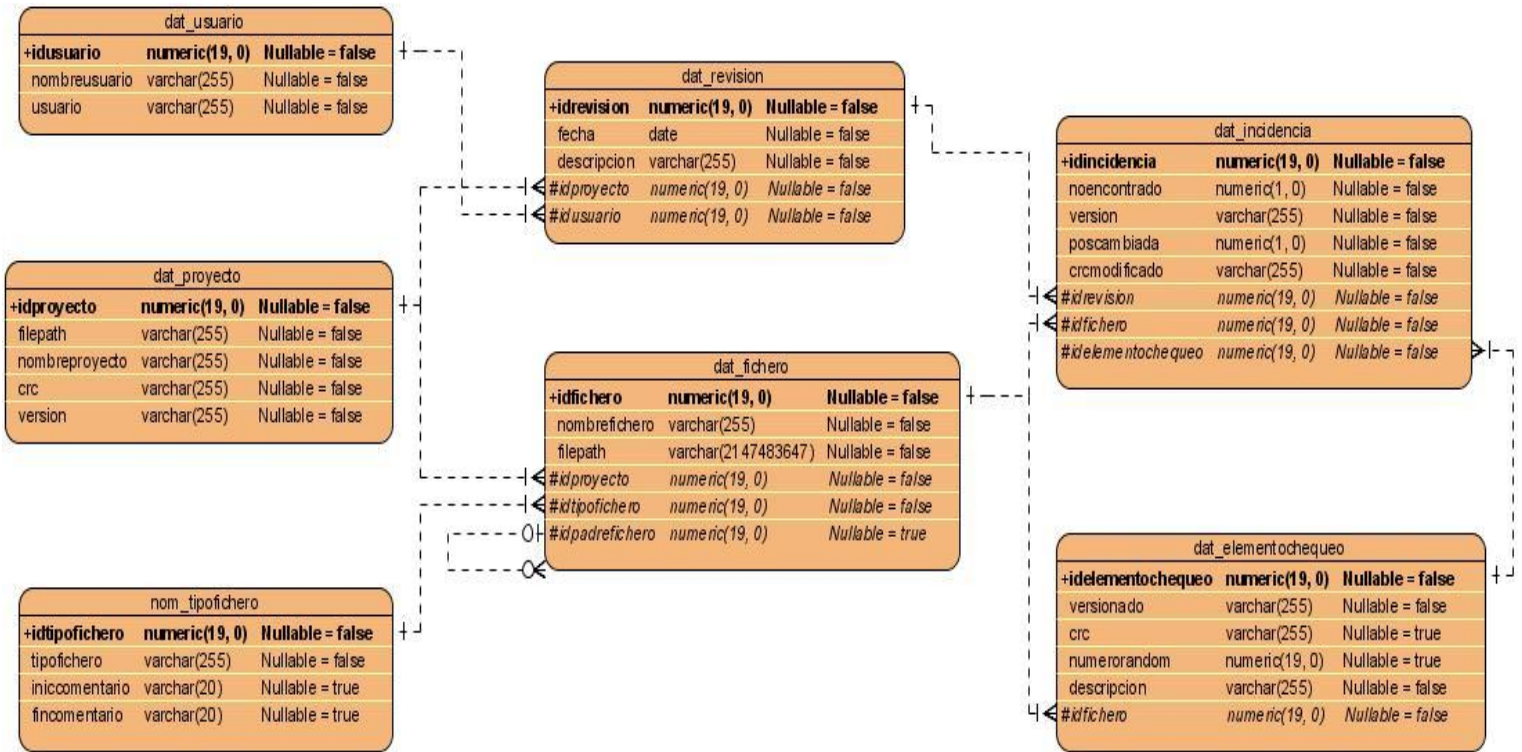


Figura 26: Modelo de datos del componente Auditor de Código Fuente.

Fuente: Vargas D. y Hernández M. 2009.

Casos de pruebas

Los casos de pruebas o Test Case son condiciones o variables para determinar si los requisitos de una aplicación son parciales o satisfactorios. Se pueden realizar muchos casos de pruebas para determinar si un requisito es satisfactorio o no. Para esto se puede realizar un caso de pruebas como mínimo por requisitos a no ser que dicho requisito presente requisitos secundarios, en ese caso cada requisito secundario deberá tener como mínimo un caso de

← →
prueba para evaluar sus funcionalidades. (Wikipedia Foundation .Inc, 2009) Algunas metodologías como RUP recomiendan realizar dos casos de pruebas como mínimo por requisitos, donde una realizaría las pruebas positivas y otras las negativas. (IBM Corporation, 2003)

Si la aplicación es creada sin requisitos formales, entonces los casos de pruebas se escriben basados en la operación normal de programas de una clase similar. Lo que caracteriza un escrito formal de caso de prueba es que haya una *entrada conocida* y una *salida esperada*, los cuales son formulados antes de que se ejecute la prueba. La *entrada conocida* debe probar una precondition y la *salida esperada* debe probar una post - condición. (Wikipedia Foundation .Inc, 2009)

Características de los casos de pruebas dirigidos a escenarios de casos de uso de un sistema.

Los requerimientos son la fuente principal para obtener los casos de prueba pero no son el único medio. Muchas veces son insuficientes para proporcionar una base completa que permita desarrollar las pruebas. Por lo que es necesario considerar otros elementos como riesgos, restricciones, tecnologías, cambios, fallos, etc. (Moreno, 2006)

Por lo normal los casos de pruebas se derivan de los casos de uso del modelo de casos de uso o del modelo de casos del diseño, con los que se validan los requisitos del sistema. Algunos de los casos de pruebas más comunes son:

1. **Los Casos de Pruebas que especifican cómo probar un caso de uso o un escenario específico de un caso de uso:** Efectúan la verificación del resultado de la interacción entre los actores y el sistema, la satisfacción de las precondiciones y pos condiciones especificadas en el caso de uso y que sigue la secuencia de acciones detalladas en el mismo. Un Caso de Prueba basado en un caso de uso es una prueba del sistema como “caja negra”, es decir, una prueba del comportamiento externo del sistema. (Moreno, 2006)

2. **Los Casos de Prueba que especifican cómo probar una realización de caso de uso del diseño o un escenario específico de la realización.** Efectúa la verificación de la interacción entre los componentes que implementan dicho caso de uso. Los Casos de

←

Pruebas basadas en una realización de un caso de uso son pruebas del sistema como “caja blanca”, es decir, una prueba de la interacción interna entre los componentes del sistema. (Moreno, 2006)

Casos de prueba realizados a la solución propuesta.

Se puede confirmar que la realización de casos de pruebas tiene como objetivo enmarcar todas las funcionalidades que brinda el sistema y comprobar que las mismas cumplen con los requisitos funcionales trazados durante la fase de desarrollo del componente. Además permiten valorar el correcto funcionamiento de cada escenario con sus características particulares. También sirve como guía al usuario para saber qué acción o respuesta corresponde luego de realizar alguna operación dentro de la aplicación.

Los casos de pruebas realizados a la solución propuesta fueron aquellos que **especifican cómo probar un caso de uso o un escenario específico de un caso de uso**. La descripción de los mismos se realizó mediante escenarios.

A continuación se muestra una breve descripción del componente Generador de Aspectos enfocándose en los escenarios: Crear Ficheros, Cargar Ficheros, Guardar Ficheros y Buscar, los mismos se pueden apreciar en (Tabla 10), las demás descripciones correspondientes a dicho componente se podrán encontrar en el (Anexo 12: Diseño de casos de pruebas del componente Generador de Aspectos.), al igual que la del componente Generador de Excepciones en (Anexo 13: Diseño de casos de pruebas del componente Generador de Excepciones.), así como también los que atribuyen al componente Generador de Inversión de Control en (Anexo 14: Diseño de casos de pruebas del Generador de Inversión de Control.) y al Componente Auditor de Código Fuente en (Anexo 15: Diseño de casos de pruebas del componente Auditor de Código Fuente.).

- Generador de Aspectos.

Tabla 10: Paquete de DCP del escenario Gestionar ficheros perteneciente al componente generador de aspectos.

DCP Crear Ficheros.

1. Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el componente IDE, en Generador de aspectos.
- El fichero que desea crear debe ubicarlo en una carpeta que no tenga implícito ningún fichero del mismo tipo.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Crear ficheros.	Se crean nuevos ficheros al sistema.	EP 1.1: Crear ficheros al sistema.	- Se presiona el botón superior izquierdo del menú. - Se elije la opción Crear del menú. - Se elije la ubicación donde se van a crear los ficheros. - Se presiona el botón Aceptar .
		EP 1.2: Crear ficheros dejando campo requeridos en blanco.	NA.
		EP 1.3: Crear ficheros introduciendo errores en los datos.	NA.
		EP 1.4: Cancelar.	- Se presiona el botón superior izquierdo del menú. - Se elije la opción Crear del menú. - Se elije o no la ubicación donde se van a crear los ficheros. - Se presiona el botón Cancelar .

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Crear ficheros al sistema.	NA.	Para la opción de Crear : Se crea o busca la ubicación donde se van a crear los ficheros, se crean y se cierra la ventana.	
EP 1.2	Crear ficheros dejando campo requeridos en blanco.	NA.	NA.	
EP 1.3	Crear ficheros introduciendo errores en los datos.	NA.	NA.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Cargar Ficheros.

1. Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el componente IDE, en Generador de aspectos.
- El fichero que desea cargar debe existir en la carpeta de la ubicación elegida.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Cargar ficheros.	Se cargan ficheros existentes al sistema.	EP 1.1: Cargar ficheros al sistema.	<ul style="list-style-type: none"> - Se presiona el botón superior izquierdo del menú. - Se elije la opción Cargar del menú. - Se elije la ubicación donde están ubicados los ficheros existentes. - Se presiona el botón Aceptar.
		EP 1.2: Cargar ficheros dejando campo requeridos en blanco.	NA.
		EP 1.3: Cargar ficheros introduciendo errores en los datos.	NA.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se presiona el botón superior izquierdo del menú. - Se elije la opción Cargar del menú. - Se elije o no la ubicación de donde se va a cargar los ficheros. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Cargar ficheros existentes al sistema.	NA.	Para la opción de Cargar : Se busca la ubicación donde se encuentran los ficheros	

			existentes, se cargar y se cierra la ventana.	
EP 1.2	Cargar ficheros dejando campo requeridos en blanco.	NA.	NA.	
EP 1.3	Cargar ficheros introduciendo errores en los datos.	NA.	NA.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Guardar Ficheros.

1. Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el componente IDE, en Generador de aspectos.
- Se debe de haber cargado o creado los ficheros.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Guardar ficheros.	Se guarda los ficheros del sistema.	EP 1.1: Guardar ficheros del sistema.	- Se presiona el botón superior izquierdo del menú. - Se elije la opción Guardar del menú. - Se elije la ubicación donde serán ubicados los ficheros. - Se presiona el botón Aceptar .
		EP 1.2: Guardar ficheros dejando campo requeridos en blanco.	NA.
		EP 1.3: Cargar ficheros introduciendo errores en los datos.	NA.

		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se presiona el botón superior izquierdo del menú. - Se elije la opción Guardar del menú. - Se elije o no la ubicación de donde se va a guardar los ficheros. - Se presiona el botón Cancelar.
--	--	-------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Guardar ficheros existentes al sistema.	NA.	Para la opción de Guardar : Se busca la ubicación donde se guardarán los ficheros, se guarda y se cierra la ventana.	
EP 1.2	Guardar ficheros dejando campo requeridos en blanco.	NA.	NA.	
EP 1.3	Guardar ficheros introduciendo errores en los datos.	NA.	NA.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Buscar.

1. Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el componente IDE, en Generador de aspectos.
- Se debe de haber cargado o creado los ficheros.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Buscar.	Se busca algún concepto (Módulo, Controlador, Acción, Plantilla de aspectos).	EP 1.1: Buscar conceptos.	- Se presiona clic derecho sobre el árbol de navegación. - Se elije la opción Buscar del menú. - Se elije el concepto a buscar. - Se presiona el botón Buscar .
		EP 1.2: Buscar conceptos dejando campo requeridos en blanco.	- Se elije la opción Buscar del menú - Se introduce los datos en el formulario de buscar dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra la ventana que muestra el mensaje de información.
		EP 1.3: Cargar ficheros introduciendo errores en los datos.	- Se elije la opción Buscar del menú - Se introduce los datos en el formulario de buscar introduciendo datos con errores. - Se presiona el botón Aceptar o se cierra la ventana que muestra el mensaje de información.
		EP 1.4: Cancelar.	- Se elije la opción Buscar del menú - Se introducen o no los datos en el formulario. - Se cierra el formulario.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Buscar conceptos.	V(Módulo, Controlador, Acción, Plantilla de aspectos)	Para el botón de Buscar : Se busca el concepto y se muestra la ubicación del mismo en el árbol de navegación conjuntamente con sus nodos hijos.	
EP 1.2	Buscar conceptos dejando campo requeridos en blanco.	I(Vacío)	Para el botón de Buscar : Se deja en blanco el campo de texto y sale un cartel que indica que los datos son incorrectos.	
EP 1.3	Buscar conceptos introduciendo errores en los datos.	I (/ *- / - ' .)	Para el botón de Buscar : Se deja en blanco el campo de texto y sale un cartel que indica que los datos son incorrectos.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Después de haber realizado los casos de pruebas a los distintos escenarios existentes para cada componente que integran la solución propuesta se pudo comprobar que durante un ciclo de 3 iteraciones durante todo el proceso de prueba en las distintas líneas y modulo del proyecto

ERP – Cuba, se encontraron un promedio de 3 errores por cada componente. Los mismos fueron eliminados durante todo el proceso de pruebas cubriendo los requisitos funcionales levantados durante el proceso de análisis. Cada uno de los componentes fue liberado con un alto grado de aceptación por parte de la línea de calidad.

Conclusiones

Durante el transcurso del capítulo se argumentaron los fundamentales aspectos que se llevan a cabo durante el proceso de análisis, comenzando por el levantamiento de los requisitos funcionales que contienen cada uno de los subsistemas integrados en el componente principal y de los requisitos no funcionales para garantizar una ejecución eficaz de la aplicación. También se reflejaron los distintos prototipos de interfaz que se relacionan con los requisitos levantados, se plasmó el diseño de clases para una mejor comprensión de cómo están estructuradas las clases que conforman toda la capa del negocio de los componentes. Se expusieron además los distintos artefactos de implementación generados: Esquemas de configuración de los archivos XML, diagrama de componentes y el modelo de datos, y se describieron los casos de pruebas de todos los componentes para comprobar si las funcionalidades implementadas en cada uno cumple con los requisitos argumentados en el proceso de desarrollo.

Capítulo 3: Evaluación de la solución.

Introducción.

En el presente capítulo se muestran algunas métricas que se aplican en la actualidad para validar la calidad en el diseño de software y se definen cuáles se aplicaron al diseño de cada uno de los componentes que integran la solución propuesta. Las mismas ayudan a comprender todo el proceso técnico que se utiliza para desarrollar un producto, así como el propio producto. También brinda una valoración de la solución del componente y la matriz de inferencia de indicadores de calidad.

Valoración de la solución.

Confirmación por inspección y provisión de evidencia objetiva de que los requerimientos particulares para un uso específico son alcanzados. En diseño y desarrollo, la validación está relacionada con el proceso de reexaminación de un producto para determinar la conformidad con las necesidades del usuario. La validación es realizada normalmente sobre el producto final bajo condiciones operacionales definidas (Oficina Nacional del Gobierno Electrónico e Informática - ONGEI., 2004). La valoración de la solución es el paso final en el proceso de evaluación del software. Donde una solución puede ser valorada de diversas maneras, debido a que existen distintos tipos de valoraciones como son:

- Valoración cualitativa.
- Valoración indirecta.
- Valoración directa.

Una valoración se realiza mediante una métrica para asignar uno de los valores de una escala (el mismo puede ser número o categoría) al atributo de la entidad (sistema, subsistemas o componentes).



Valoración cualitativa: Es una evaluación sistemática del grado o capacidad de una entidad para satisfacer necesidades o requerimientos específicos. Además se emplean categorías, como algunos de los atributos más importantes de una entidad, ejemplo: el lenguaje de desarrollo del programa (C, C ++, C #, PHP, JAVA).

Valoración indirecta: Es la valoración de un atributo derivada del valor de uno o más atributos diferentes. Es la valoración externa de un atributo de un sistema, ejemplo: el tiempo de respuesta a la información alimentada por el usuario, es una valoración indirecta de los atributos del software, debido a que esta medida se verá influenciada por los atributos externos del sistema, así como los propios internos.

Valoración directa: Es una valoración del producto, de forma indirecta o directa. Ejemplo: El número de líneas de código, las valoraciones de la complejidad, el número de fallas encontradas durante el proceso y el índice de señales o alertas, son todas las valoraciones internas propias del producto en sí. (Oficina Nacional del Gobierno Electrónico e Informática - ONGEI., 2004).

1.1 Valoraciones aplicadas a la solución propuesta.

Dentro de los distintos tipos de validación del software existentes se escogieron para la solución propuesta las de tipo cualitativa y directa.

Debido a las características que debería representar el sistema se realizó sobre la plataforma de desarrollo .NET de Microsoft, por la gama de facilidades que brinda para el desarrollo de componentes de software, posibilitando para el mismo la interacción con distintos lenguajes y compiladores, así como un rápido desarrollo de aplicaciones (una de las fundamentales necesidades del proyecto). Se empleó como gestor de base de datos la herramienta PostgreSQL debido a que es una herramienta con características importantes como: presenta interfaz amigable y ágil de manipular, permite una conexión segura con la plataforma .NET y fundamentalmente es una herramienta de software libre. El sistema contiene 93826 líneas de código, es totalmente seguro y eficaz debido a que los ficheros no pueden ser configurables, presenta una interfaz amigable para el usuario y fácil de manipular, así como una conexión segura con la base de datos y con los ficheros a gestionar, al igual que con el estandarizado de los mismos.



Métricas que evalúan el diseño, análisis de los resultados.

Cuando se planifica un proyecto se tiene que obtener estimaciones del costo y esfuerzo humano requerido por medio de las mediciones de software que se utilizan para recolectar los datos cualitativos acerca del software y sus procesos para aumentar su calidad. Dichas mediciones se realizan mediante un proceso conocido como Métricas, que no es más que un instrumento que cuantifica un criterio.

2.1 Métricas de software.

Las métricas del software no mantienen un tamaño estandarizado o estructurado, simplemente el proyecto debe elegir las métricas que cumplen con las características o necesidades. Las mismas se basan en cuantificar la complejidad, funcionalidad y eficiencia relacionada con el desarrollo del software, enfatizando sus objetivos en:

- Comprender mejor la calidad del producto.
- Estimar la efectividad del proceso.
- Mejorar la calidad del trabajo realizado en el nivel del proyecto.

Las métricas del software se pueden catalogar como:

- **Métricas técnicas:** Que se centran en las características de software por ejemplo: la complejidad lógica y el grado de modularidad.
- **Métricas de calidad:** proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente.
- **Métricas de productividad.** Se centran en el rendimiento del proceso de la ingeniería del software.
- **Métricas orientadas a la función** y las **Métricas orientadas al tamaño.**

2.2 Métricas orientadas a objetos.

Las métricas orientadas a objetos se crearon para resaltar lo esencial para el ingeniero de software que es la calidad del producto. Donde el enfoque fundamental de las métricas orientadas a objetos está en las clases, pieza fundamental para la construcción del software.

El Software Orientado a Objetos (OO) es fundamentalmente distinto del software que se desarrolla utilizando métodos convencionales. Las métricas para sistemas OO deben de ajustarse a las características que distinguen el software OO del software convencional. Estas métricas hacen hincapié en el encapsulamiento, la herencia, complejidad de clases y polimorfismo. Por lo tanto las métricas OO se centran en métricas que se pueden aplicar a las características de encapsulamiento, ocultamiento de información, herencia y técnicas de abstracción de objetos que hagan única a esa clase. Como en todas las métricas los objetivos principales de las métricas OO se derivan del software convencional: comprender mejor la calidad del producto, estimar la efectividad del proceso y mejorar la calidad del trabajo realizado a nivel del proyecto (ARREGUI, Septiembre 2005.).

Las medidas y las métricas son componentes claves de cualquier disciplina de la ingeniería; la ingeniería de software orientada a objetos no es una excepción. Lamentablemente, la utilización de métricas para sistemas orientados a objetos ha progresado con mucha más lentitud que la utilización de los demás métodos OO. Sin embargo, a medida que los sistemas OO van siendo más habituales, resulta fundamental que los ingenieros del software dispongan de mecanismos cuantitativos para estimar la calidad de los diseños y la efectividad de los programas OO (ARREGUI, Septiembre 2005.).

Características de las métricas orientadas a objetos.

El software orientado a objetos es esencialmente distinto del software que se desarrolla utilizando métodos convencionales. Por esta razón, las métricas para sistemas OO deben de adecuarse a las características que distinguen al software OO del software convencional (ARREGUI, Septiembre 2005.). Donde Berard definió cinco características que cumplen con dicho planteamiento: localización, encapsulamiento, ocultamiento de información, herencia y técnicas de abstracción de objetos.

Localización: Es una característica del software que indica la forma en que se concentra la información dentro de un programa.

Encapsulamiento: Es el empaquetamiento (o enlazado) de una colección de elementos. Entre los ejemplos de encapsulamiento de bajo nivel (software convencional) se cuentan los registros y matrices, y los subprogramas (procedimientos, funciones, subrutinas y párrafos) son mecanismos de nivel medio.

Ocultamiento de información: El ocultamiento de información suprime los detalles operativos de un componente de un programa. Tan solo se proporciona la información necesaria para acceder a ese componente o a aquellos otros componentes que deseen acceder a él.

Herencia: La herencia es un mecanismo que hace posible que los compromisos de un objeto se difundan a otros objetos. La herencia se produce a lo largo de todos los niveles de la jerarquía de clases

Abstracción: La abstracción es un mecanismo que permite al diseñador centrarse en los detalles esenciales de algún componente de un programa (tanto si es un dato como si es un proceso) sin preocuparse por los detalles de nivel inferior.

2.3 Métricas orientadas a clases.

La clase es la unidad principal de todo sistema OO. Entonces las medidas y métricas para una clase individual, la jerarquía de clases, y las colaboraciones de clases resultarán sumamente valiosas para un ingeniero de software que tenga que estimar la calidad de un diseño.

La clase encapsula a las operaciones (procesamiento) y a los atributos (datos), colabora con otras clases. Siendo todas estas características el factor fundamental de las métricas de CK.

Conjunto de métricas más aplicadas en la actualidad por los desarrolladores de software, propuestas por: Chidamber y Kemener, para un total de seis métricas y conocidas por: “La serie de métricas CK”:

- Métodos ponderados por clase (MPC).
- Árbol de profundidad de herencia (APH).
- Número de descendiente (NDD).

- Acoplamiento entre clases objeto (ACO).
- Respuesta para una clase (RPC).
- Carencia de cohesión en los métodos (CCM).

Otra de las propuestas de métricas más aplicadas son las métricas propuestas por Lorenz y Kidd, separándolas en cuatro amplias categorías:

- Tamaño
- Herencia
- Valores internos
- Valores externos

2.4 Métricas aplicadas a la solución propuesta.

Las métricas escogidas para medir la calidad de los componentes que integran la solución propuesta fueron decisión del proyecto, debido a que las mismas cubren los atributos de calidad de responsabilidad, complejidad del diseño, complejidad de implementación, reutilización, acoplamiento, complejidad del mantenimiento, cantidad de pruebas, nivel de cohesión y abstracción del diseño.

Dentro de las utilizadas se escogieron de la serie de métricas CK:

- Árbol de profundidad de herencia (APH): Está dado por la profundidad en la herencia de las clases heredadas de un nodo padre.
- Número de descendiente (NDD): Está dado por la cantidad de clases que heredan de un padre.
- Respuesta para una clase (RPC): Está dado por el número de relaciones de uso de una clase con otras.

Además se utilizaron dos métricas de la serie definida por Lorenz y Kidd:

- Tamaño de Clases (TC): Esta dado por el número de métodos asignados a una clase.
- Número de Operaciones Redefinidas para una Sub-Clase (NOR): Está dado por la cantidad de operaciones redefinidas en cada clase hija.



Atributos de calidad que se abarcan:

1. Responsabilidad. Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
2. Complejidad del diseño. Consiste en la complejidad que posee una estructura de diseño de clases.
3. Complejidad de implementación. Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.
4. Reutilización. Consiste en el grado de reutilización que esté presente en una clase o estructura de clase, dentro de un diseño de software.
5. Acoplamiento. Consiste en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, está muy ligada a la característica de Reutilización.
6. Complejidad del mantenimiento. Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costes y la planificación del proyecto.
7. Cantidad de pruebas. Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (Unidad) del producto (Componente, modulo, clase, conjunto de clases, etc.) diseñado.
8. Nivel de Cohesión. Consiste en el grado de especialización de las clases concebidas para modelar un dominio o concepto específico.
9. Abstracción del diseño. Consiste en la capacidad de modelar lo más cercano posible a la realidad un concepto o dominio determinado. (Lazo & Yzquierdo, 2007)

Las métricas concebidas como instrumento para evaluar la calidad del diseño de los componentes Generador de Aspectos, de Excepciones, de Inversión de Control y el Auditor de Código Fuente y su relación con los atributos de calidad definidos en este trabajo son las siguientes:



➤ **Tamaño de Clases (TC):**

Tabla 11: Tamaño de Clases (TC).

Atributo que afecta	Modo en que lo afecta
Responsabilidad	Un aumento del TC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	Un aumento del TC implica un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento del TC implica una disminución en el grado de reutilización de la clase.

Fuente: Vargas D. y Hernández M. 2009.

➤ **Respuesta para una clase (RPC):**

Tabla 12: Respuesta para una clase (RPC).

Atributo que afecta	Modo en que lo afecta
Acoplamiento	Un aumento del RPC implica un aumento del acoplamiento de la clase.
Complejidad de mantenimiento	Un aumento del RPC implica un aumento de la complejidad del mantenimiento de la clase.
Reutilización	Un aumento del RPC implica una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	Un aumento del RPC implica un aumento de la cantidad de pruebas de unidad necesarias para probar una clase.

Fuente: Vargas D. y Hernández M. 2009.

➤ **Árbol de profundidad de herencia (APH):**

Tabla 13: Árbol de profundidad de herencia (APH).

Atributo que afecta	Modo en que lo afecta
Complejidad de mantenimiento	Un aumento del APH implica una disminución de la complejidad del mantenimiento de la clase.
Complejidad de diseño	Un aumento del APH implica una disminución de la complejidad del diseño de la clase.
Reutilización	Un aumento del APH implica un aumento en el grado de reutilización de la clase.



Fuente: Vargas D. y Hernández M. 2009.

➤ **Número de descendiente (NDD):**

Tabla 14: Número de descendientes (NDD).

Atributo que afecta	Modo en que lo afecta
Abstracción del diseño	Un aumento del NDD puede diluir la abstracción del diseño de clases.
Cantidad de pruebas	Un aumento del NDD implica un aumento de la cantidad de pruebas de unidad necesarias para probar una clase.
Reutilización	Un aumento del NDD implica un aumento en el grado de reutilización de la clase.
Cantidad de pruebas	Un aumento del NDD implica una disminución en el nivel de cohesión de la clase.

Fuente: Vargas D. y Hernández M. 2009.

➤ **Número de Operaciones Redefinidas para una Sub-Clase (NOR):**

Tabla 15: Número de Operaciones Redefinidas para una Sub-Clase (NOR).

Atributo que afecta	Modo en que lo afecta
Abstracción del diseño	Un aumento del NOR implica que se ha logrado una buena definición de la abstracción del diseño de clases.
Cantidad de pruebas	Un aumento del NOR implica un aumento de la cantidad de pruebas de unidad necesarias para probar una clase.
Complejidad del mantenimiento	Un aumento del NOR implica un aumento de la complejidad del mantenimiento de la clase.

Fuente: Vargas D. y Hernández M. 2009.

(Lazo & Yzquierdo, 2007). A continuación se presentan los resultados obtenidos de la aplicación de los instrumentos de evaluación para medir las métricas anteriormente descritas, y una valoración de los mismos, que fueron realizados a la Capa de negocio de los componentes que conforman la solución propuesta.



2.4.1 Resultados del instrumento de evaluación de la métrica Tamaño de Clases (TC).

Ver instrumentos y tabla de resultados en (Anexo 16: Instrumento de medición de la métrica de Tamaño de Clases (TC).)

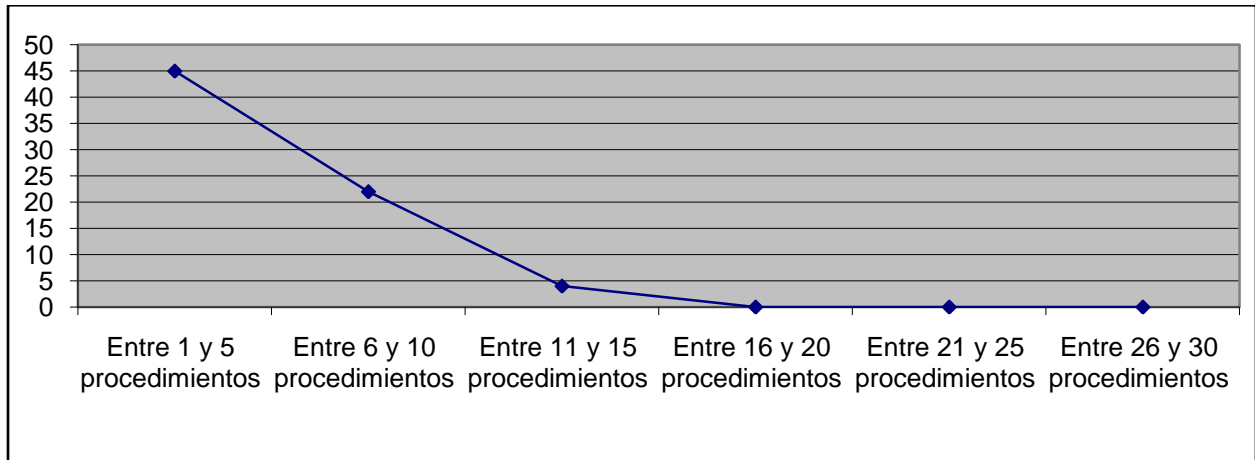


Figura 27: Representación de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

Fuente: Vargas D. y Hernández M. 2009.

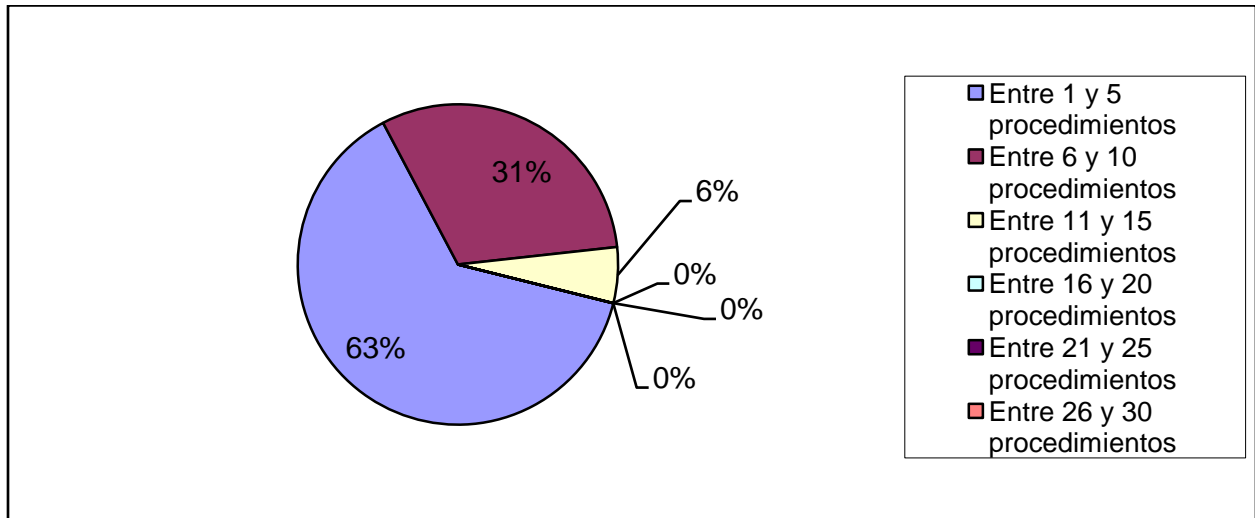


Figura 28: Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

Fuente: Vargas D. y Hernández M. 2009.

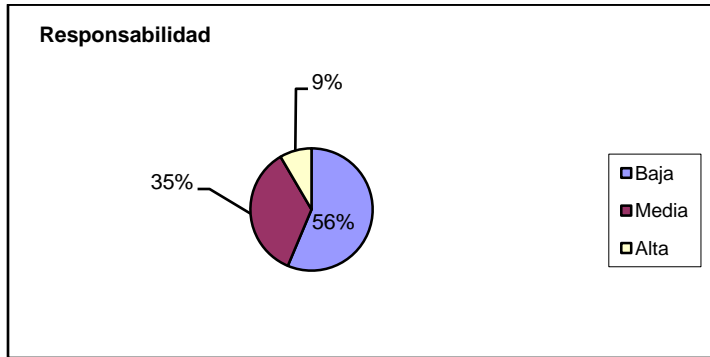


Figura 29: Representación de la incidencia de los resultados de la evaluación de la métrica TC en el atributo Responsabilidad.

Fuente: Vargas D. y Hernández M. 2009.

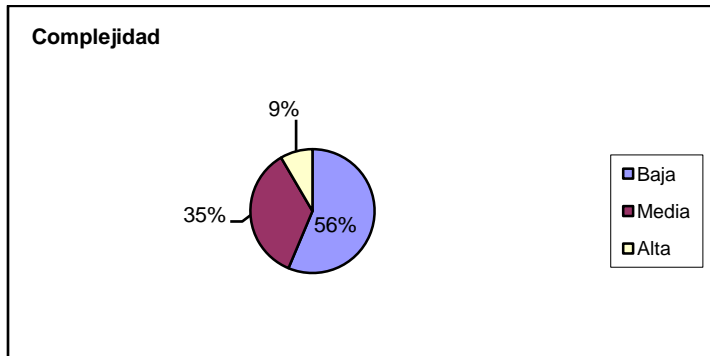


Figura 30: Representación de la incidencia de los resultados de la evaluación de la métrica TC en el atributo Complejidad de Implementación.

Fuente: Vargas D. y Hernández M. 2009.

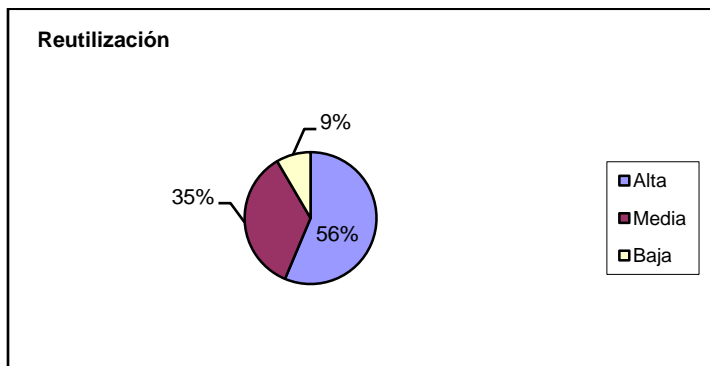


Figura 31: Representación de la incidencia de los resultados de la evaluación de la métrica TC en el atributo Reutilización.

Fuente: Vargas D. y Hernández M. 2009.

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica TC, se puede concluir que el diseño de los componentes Generador de Aspecto, Generador de Excepciones, Generador de Inversión de Control y Auditor de Código Fuente tienen una calidad favorable debido a que el 91% de las clases incluidas en estos componentes poseen menos cantidad de operaciones que la mitad del valor máximo registrado en las mediciones. Además el 91% de las clases poseen evaluaciones positivas en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).

2.4.2 Resultado del instrumento de evaluación de la métrica Respuesta para una clase (RPC).

Ver instrumentos y tablas en (Anexo 17: Instrumento de medición de la métrica Respuesta para una clase (RPC).)

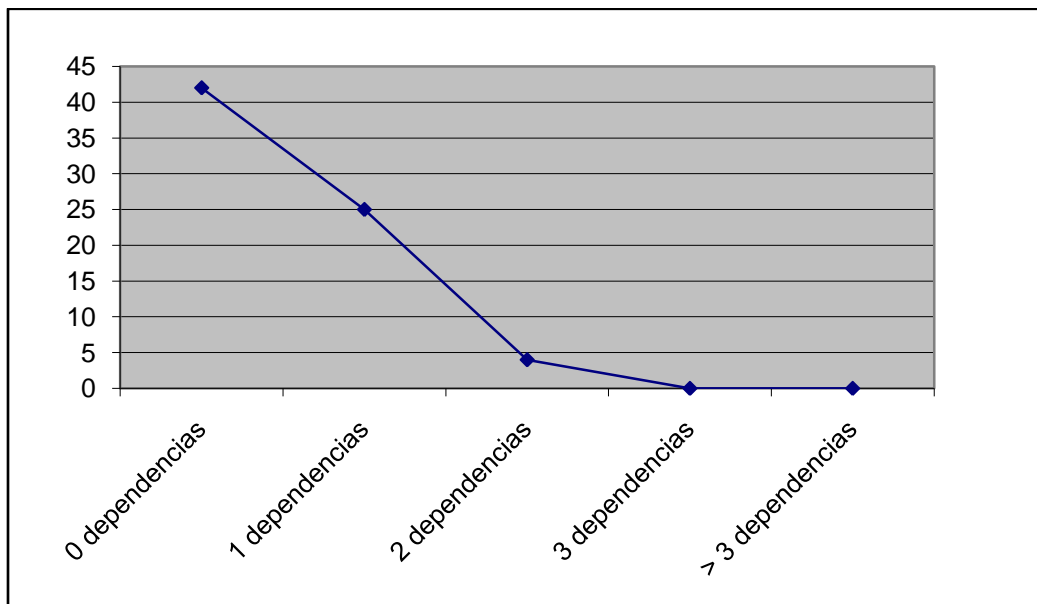


Figura 32: Gráfica de los resultados de la evaluación de la métrica RPC agrupados por la tendencia de los valores.

Fuente: Vargas D. y Hernández M. 2009.

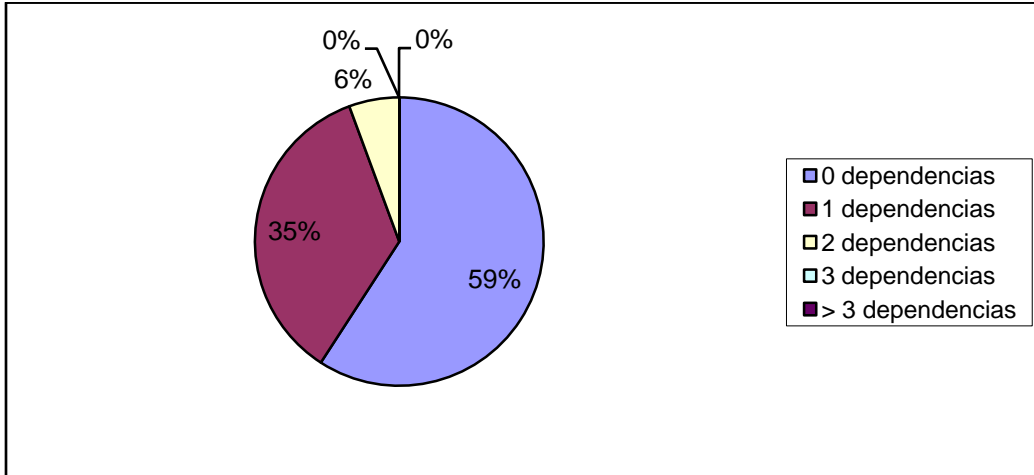


Figura 33: Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

Fuente: Vargas D. y Hernández M. 2009.

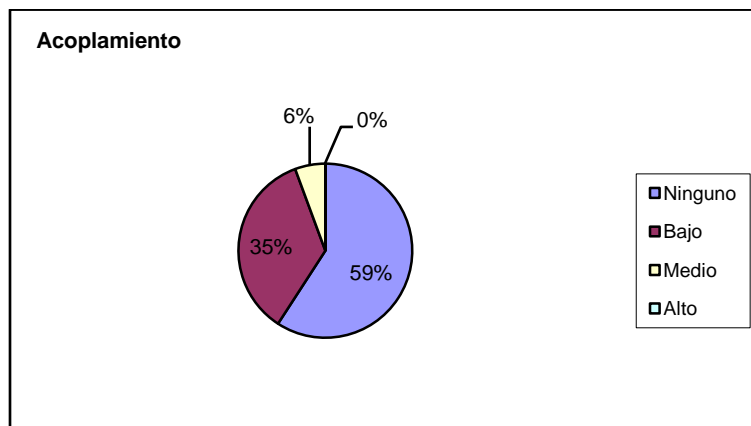


Figura 34: Representación de la incidencia de los resultados de la evaluación de la métrica RPC en el atributo Acoplamiento.

Fuente: Vargas D. y Hernández M. 2009.

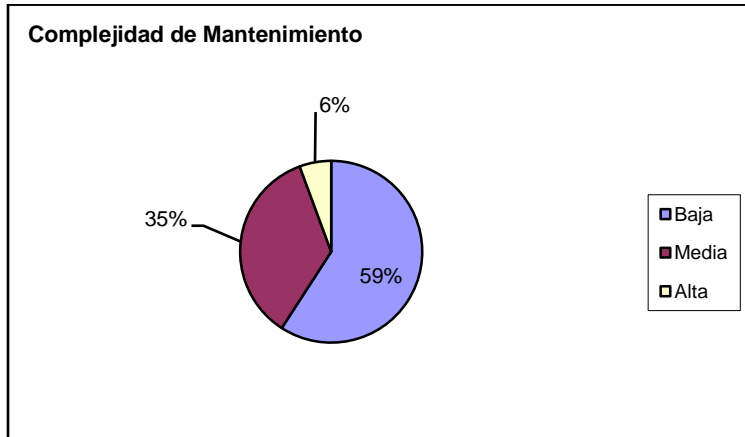


Figura 35: Representación de la incidencia de los resultados de la evaluación de la métrica RPC en el atributo Complejidad de Mantenimiento.

Fuente: Vargas D. y Hernández M. 2009.

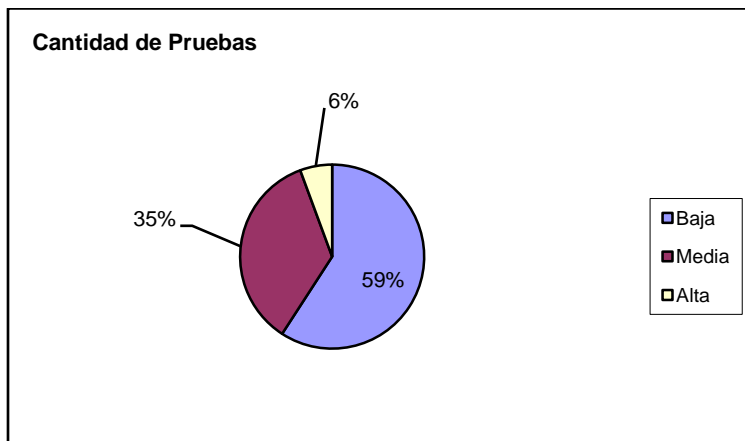


Figura 36: Representación de la incidencia de los resultados de la evaluación de la métrica RPC en el atributo Cantidad de Pruebas.

Fuente: Vargas D. y Hernández M. 2009.

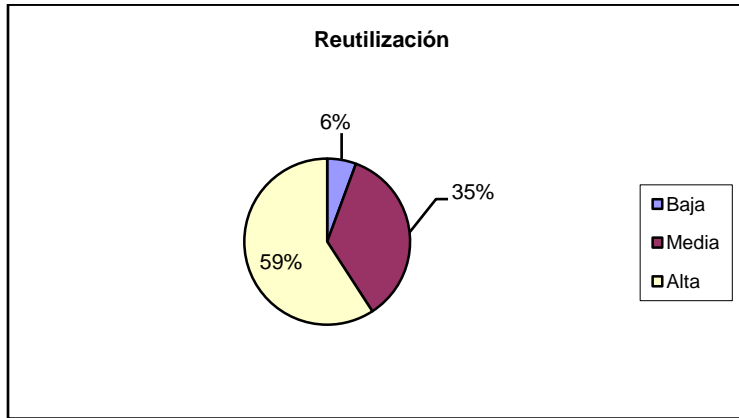


Figura 37: Representación de la incidencia de los resultados de la evaluación de la métrica RPC en el atributo Reutilización.

Fuente: Vargas D. y Hernández M. 2009.

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica RPC, se puede concluir que el diseño de los componentes Generador de Aspectos, Generador de Excepciones, Generador de Inversión de Control y Auditor de Código Fuente tienen una calidad aceptable teniendo en cuenta que el 100% de las clases incluidas en estos componentes posee menos de 3 dependencias de otras clases. Además el 59% de las clases no poseen acoplamiento con otras y el 94% posee índices aceptables en cuanto a Acoplamiento. Así mismo los atributos de calidad Complejidad de Mantenimiento, Cantidad de Pruebas y Reutilización se comportan satisfactoriamente en un 94% de las clases.

2.4.3 Resultados del instrumento de evaluación de la métrica Árbol de profundidad de herencia (APH).

Ver instrumentos y tablas de resultados en (Anexo 18: Instrumento de medición de la métrica Árbol de profundidad de Herencia (APH).)

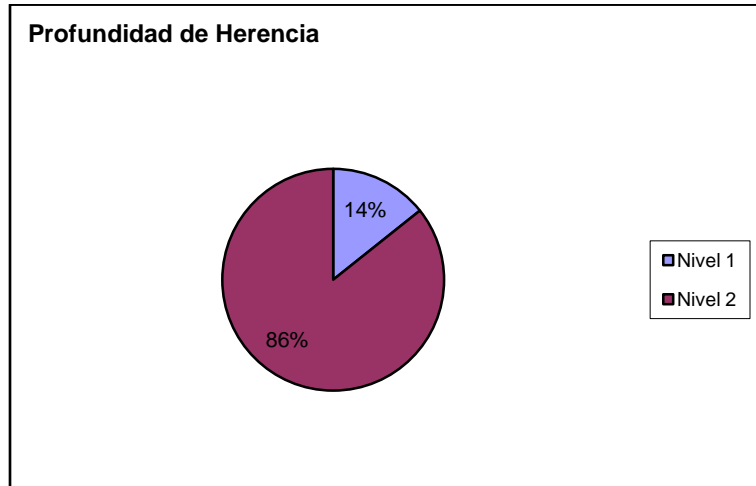


Figura 38: Representación en % de los resultados obtenidos en el instrumento agrupados por nivel.

Fuente: Vargas D. y Hernández M. 2009.

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica APH, se puede concluir que el diseño de los componentes Generador de Aspectos, Generador de Excepciones, Generador de Inversión de Control y Auditor de Código Fuente tienen una calidad aceptable teniendo en cuenta que la profundidad de herencia presentes en los subsistemas desarrollados es siempre de 2, el cual es el valor mínimo posible. Analizando los atributos de calidad Complejidad de Mantenimiento, Complejidad de Diseño se puede decir que sin duda se tienen buenos índices debido a que solo se cuenta como máximo con un nivel 2 de profundidad en la herencia. Sin embargo el atributo Reutilización no posee indicadores positivos debido al bajo nivel de profundidad de herencia.

2.4.4 Resultados del instrumento de evaluación de la métrica Número de Descendientes (NDD).

Ver instrumentos y tablas de resultados en (Anexo 19: Instrumento de medición de la métrica Número de Descendientes (NDD).)

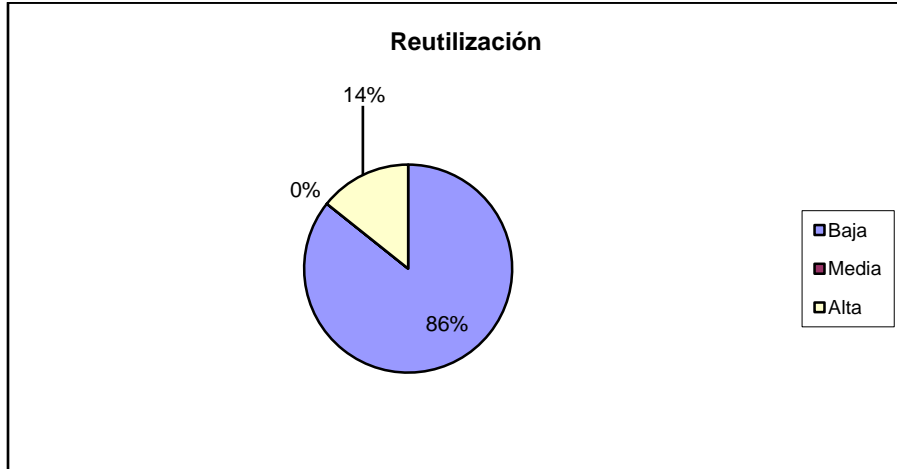


Figura 39: Representación de la incidencia de los resultados de la evaluación de la métrica NND en el atributo Reutilización.

Fuente: Vargas D. y Hernández M. 2009.

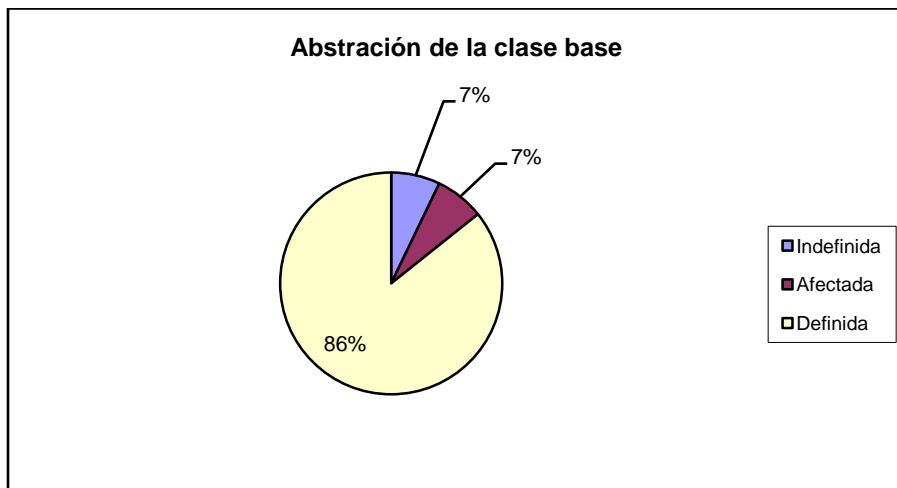


Figura 40: Representación de la incidencia de los resultados de la evaluación de la métrica NDD en el atributo Abstracción de la clase base.

Fuente: Vargas D. y Hernández M. 2009.

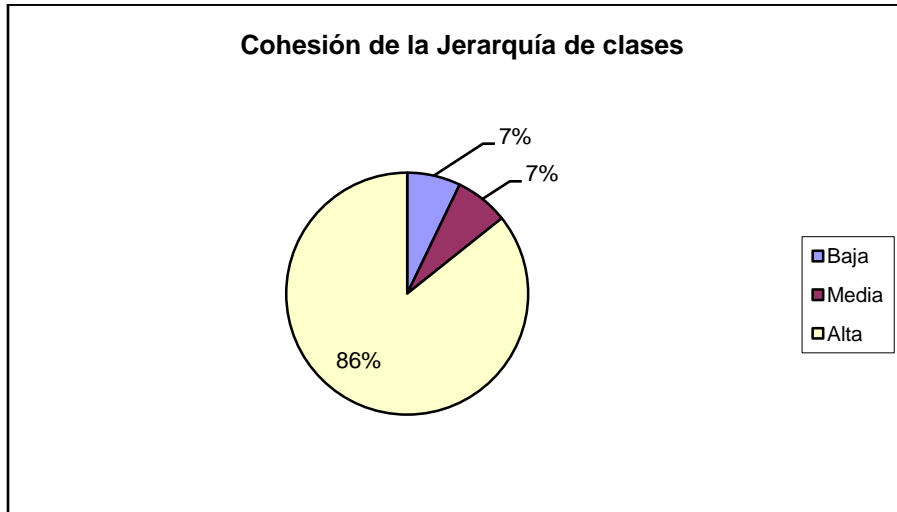


Figura 41: Representación de la incidencia de los resultados de la evaluación de la métrica NDD en el atributo Cohesión de la Jerarquía de clases.

Fuente: Vargas D. y Hernández M. 2009.

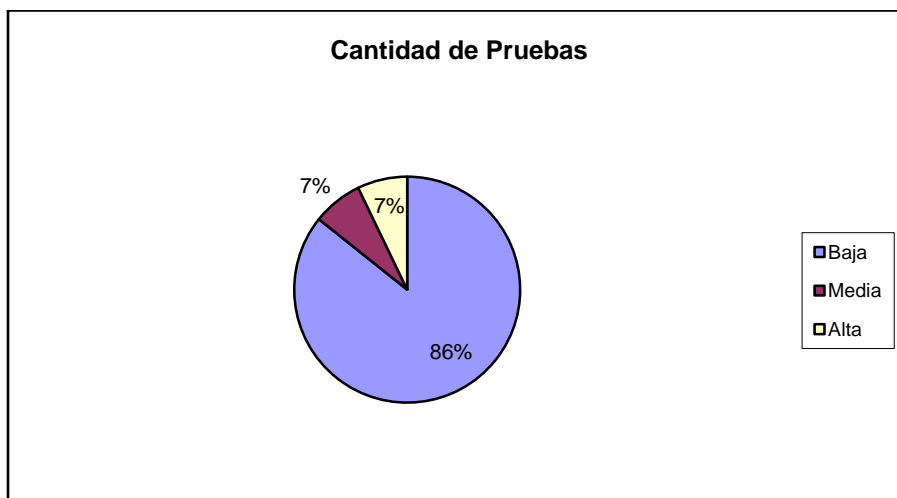


Figura 42: Representación de la incidencia de los resultados de la evaluación de la métrica NDD en el atributo Cantidad de Pruebas.

Fuente: Vargas D. y Hernández M. 2009.

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica NDD, se puede concluir que el diseño de los componentes Generador de Aspectos, Generador de Excepciones, Generador de Inversión de Control y Auditor de Código Fuente tienen una calidad aceptable teniendo en cuenta que solo se emplea la herencia en

casos bien identificados a partir de las necesidades del negocio o del diseño. Solo 2 clases poseen descendientes y solo en uno de los casos la cantidad de descendientes supera la cantidad de 5. Valorando el atributo de calidad Reutilización se puede decir que por lo antes explicado los índices de Reutilización se mantienen bajos representados por un 85%. En cuanto al atributo Abstracción de la clase base se muestra como existe una tendencia a la conservación de la abstracción. Tanto para la Cohesión de la jerarquía de clases como para el atributo Cantidad de Pruebas los índices son positivos favoreciendo esto al diseño.

2.4.5 Resultados del instrumento de evaluación de la métrica Número de Operaciones Redefinidas para una Sub-Clase (NOR).

Ver instrumentos y tabla de resultados en (Anexo 20: Instrumento de medición de la métrica Número de Operaciones Redefinidas para una Sub-Clase (NOR).)

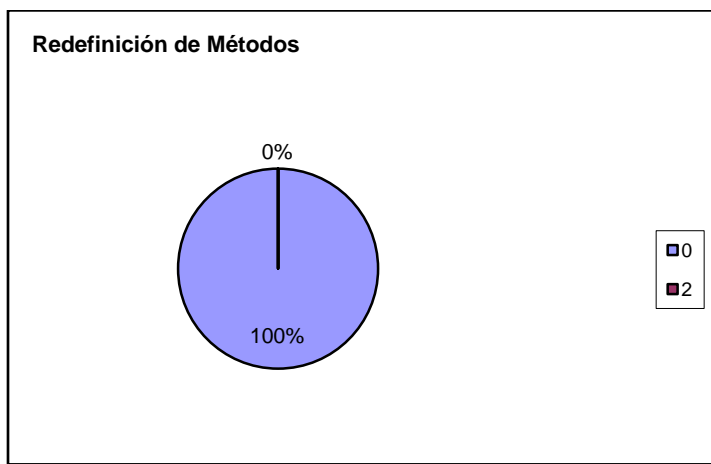


Figura 43: Representación en % de los resultados obtenidos en el instrumento agrupados en los valores existentes.

Fuente: Vargas D. y Hernández M. 2009.



Figura 44: Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Complejidad del Mantenimiento.

Fuente: Vargas D. y Hernández M. 2009.

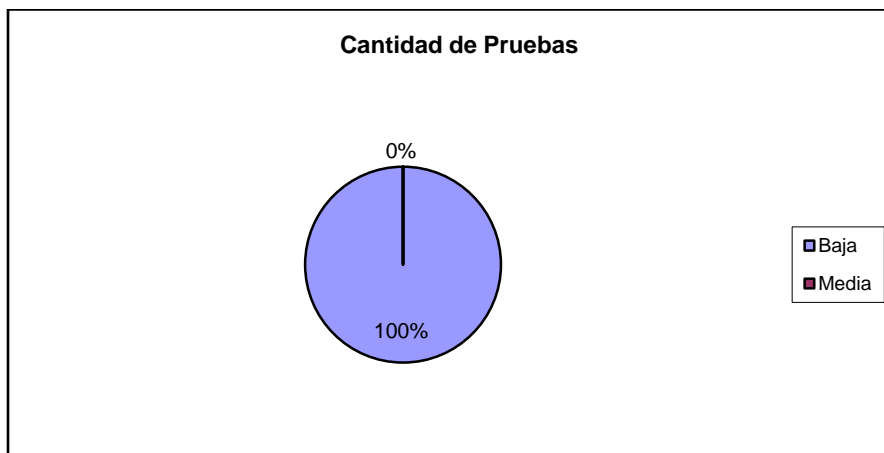


Figura 45: Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Cantidad de Pruebas.

Fuente: Vargas D. y Hernández M. 2009.



Figura 46: Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Violación de la Abstracción representada por la superclase.

Fuente: Vargas D. y Hernández M. 2009.

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica NOR, se puede concluir que el diseño de los componentes Generador de Aspectos, Generador de Excepciones, Generador de Inversión de Control y Auditor de Código Fuente tienen una calidad aceptable teniendo en cuenta que ninguna de las clases incluidas en el análisis (solo relacionadas con la herencia) redefinen algún método heredado. Además los indicadores se comportan de forma adecuada para los atributos de calidad Complejidad del Mantenimiento, Cantidad de Pruebas, Violación de la Abstracción representada por la superclase.

Matriz de cubrimiento o matriz de inferencia de indicadores de calidad.

La matriz de cubrimiento o matriz inferencia de indicadores de calidad es una representación estructurada de los atributos de calidad y métricas utilizadas en el capítulo anterior para evaluar la calidad del diseño de los componentes que integran la solución propuesta. La misma permite conocer si el resultado obtenido de la relación atributo/métricas para cada componente es positivo o negativo. Llevando estos resultados a una escalabilidad numérica donde, si los resultados son positivos tendrá un valor de 1, si son negativos de 0 y si no existe relación alguna se tomará como nula (-). Una vez completado los datos de dicha relación se realiza un

← →
 cálculo donde se promedia la sumatoria de los valores obtenidos de un atributo por cada métrica evaluada, y la división de dicha sumatoria por la cantidad de métricas evaluadas (solo se promedian las que arrojan un resultado, las nulas no). Este valor es el que va a tener el atributo dentro de una tabla que medirá si los atributos fueron buenos, regulares o malos. Se lograron los siguientes resultados:

Tabla 16: Resultados evaluados de la relación Atributos/Métricas por cada componente que integran la solución.

Atributos/Métricas	TOC	RPC	APH	NDD	NOR	Promedio
Responsabilidad	1	-	-	-	-	1
Complejidad del diseño	-	-	1	-	-	1
Complejidad de implementación	1	-	-	-	-	1
Reutilización	1	1	-	0	-	0,7
Acoplamiento	-	1	-	-	-	1
Complejidad del mantenimiento	-	1	-	-	1	1
Cantidad de pruebas	-	1	-	1	1	1
Nivel de cohesión	-	-	-	1	-	1
Abstracción del diseño	-	-	-	1	1	1

Fuente: Vargas D. y Hernández M. 2009.

Tabla 17: Rango de valores para la evaluación técnica de los atributos de calidad evaluados por cada métrica.

Calidad	Evaluación
Malo	≤ 0.4
Regular	>0.4 y ≤ 0.7
Bueno	>0.7

Fuente: Vargas D. y Hernández M. 2009.

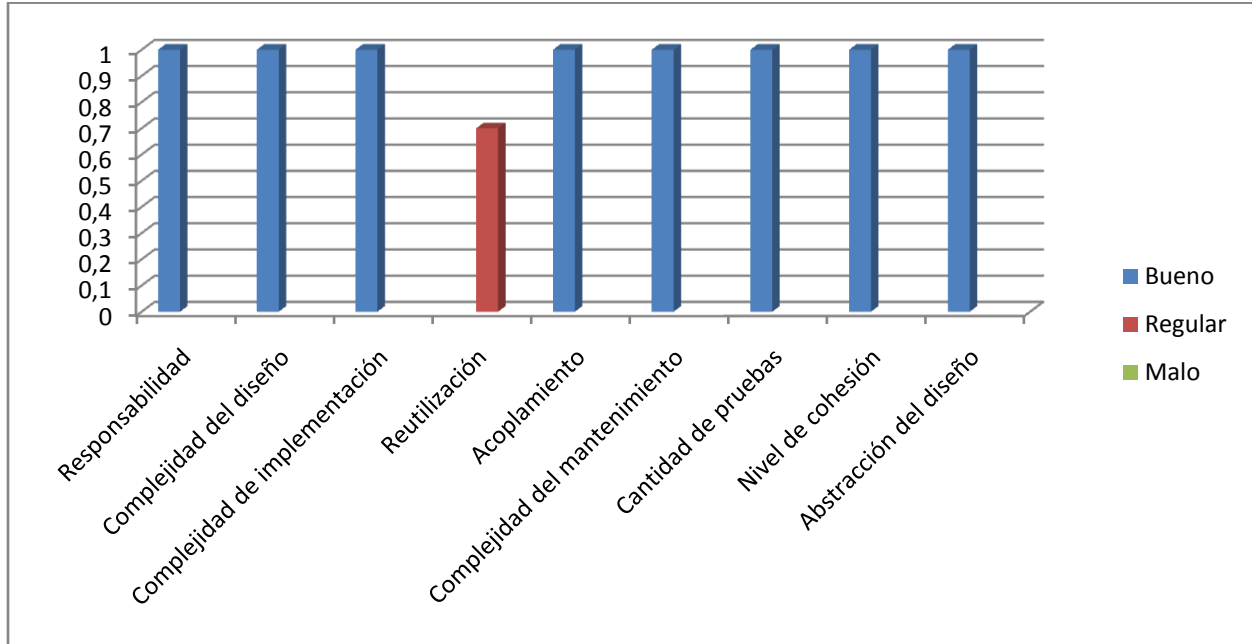


Figura 47: Gráfica de los resultados obtenidos de los atributos de calidad evaluados en las métricas.

Fuente: Vargas D. y Hernández M. 2009.

Luego de los resultados estadísticos plasmados en la matriz de cubrimiento o inferencia de indicadores de calidad se puede valorar que la solución propuesta mantiene una factible funcionalidad, manteniendo valores positivos para cada uno de los atributos de calidad relacionados con las métricas evaluadas en dichos componentes.

Conclusiones.

A modo de conclusión se puede decir que dicho capítulo abarcó los principales métodos para obtener el nivel de calidad de la solución propuesta. Mediante instrumentos inspirados en métricas para la calidad del diseño y basándose sobre los atributos de calidad trazados para llevar a cabo una medición estricta de cada uno de los componentes, los cuales permitieron afirmar que el diseño realizado se puede valorar de aceptable.

Conclusiones generales

Los estudios realizados para determinar el estado actual de los componentes que permiten configurar los aspectos arquitectónicos y la integración de aplicaciones, permitieron documentar la necesidad de desarrollar los componentes que motivan el presente trabajo de diploma, para lo cual se cumplió con los objetivos específicos propuestos.

Se diseñó e implementó un grupo de componentes que conforman la solución propuesta, para lo cual se utilizaron patrones arquitectónicos y estándares de diseño que propiciarían resultados positivos en materia de cohesión, acoplamiento, reutilización, calidad y facilidad de mantenimiento, lo que permitió obtener una herramienta capaz de manejar el marco de trabajo de una forma exitosa y consistente.

La solución fue probada por los métodos validados durante la investigación en el segundo capítulo respondiendo a los escenarios conformados por los requisitos funcionales. De esta forma la solución aportada cumple con las características de calidad exigidas por el proyecto ERP Cuba.

Para darle cumplimiento al último objetivo específico y base a la afirmación anterior se realiza una validación del diseño mediante la utilización de instrumentos de medición que se inspiraron en métricas para la calidad del diseño. Los resultados arrojados permitieron concluir que el diseño presentaba valores positivos en indicadores de calidad tales como Reutilización, Facilidad de Mantenimiento, Complejidad del Diseño, Complejidad de Implementación, Cohesión, Acoplamiento, Cantidad de pruebas, entre otros. Esto apoya la afirmación de que el diseño desarrollado se puede considerar como satisfactorio.

Recomendaciones

El equipo de trabajo que realiza la presente investigación considera importante distinguir las siguientes recomendaciones.

- Extender el uso de la aplicación a plataformas libres, basadas en GNU/Linux, para lo cual se recomienda compilar la aplicación utilizando el proyecto Mono, el cual es una herramienta libre compatible con el framework de .NET.
- Proveer la aplicación de otros componentes para la gestión del marco de trabajo de Cedrux, contribuyendo así al dinamismo en la producción.

Se recomienda finalmente el trazado de una estrategia que responda a la adaptación de los desarrolladores para el uso de los componentes del IDE y la aplicación creada en esta investigación, elevando de esta forma los indicadores de rendimiento y productividad en las líneas de desarrollo y módulos de Cedrux.

Bibliografía

Adison, S. (2003). *The Art of Software Architecture: Design methods and techniques*. Nueva York: Adison Wiley.

Alur, D., Crupi, J., & Malks, D. (2004). *Core J2EE Patterns, Best Practices and Design Strategies*. Prentice Hall.

ARREGUI, J. J. (Septiembre 2005.). *Revisión Sistemática de Métricas de Diseño Orientado a Objetos*. Madrid, España.: Universidad Politécnica de Madrid, Facultad de Informática.

Avila, S. (2005). *Introducción a Microsoft Solutions Framework*.

Barrios, N. F. (2005). *Investigación de la plataforma J2EE y su aplicación práctica*. Santiago, Chile.

Booch, G. (1991). *Object-Oriented Design with Applications*. The Benjamin: Cummings Publishing Company.

Buschmann, F. (2006). *Pattern-Oriented Software Architecture*. John Wiley & Sons.

Campos, S., & Martinez, L. (2006). *Programación Extrema: Prácticas, Aceptación y Controversia Volume I*.

Canos, J. H., & Penades, C. (2006). *Metodologías Agiles en el Desarrollo de Software*.

Castle, P. (2009). www.castleproject.org. Retrieved from Castle Project: <http://www.castleproject.org/container/gettingstarted/index.html>

Daedalus. (2006). *Diseño de sistemas*. Retrieved from <http://www.daedalus.es/AreasISDiseno-E.php>.

DeepInder, B. M. (2004). An integrative framework for the assimilation of Enterprise Resource Planning . In B. M. DeepInder, *An integrative framework for the assimilation of Enterprise Resource Planning* (Vols. Vol. 44, Iss. 3; pp. 81, 10.). Spring .

Dharma, S., Fell, S., & Sells, C. (2002). Aspect-Oriented Programming Enables Better Code Encapsulation and Reuse. *March 2002*.

Eini, O. (2006, November). *Inversion of Control and Dependency Injection: Working with Windsor Container*. Retrieved from MSDN.

Equipo Arquitectura del ERP Cedrux. (2008). *Especificación Técnica para el marco de la arquitectura*.

Estrada, A., & Estévez, I. (2003). Medir el proceso de control de configuración, ¿Una utopía para la Industria Nacional de Software?

Figueras, J. M. (2006). *Hacia una arquitectura con JavaServer Faces, Spring, Hibernate y otros frameworks*. Sevilla: Ediciones Alfar S.A.

Fowler, M. (2005, June 26). *Inversion of Control*. Retrieved from <http://www.martinfowler.com>

García, P. F., & López, O. (2002). Reutilización de Requisitos de usuario. . *Revista Comlombiana de Computación* .

Garlan, D. S. (1994). *An Introduction to Software Architecture*.

Greca, P. L. (2007, Octubre 17). *Bloque de conocimiento de de Epidata Consulting SRL: Spring IoC Container*. Retrieved abril 5 , 2009, from Bloque de conocimiento de de Epidata Consulting SRL: Spring IoC Container: <http://www.epidataconsulting.com>

IBM Corporation. (2003). *IBM Rational Unified Process*. Nueva York: United States of America.

Jacobson, I., Booch, G., & Rumbaugh, J. (2000). In *The Unified Software Development Process*. Addison-Wesley.

Lazo, O., & Yzquierdo, R. (2007). *El modelo de diseño del sistema HyperWeb. Módulos de Tratamiento Farmacológico y Configuración*. Habana.

López, O., Laguna, M. Á., & Marqués, J. M. (2005). Reutilización del Software a partir de Requisitos Funcionales en el Modelo Mecano: Comparación de escenarios. *Software Privileges Magazine* , 41-50.

Marinescu, F. (2004). *EJB Design Patterns*. Jhon Wiley & Sons.

Microsoft Corporation. (2006). *Microsoft Solution Framework*.

Microsoft Corporation. MSDN Community. (2009). Retrieved from <http://www.msdn.com/netframework>

Microsoft. (2007). *Diseño del software*.

Moreno, G. A. (2006, abril 25). *L.S.I. Departamento de lenguajes y sistemas informáticos. Universidad de Granada*. Retrieved abril 15, 2009, from L.S.I. Departamento de lenguajes y sistemas informáticos. Universidad de Granada.: <http://lsi.ugr.es>

Oficina Nacional del Gobierno Electrónico e Informática - ONGEI. (2004, mayo 28). Guía técnica sobre evaluación de software en la administración pública. *El Peruano* , p. 32.

Pressman, R. (2000). In R. Pressman, *Ingeniería del software: Un enfoque práctico*. McGraw Hill.

Proyect, C. (2009). www.castleproject.org. Retrieved from Castle Project: <http://www.castleproject.org/container/gettingstarted/index.html>

Rumbaugh, J., Jacobson, I., & Booch, G. (1998). *The Unified Modeling Language. Reference Manual*. Massachusetts: ADDISON-WESLEY. 568.

Rumbaugh, Jacobson, & Booch. (2002). *The unified Modeling Language Reference Manual*. Addison-Wesley.

Shang, S. (2000). A Comprehensive Framework for Classifying the Benefits of ERP Systems.

SHANG, S. (2000). A Comprehensive Framework for Classifying the Benefits of ERP Systems.

Sun Microsystems, Inc. (2007). *J2EE Connector Architecture Specification, Version 1.0*.

Valdés, I. J. (2008, Marzo). *Info.DvaT*. Retrieved Mayo 2009, from <http://jorgevaldes.over-blog.com/>

Vargas, A. (2005). Conferencias sobre Metodología de la Investigación. . *Impr. Universitaria. La Habana.* , 28-44.

Wikipedia Foundation .Inc. (2009, febrero 12). *Wikipedia, la enciclopedia libre - Caso de prueba*. Retrieved abril 15, 2009, from Wikipedia, la enciclopedia libre - Caso de prueba: <http://es.wikipedia.org>

Anexos

Anexo1: Plantilla de encuesta realizada a los distintos roles que integran el proyecto Cedrux.

Encuesta para la valoración de la configuración manual de los aspectos arquitectónicos de Cedrux.

Estimado usuario:

Con el objetivo de ayudarlos a hacer más eficiente su trabajo, estamos haciendo un diagnóstico sobre los problemas relacionados con la gestión de configuración de los aspectos arquitectónicos de Cedrux.

Le pedimos responda con la mayor veracidad el siguiente cuestionario:

***1. Señale el rol que ocupa dentro de su equipo de trabajo.**

- Jefe de Línea
- Analista
- Planificador
- Arquitecto
- Desarrollador
- Otro (Por favor especifique)

***2. Diga si Vd. ha utilizado los archivos de configuración del marco de trabajo.(ioc, exception, validation, weaver, etc.)**

Si No

***3. ¿Se encuentra satisfecho con la configuración manual de los archivos que gestionan el marco de trabajo?**

Si No

***4. Señale las dificultades encontradas durante la configuración manual de los archivos de gestión de la arquitectura.**

- Perdida de tiempo.
- Introducción de errores.
- Incremento del tiempo de máquina.
- Baja productividad del equipo de trabajo.
- Disminución de la moral del equipo de trabajo.
- Otro (Por favor especifique)

***5. Considera necesario el desarrollo de un componente para la configuración de los archivos que gestionan el marco de trabajo de la arquitectura de Cedrux?**

Si No

Fuente: Vargas D. y Hernández M. 2009.

Anexo 2: Requisitos funcionales del componente Generador de excepciones.

- R1 Gestionar excepciones.

Tabla 18: Paquete de R.F. de la gestión de ficheros del componente Generador de excepciones.

R1.1 Crear fichero.

Conceptos tratados	Conceptos	Atributos
	No procede.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	No procede.	No procede.
Descripción	Permite gestionar todo lo que lleva implícito el XML dentro.	
Validaciones	No procede.	
Post-condiciones	Una vez que este RF se lleve a cabo se crean los ficheros XML, y el árbol de navegación se actualiza con el nombre del root .	
Post-requisito	R1.1.2.1.1 Adicionar.	

Fuente: Vargas D. y Hernández M. 2009.

R1.2 Cargar fichero.

Conceptos tratados	Conceptos	Atributos
	No procede.	No procede.
Precondiciones	Precondiciones	Pre-requisito

	Tienen que existir los ficheros.	No procede.
Descripción	Permite gestionar los datos implícitos en el XML cargado.	
Validaciones	No procede.	
Post-condiciones	Una vez que este RF se lleve a cabo actualiza el árbol de navegación con la información contenida en los ficheros XML y en las tablas que se encuentran en la interfaz principal.	
Post-requisito	R1.1.2.1 Gestionar Módulos, R1.1.2.2 Gestionar controlador, R1.1.2.3 Gestionar acción, R1.1.1.1 Gestionar excepciones, R1.1.1.2 Gestionar lenguajes.	

Fuente: Vargas D. y Hernández M. 2009.

R1.3 Guardar fichero.

Conceptos tratados	Conceptos	Atributos
	No procede.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	No procede.	R1.1 Crear fichero o R1.2 Cargar fichero.
Descripción	Permite salvar todos los datos gestionados en los XML.	
Validaciones	No procede.	
Post-condiciones	Los ficheros han sido guardados en la ubicación elegida.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R1.4 Buscar.

Conceptos tratados	Conceptos	Atributos
---------------------------	------------------	------------------

	Módulo, Controlador, Acción, Excepción.	Nombre, código.
Precondiciones	Precondiciones	Pre-requisito
	Deben existir datos en el árbol de navegación.	R1.1 Crear fichero o R1.2 Cargar fichero.
Descripción	Permite encontrar algún módulo, controlador, acción o excepción que se encuentre en los ficheros XML o saber si no se encuentra para poder gestionarlo.	
Validaciones	No procede.	
Post-condiciones	Se muestra la ubicación física en el árbol de navegación.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R1.1.1 Gestionar excepciones (exception.xml).

R1.1.1.1 Gestionar excepciones.

Tabla 19: CRUD. del R1.1.1.1 Gestionar excepciones.

R1.1.1.1.1 Adicionar.

Conceptos tratados	Conceptos	Atributos
	Excepción.	Nombre, código, tipo de excepción, filtro.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir una acción.	R1.1.2.3.1 Adicionar.
Descripción	Permite registrar una excepción a una acción. Persiste en el fichero XML "exception" y "managerexception" los cambios.	
Validaciones	No puede tener el mismo código que alguna excepción existente.	

Post-condiciones	Ha sido adicionado una excepción a la acción elegida, y modificado los XML “managerexception” y “exception”.
Post-requisito	No procede.

Fuente: Vargas D. y Hernández M. 2009.

R1.1.1.1.2 Modificar.

Conceptos tratados	Conceptos	Atributos
	Excepción.	Nombre, código, tipo de excepción, filtro.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir la excepción.	R1.1.1.1.1 Adicionar.
Descripción	Permite modificar una excepción en una acción. Permite en el fichero XML “managerexception” y “exception” los cambios.	
Validaciones	No puede tener el mismo código que alguna excepción existente y demás datos.	
Post-condiciones	Ha sido modificada la excepción de la acción elegida, y modificado los XML “managerexception” y “exception”.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R1.1.1.1.3 Eliminar.

Conceptos tratados	Conceptos	Atributos
	Excepción.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir la excepción.	R1.1.1.1.1 Adicionar.
Descripción	Permite eliminar una excepción en una acción.	

	Persiste en el fichero XML “managerexception” y “exception” los cambios.
Validaciones	No procede.
Post-condiciones	Ha sido eliminada una excepción de la acción elegida, y modificado los XML “managerexception” y “exception”.
Post-requisito	No procede.

Fuente: Vargas D. y Hernández M. 2009.

R1.1.1.2 Gestionar lenguajes.

Tabla 20: CRUD. del R1.1.1.2 Gestionar lenguajes.

R1.1.1.2.1 Adicionar.

Conceptos tratados	Conceptos	Atributos
	Lenguaje.	Tipo de lenguaje, mensaje, descripción.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir una excepción.	R1.1.1.1.1 Adicionar.
Descripción	Permite registrar un lenguaje a una excepción. Persiste en el fichero XML “exception” los cambios.	
Validaciones	No puede tener los mismos datos que algún lenguaje existente.	
Post-condiciones	Ha sido adicionado un lenguaje a la excepción elegida, y modificado el XML “exception”.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R1.1.1.2.2 Modificar.

Conceptos tratados	Conceptos	Atributos
	Lenguaje.	Tipo de lenguaje, mensaje, descripción.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir el lenguaje.	R1.1.1.2.1 Adicionar.
Descripción	Permite modificar un lenguaje en una excepción. Permite en el fichero XML "exception" los cambios.	
Validaciones	No puede tener los mismos datos que algún lenguaje existente.	
Post-condiciones	Ha sido modificado el lenguaje de la excepción elegida, y modificado el XML "exception".	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R1.1.1.2.3 Eliminar.

Conceptos tratados	Conceptos	Atributos
	Lenguaje.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir el lenguaje.	R1.1.1.2.1 Adicionar.
Descripción	Permite eliminar un lenguaje en una excepción. Persiste en el fichero XML "exception" los cambios.	
Validaciones	No procede.	
Post-condiciones	Ha sido eliminada un lenguaje de la excepción elegida, y modificado el XML "exception".	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R1.1.2 Gestionar administrador de excepciones (managerexception.xml).

R1.1.2.1 Gestionar Módulos.

Tabla 21: CRUD. del R1.1.2.1 Gestionar módulos.

R1.1.2.1.1 Adicionar.

Conceptos tratados	Conceptos	Atributos
	Módulo.	Nombre.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir el nombre del root en el árbol de navegación.	R1.1 Crear fichero o R1.2 Guardar fichero.
Descripción	Permite registrar un módulo. Persiste en el fichero XML “managerexception” el nuevo módulo.	
Validaciones	No puede tener el mismo nombre que algún módulo existente.	
Post-condiciones	Ha sido adicionado un módulo nuevo, y modificado el XML “managerexception”.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R1.1.2.1.2 Modificar.

Conceptos tratados	Conceptos	Atributos
	Módulo.	Nombre.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir el módulo.	R1.1.2.1.1 Adicionar.

Descripción	Permite modificar los atributos de un módulo. Persiste en el fichero XML “managerexception” los cambios al módulo.
Validaciones	No puede tener el mismo nombre que algún módulo existente.
Post-condiciones	Ha sido modificado el módulo elegido, y modificado el XML “managerexception”.
Post-requisito	No procede.

Fuente: Vargas D. y Hernández M. 2009.

R1.1.2.1.3 Eliminar.

Conceptos tratados	Conceptos	Atributos
	Módulo.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir el módulo.	R1.1.2.1.1 Adicionar.
Descripción	Permite eliminar un módulo. Persiste en el fichero XML “managerexception” los cambios.	
Validaciones	No procede.	
Post-condiciones	Ha sido eliminado el módulo elegido, y modificado el XML “managerexception”.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R1.1.2.2 Gestionar controlador.

Tabla 22: CRUD. Del R1.1.2.2 Gestionar controlador.

R1.1.2.2.1 Adicionar.

Conceptos tratados	Conceptos	Atributos
---------------------------	------------------	------------------

	Controlador.	Nombre.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir un módulo en el árbol de navegación.	R1.1.2.1.1 Adicionar.
Descripción	<p>Permite registrar un controlador a un módulo.</p> <p>Permite adicionar un controlador que exista en otro módulo y no en este.</p> <p>Persiste en el fichero XML “managerexception” los cambios.</p>	
Validaciones	No puede tener el mismo nombre que algún controlador existente	
Post-condiciones	Ha sido adicionado un controlador al módulo elegido, y modificado el XML “managerexception”.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R1.1.2.2.2 Modificar.

Conceptos tratados	Conceptos	Atributos
	Controlador.	Nombre.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir el controlador.	R1.1.2.2.1 Adicionar.
Descripción	<p>Permite registrar un controlador a un módulo.</p> <p>Permite adicionar un controlador que exista en otro módulo y no en el presente.</p> <p>Persiste en el fichero XML “managerexception” los cambios.</p>	
Validaciones	No puede tener el mismo nombre que algún controlador existente.	
Post-condiciones	Ha sido modificado el controlador del módulo elegido, y modificado el XML “managerexception”.	

Post-requisito	No procede.
-----------------------	-------------

Fuente: Vargas D. y Hernández M. 2009.

R1.1.2.2.3 Eliminar.

Conceptos tratados	Conceptos	Atributos
	Controlador.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir el controlador.	R1.1.2.2.1 Adicionar.
Descripción	Permite eliminar un controlador de un módulo. Persiste en el fichero XML "managerexception" los cambios.	
Validaciones	No procede.	
Post-condiciones	Ha sido eliminado el controlador del módulo elegido, y modificado el XML "managerexception".	
Post-requisito	No procede.	

R1.1.2.3 Gestionar acción.

Tabla 23: CRUD. del R1.1.2.3 Gestionar acción.

R1.1.2.3.1 Adicionar.

Conceptos tratados	Conceptos	Atributos
	Acción.	Nombre.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir un controlador en el árbol de navegación.	R1.1.2.2.1 Adicionar.
Descripción	Permite registrar una acción a un controlador.	

	Persiste en el fichero XML “managerexception” los cambios.
Validaciones	No puede tener el mismo nombre que alguna acción existente, y modificado el XML “managerexception”.
Post-condiciones	Ha sido adicionada una acción al controlador elegido.
Post-requisito	No procede.

Fuente: Vargas D. y Hernández M. 2009.

R1.1.2.3.2 Modificar.

Conceptos tratados	Conceptos	Atributos
	Acción.	Nombre.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir la acción.	R1.1.2.3.1 Adicionar.
Descripción	Permite modificar una acción en un controlador. Persiste en el fichero XML “managerexception” los cambios.	
Validaciones	No puede tener el mismo nombre que alguna acción existente.	
Post-condiciones	Ha sido modificada la acción del controlador elegido, y modificado el XML “managerexception”.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R1.1.2.3.3 Eliminar.

Conceptos tratados	Conceptos	Atributos
	Acción.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir la acción.	R1.1.2.3.1 Adicionar.

Descripción	Permite eliminar una acción en un controlador. Persiste en el fichero XML “managerexception” los cambios.
Validaciones	No procede.
Post-condiciones	Ha sido eliminada la acción del controlador elegido, y modificado el XML “managerexception”.
Post-requisito	No procede.

Fuente: Vargas D. y Hernández M. 2009.

Anexo 3: Requisitos funcionales del componente Generador de aspectos.

- R2 Gestionar aspectos.

Tabla 24: Paquete de R.F. de la gestión de ficheros del componente Generador de aspectos.

R2.1 Crear fichero.

Conceptos tratados	Conceptos	Atributos
	No procede.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	No procede.	No procede.
Descripción	Permite gestionar todo lo que lleva implícito el XML dentro.	
Validaciones	No procede.	
Post-condiciones	Una vez que este RF se lleve a cabo se crean los ficheros XML, y el árbol de navegación se actualiza con el nombre del root .	
Post-requisito	R2.1.3.1.1 Adicionar.	

Fuente: Vargas D. y Hernández M. 2009.

R2.2 Cargar fichero.

Conceptos tratados	Conceptos	Atributos
	No procede.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	Tienen que existir los ficheros.	No procede.
Descripción	Permite gestionar los datos implícitos en el XML cargado.	
Validaciones	No procede.	
Post-condiciones	Una vez que este RF se lleve a cabo actualiza el árbol de navegación con la información contenida en los ficheros XML y en las tablas que se encuentran en la interfaz principal.	
Post-requisito	R2.1.3.1 Gestionar módulo, R2.1.3.2 Gestionar controlador, R2.1.3.3 Gestionar acción, R2.1.2 Gestionar plantilla de aspectos, R2.1.1 Gestionar aspectos arquitectónicos.	

Fuente: Vargas D. y Hernández M. 2009.

R2.3 Guardar fichero.

Conceptos tratados	Conceptos	Atributos
	No procede.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	No procede.	R2.1 Crear fichero o R2.2 Cargar fichero.
Descripción	Permite salvar todos los datos gestionados en los XML.	
Validaciones	No procede.	
Post-condiciones	Los ficheros han sido guardados en la ubicación elegida.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R2.4 Buscar.

Conceptos tratados	Conceptos	Atributos
	Módulo, Controlador, Acción, Plantilla de aspectos.	Nombre, código.
Precondiciones	Precondiciones	Pre-requisito
	Deben existir datos en el árbol de navegación.	R2.1 Crear fichero o R2.2 Cargar fichero.
Descripción	Permite encontrar algún módulo, controlador, acción o excepción que se encuentre en los ficheros XML o saber si no se encuentra para poder gestionarlo.	
Validaciones	No procede.	
Post-condiciones	Se muestra la ubicación física en el árbol de navegación.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R2.1.1 Gestionar aspectos arquitectónicos (aspect.xml).

Tabla 25: CRUD. del R2.1.1 Gestionar aspectos arquitectónicos (aspect.xml).

R2.1.1.1 Adicionar.

Conceptos tratados	Conceptos	Atributos
	Aspecto, Plantilla de aspectos	Nombre, Clase, Ejecución, Activo.
Precondiciones	Precondiciones	Pre-requisito
	Exista la plantilla de aspectos correspondiente.	R2.1.2.1 Adicionar.
Descripción	Permite configurar un nuevo aspecto y adicionarlo a una plantilla de	

	<p>aspectos.</p> <p>Permite adicionar un aspecto existente en otra plantilla a esta.</p> <p>Persiste la información en los correspondientes ficheros XML.</p>
Validaciones	<p>Los datos concuerdan con la expresión regular creada para los mismos.</p> <p>Que no exista un aspecto con el mismo nombre en dicha plantilla.</p> <p>Persiste el nuevo aspecto en los ficheros XML "aspect" y "aspecttemplate"</p>
Post-condiciones	Se ha registrado un nuevo aspecto
Post-requisito	No procede.

Fuente: Vargas D. y Hernández M. 2009.

R2.1.1.2 Modificar.

Conceptos tratados	Conceptos	Atributos
	Aspecto, Plantilla de aspectos	Nombre, Clase, Ejecución, Activo.
Precondiciones	Precondiciones	Pre-requisito
	Exista la plantilla de aspectos correspondiente.	R2.1.1.1 Adicionar.
Descripción	<p>Permite modificar los atributos de aspecto y persistir la nueva configuración</p> <p>Persiste el nuevo aspecto en los ficheros XML "aspect" y "aspecttemplate"</p>	
Validaciones	<p>Los datos concuerdan con la expresión regular creada para los mismos.</p> <p>Que no exista un aspecto con el mismo nombre en dicha plantilla.</p>	
Post-condiciones	Se han modificado los atributos del aspecto	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R2.1.1.3 Eliminar.

Conceptos tratados	Conceptos	Atributos
	Aspecto, Plantilla de aspectos	Nombre, Clase, Ejecución, Activo.
Precondiciones	Precondiciones	Pre-requisito
	Exista la plantilla de aspectos correspondiente.	R2.1.1.1 Adicionar.
Descripción	Permite eliminar un aspecto de una plantilla de aspectos. Antes de eliminar se solicita confirmación al usuario. Persiste el nuevo aspecto en los ficheros XML "aspect" y "aspecttemplate"	
Validaciones	No procede.	
Post-condiciones	Se ha eliminado el aspecto seleccionado.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R2.1.2 Gestionar plantilla de aspectos (aspecttemplate.xml).

Tabla 26: CRUD. Del R2.1.2 Gestionar plantilla de aspectos (aspecttemplate.xml).

R2.1.2.1 Adicionar.

Conceptos tratados	Conceptos	Atributos
	Aspecto, Plantilla de aspectos, Acción	Nombre
Precondiciones	Precondiciones	Pre-requisito
	Existe la acción correspondiente.	R2.1.3.3.1 Adicionar.

Descripción	<p>Permite registrar una plantilla de aspectos.</p> <p>Persiste en los ficheros XML “weaver” y “aspecttemplate” la nueva plantilla de aspectos en su acción correspondiente.</p>
Validaciones	Los datos concuerdan con la expresión regular creada para los mismos.
Post-condiciones	Se ha registrado la nueva plantilla de aspectos en los ficheros correspondientes.
Post-requisito	No procede.

Fuente: Vargas D. y Hernández M. 2009.

R2.1.2.2 Modificar.

Conceptos tratados	Conceptos	Atributos
	Aspecto, Plantilla de aspectos, Acción	Nombre
Precondiciones	Precondiciones	Pre-requisito
	Existe la acción correspondiente.	R2.1.2.1 Adicionar.
Descripción	<p>Permite modificar los atributos de la plantilla de aspectos.</p> <p>Persiste en los ficheros XML “weaver” y “aspecttemplate” los cambios a la plantilla de aspectos.</p>	
Validaciones	Los datos concuerdan con la expresión regular creada para los mismos.	
Post-condiciones	Se ha registrado la nueva plantilla de aspectos en los ficheros correspondientes.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R2.1.2.3 Eliminar.

Conceptos tratados	Conceptos	Atributos
	Aspecto, Plantilla de aspectos, Acción	Nombre
Precondiciones	Precondiciones	Pre-requisito
	Existe la acción correspondiente.	R2.1.2.1 Adicionar.
Descripción	Permite eliminar la plantilla de aspectos correspondiente. Persiste en los ficheros XML “weaver” y “aspecttemplate”.	
Validaciones	No procede.	
Post-condiciones	Se ha eliminado nueva plantilla de aspectos en los ficheros correspondientes.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R2.1.3 Gestionar tejedor de aspectos (weaver.xml).

R2.1.3.1 Gestionar módulo

Tabla 27: CRUD. del R2.1.3.1 Gestionar módulo perteneciente al R2.1.3 Gestionar tejedor de aspectos (weaver.xml).

R2.1.3.1.1 Adicionar.

Conceptos tratados	Conceptos	Atributos
	Módulo.	Nombre módulo
Precondiciones	Precondiciones	Pre-requisito
	Han sido cargado o creados los ficheros.	R2.1 Crear fichero o R2.2 Cargar fichero.

Descripción	Permite registrar una plantilla de aspectos. Persiste en el fichero XML “weaver” el nuevo módulo.
Validaciones	Los datos concuerdan con la expresión regular creada para los mismos. No existe un módulo con ese nombre.
Post- condiciones	Se ha registrado el nuevo módulo.
Post-requisito	No procede.

Fuente: Vargas D. y Hernández M. 2009.

R2.1.3.1.2 Modificar.

Conceptos tratados	Conceptos	Atributos
	Módulo.	Nombre módulo
Precondiciones	Precondiciones	Pre-requisito
	Existe el módulo correspondiente.	R2.1.3.1.1 Adicionar.
Descripción	Permite modificar los atributos de un módulo. Persiste en el fichero XML “weaver” los cambios al módulo.	
Validaciones	Los datos concuerdan con la expresión regular creada para los mismos. No existe un módulo con ese nombre.	
Post- condiciones	Se ha cambiado el nuevo módulo.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R2.1.3.1.3 Eliminar.

Conceptos tratados	Conceptos	Atributos
	Módulo.	Nombre módulo
Precondiciones	Precondiciones	Pre-requisito
	Existe el módulo correspondiente.	R2.1.3.1.1 Adicionar.
Descripción	Permite eliminar un módulo. Persiste en el fichero XML “weaver” los cambios.	
Validaciones	No procede.	
Post-condiciones	Se ha cambiado eliminado el módulo.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R2.1.3.2 Gestionar controlador.

Tabla 28: CRUD. del R2.1.3.2 Gestionar controlador.

R2.1.3.2.1 Adicionar.

Conceptos tratados	Conceptos	Atributos
	Controlador, módulo.	Nombre controlador.
Precondiciones	Precondiciones	Pre-requisito
	Existe el módulo para agregar el controlador correspondiente.	R2.1.3.1.1 Adicionar.
Descripción	Permite registrar un controlador a un módulo. Permite adicionar un controlador que exista en otro módulo y no en este.	

	Persiste en el fichero XML “weaver” los cambios.
Validaciones	Los datos concuerdan con la expresión regular creada para los mismos. No existe un controlador con el mismo nombre en este módulo.
Post-condiciones	Se ha registrado el controlador en el módulo correspondiente.
Post-requisito	No procede.

Fuente: Vargas D. y Hernández M. 2009.

R2.1.3.2.2 Modificar.

Conceptos tratados	Conceptos	Atributos
	Controlador, módulo.	Nombre controlador.
Precondiciones	Precondiciones	Pre-requisito
	Existe el controlador correspondiente.	R2.1.3.2.1 Adicionar.
Descripción	Permite registrar un controlador a un módulo. Permite adicionar un controlador que exista en otro módulo y no en el presente. Persiste en el fichero XML “weaver” los cambios.	
Validaciones	Los datos concuerdan con la expresión regular creada para los mismos. No existe un controlador con el mismo nombre en este módulo.	
Post-condiciones	Se han modificado los atributos del controlador en el módulo correspondiente.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R2.1.3.2.3 Eliminar.

Conceptos tratados	Conceptos	Atributos
	Controlador, módulo.	Nombre controlador.
Precondiciones	Precondiciones	Pre-requisito
	Existe el controlador correspondiente.	R2.1.3.2.1 Adicionar.
Descripción	Permite eliminar un controlador de un módulo. Persiste en el fichero XML “weaver” los cambios.	
Validaciones	Los datos concuerdan con la expresión regular creada para los mismos.	
Post-condiciones	Se ha eliminado el controlador en el módulo correspondiente.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R2.1.3.3 Gestionar acción.

Tabla 29: CRUD. Del R2.1.3.3 Gestionar acción.

R2.1.3.3.1 Adicionar.

Conceptos tratados	Conceptos	Atributos
	Acción, controlador.	Nombre acción, aspecttemplate
Precondiciones	Precondiciones	Pre-requisito
	Existe el controlador para agregar la acción correspondiente.	R2.1.3.2.1 Adicionar.
Descripción	Permite registrar una acción a un controlador. Persiste en el fichero XML “weaver” los cambios.	

Validaciones	Los datos concuerdan con la expresión regular creada para los mismos. No existe una acción con el mismo nombre en este controlador.
Post-condiciones	Se ha registrado la acción en el controlador correspondiente.
Post-requisito	No procede.

Fuente: Vargas D. y Hernández M. 2009.

R2.1.3.3.2 Modificar.

Conceptos tratados	Conceptos	Atributos
	Acción, controlador.	Nombre acción
Precondiciones	Precondiciones	Pre-requisito
	Existe la acción correspondiente.	R2.1.3.3.1 Adicionar.
Descripción	Permite modificar una acción en un controlador. Persiste en el fichero XML “weaver” los cambios.	
Validaciones	Los datos concuerdan con la expresión regular creada para los mismos. No existe una acción con el mismo nombre en este controlador.	
Post-condiciones	Se han modificado la acción en el controlador correspondiente.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R2.1.3.3.3 Eliminar.

Conceptos tratados	Conceptos	Atributos
	Acción, controlador.	Nombre acción

Precondiciones	Precondiciones	Pre-requisito
	Existe la acción correspondiente.	R2.1.3.3.1 Adicionar.
Descripción	Permite eliminar una acción en un controlador. Persiste en el fichero XML "weaver" los cambios.	
Validaciones	No procede.	
Post-condiciones	Se ha eliminado la acción en el controlador correspondiente.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

Anexo 4: Requisitos funcionales del componente Generador de inversión de control.

- R3 Gestionar inversión de control.

R3.1 Gestionar fichero (ioc.xml).

Tabla 30: Paquete de R.F. de la gestión de ficheros del componente Generador de inversión de control.

R3.1.1 Crear.

Conceptos tratados	Conceptos	Atributos
	fichero	Nombre, ubicación, descripción.
Precondiciones	Precondiciones	Pre-requisito
	No procede.	No procede.
Descripción	Permite gestionar todo lo que lleva implícito el XML dentro.	

Validaciones	No procede.
Post-condiciones	Una vez que este RF se lleve a cabo permite gestionar las distintas funciones que están implícitos en el componente.
Post-requisito	R3.2.1 Adicionar.

Fuente: Vargas D. y Hernández M. 2009.

R3.1.2 Cargar fichero.

Conceptos tratados	Conceptos	Atributos
	Fichero.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	Tienen que existir los ficheros.	R3.1.1 Crear.
Descripción	Permite gestionar los datos implícitos en el XML cargado.	
Validaciones	No procede.	
Post-condiciones	Una vez que este RF se lleve a cabo se actualizan todas las tablas de datos del componente.	
Post-requisito	R 3. 2 Gestionar subsistema, R3.3 Gestionar contrato.	

Fuente: Vargas D. y Hernández M. 2009.

R3.1.3 Guardar fichero.

Conceptos tratados	Conceptos	Atributos
	Fichero.	Nombre, ubicación, descripción.
Precondiciones	Precondiciones	Pre-requisito
	No procede.	R3.1.1 Crear o R3.1.2 Cargar fichero.
Descripción	Permite salvar todos los datos gestionados en los XML.	

Validaciones	No procede.
Post-condiciones	Los ficheros han sido guardados en la ubicación elegida.
Post-requisito	No procede.

Fuente: Vargas D. y Hernández M. 2009.

R 3. 2 Gestionar subsistema

Tabla 31: CRUD. del R3.2 Gestionar subsistema.

R3.2.1 Adicionar

Conceptos tratados	Conceptos	Atributos
	Subsistema.	Nombre, ubicación, descripción.
Precondiciones	Precondiciones	Pre-requisito
	Debe de existir el fichero.	R3.1.1 Crear o R3.1.2 Cargar fichero.
Descripción	Permite adicionar un subsistema nuevo al fichero. Permite modificar la información del fichero "ioc".	
Validaciones	No procede.	
Post-condiciones	Una vez que este RF se lleve a cabo permite gestionar las diversas funcionalidades inferiores a el.	
Post-requisito	R3.3 Gestionar contrato.	

Fuente: Vargas D. y Hernández M. 2009.

R3.2.2 Eliminar

Conceptos tratados	Conceptos	Atributos
	Subsistema.	No procede.
Precondiciones	Precondiciones	Pre-requisito

	Debe de existir el subsistema.	R3.2.1 Adicionar
Descripción	Permite eliminar un subsistema existente en el fichero. Permite modificar la información del fichero “ioc”.	
Validaciones	No procede.	
Post-condiciones	No procede.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R3.3 Gestionar contrato.

Tabla 32: CRUD. Del R3.3 Gestionar contrato.

R3.3.1 Adicionar.

Conceptos tratados	Conceptos	Atributos
	Contrato.	Nombre, clase inyectora, nombre función inyectora, referencia.
Precondiciones	Precondiciones	Pre-requisito
	Debe de existir el subsistema.	R3.2.1 Adicionar
Descripción	Permite adicionar un nuevo contrato al fichero. Permite adicionar un nuevo contrato al subsistema. Permite modificar la información del fichero “ioc”.	
Validaciones	No procede.	
Post-condiciones	No procede.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R3.3.1.1 Gestionar prototipo.

Tabla 33: CRUD. Del R3.3.1.1 Gestionar prototipo.

R3.3.1.1.1 Adicionar.

Conceptos tratados	Conceptos	Atributos
	Prototipo.	Nombre resultado, tipo resultado, parámetro (nombre parámetro, tipo parámetro).
Precondiciones	Precondiciones	Pre-requisito
	Debe de existir el contrato.	R3.2.1 Adicionar
Descripción	Permite adicionar un nuevo prototipo al contrato. Permite modificar la información del fichero "ioc".	
Validaciones	No procede.	
Post-condiciones	No procede.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R3.3.1.1.2 Modificar.

Conceptos tratados	Conceptos	Atributos
	Prototipo.	Nombre resultado, tipo resultado, parámetro (nombre parámetro, tipo parámetro).
Precondiciones	Precondiciones	Pre-requisito
	Debe de existir el prototipo.	R3.3.1.1.1 Adicionar.
Descripción	Permite modificar el prototipo del contrato. Permite modificar la información del fichero "ioc".	

Validaciones	No procede.
Post-condiciones	No procede.
Post-requisito	No procede.

Fuente: Vargas D. y Hernández M. 2009.

R3.3.1.1.3 Eliminar.

Conceptos tratados	Conceptos	Atributos
	Prototipo.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	Debe de existir el prototipo.	R3.3.1.1.1 Adicionar
Descripción	Permite eliminar un prototipo al contrato. Permite modificar la información del fichero "ioc".	
Validaciones	No procede.	
Post-condiciones	No procede.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R3.3.2 Modificar.

Conceptos tratados	Conceptos	Atributos
	Contrato.	Nombre, clase inyectora, nombre función inyectora, referencia.
Precondiciones	Precondiciones	Pre-requisito
	Debe de existir el prototipo.	R3.3.1 Adicionar.
Descripción	Permite modificar el contrato del subsistema. Permite modificar la información del fichero "ioc".	

Validaciones	No procede.
Post-condiciones	No procede.
Post-requisito	No procede.

Fuente: Vargas D. y Hernández M. 2009.

R3.3.3 Eliminar.

Conceptos tratados	Conceptos	Atributos
	Contrato.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	Debe de existir el prototipo.	R3.3.1 Adicionar.
Descripción	Permite eliminar el contrato del subsistema. Permite modificar la información del fichero "ioc".	
Validaciones	No procede.	
Post-condiciones	No procede.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

Anexo 5: Requisitos funcionales del componente Auditor de Código Fuente.

- Auditor de Código Fuente.

R4.1 Gestionar proyecto.

Tabla 34: Paquete de R.F. de la gestión de ficheros del componente Auditor de código fuente.

R4.1.1 Crear proyecto.

Conceptos tratados	Conceptos	Atributos
	Proyecto.	Nombre.
Precondiciones	Precondiciones	Pre-requisito
	No procede.	No procede.
Descripción	Permite crear un proyecto al sistema. Permite instanciar los ficheros del proyecto.	
Validaciones	No procede.	
Post-condiciones	No procede.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R4.1.2 Abrir proyecto.

Conceptos tratados	Conceptos	Atributos
	Proyecto.	Nombre.
Precondiciones	Precondiciones	Pre-requisito
	Debe de existir el proyecto registrado en la base de datos.	R4.1.1 Crear proyecto.
Descripción	Permite cargar un proyecto al sistema. Permite auditar los ficheros del proyecto.	
Validaciones	No procede.	
Post-condiciones	No procede.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R4.2 Gestionar ficheros.

Tabla 35: Funcionalidades del R4.2 Gestionar ficheros.

R4.2.1 Instanciar.

Conceptos tratados	Conceptos	Atributos
	No procede.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	Debe de haberse creado el proyecto.	R4.1.1 Crear proyecto.
Descripción	Permite instanciar los ficheros del proyecto. Guardar en la base de datos del sistema los datos del CRC, y la firma digital realizada a los ficheros del proyecto.	
Validaciones	No procede.	
Post-condiciones	No procede.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

R4.2.2 Auditar.

Conceptos tratados	Conceptos	Atributos
	No procede.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	Debe de haberse abierto el proyecto.	R4.1.2 Abrir proyecto.
Descripción	Permite auditar los ficheros del proyecto. Permite cargar los ficheros del sistema y calcularle el CRC para comparar con el CRC de los ficheros guardaos en la base de datos.	

Validaciones	No procede.
Post-condiciones	No procede.
Post-requisito	No procede.

Fuente: Vargas D. y Hernández M. 2009.

R4.2.3 Chequear.

Conceptos tratados	Conceptos	Atributos
	No procede.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	Debe de haberse abierto el proyecto.	R4.1.2 Abrir proyecto.
Descripción	Permite establecer una comparación entre los ficheros del sistema y los de la base de datos, en cuanto a firma digital y CRC, para la verificación de si hubo algún cambio.	
Validaciones	No procede.	
Post-condiciones	No procede.	
Post-requisito	No procede.	

Fuente: Vargas D. y Hernández M. 2009.

Anexo 6: Prototipos de interfaz que ameritan a los requisitos funcionales del componente Auditor de código fuente.

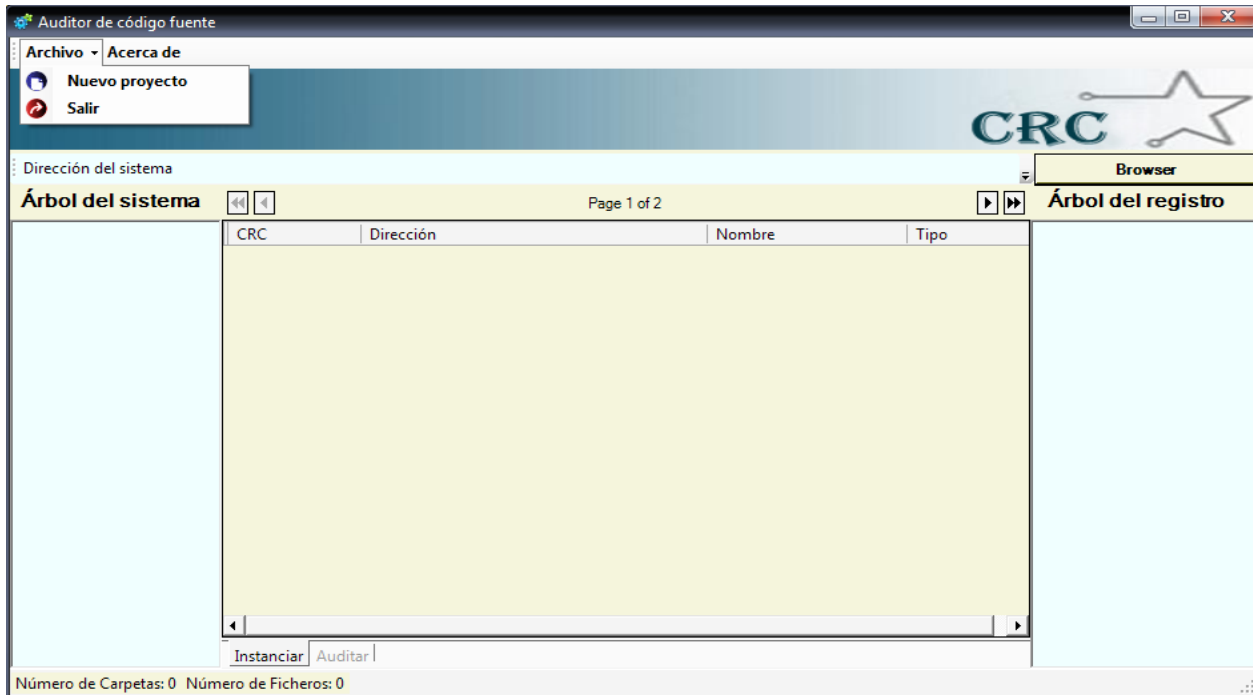


Figura 48: Interfaz que amerita al paquete R4.1 Gestionar proyectos.

Fuente: Vargas D. y Hernández M. 2009.

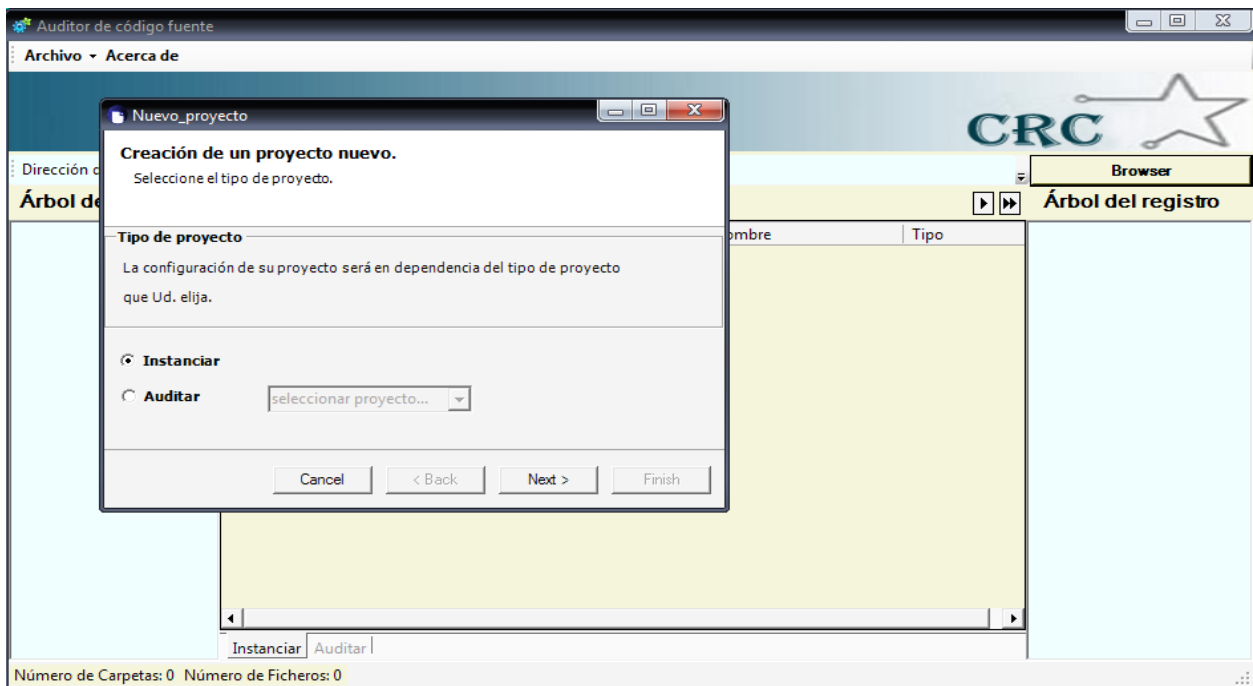


Figura 49: Interfaz que amerita al R4.2 Gestionar ficheros.

Fuente: Vargas D. y Hernández M. 2009.

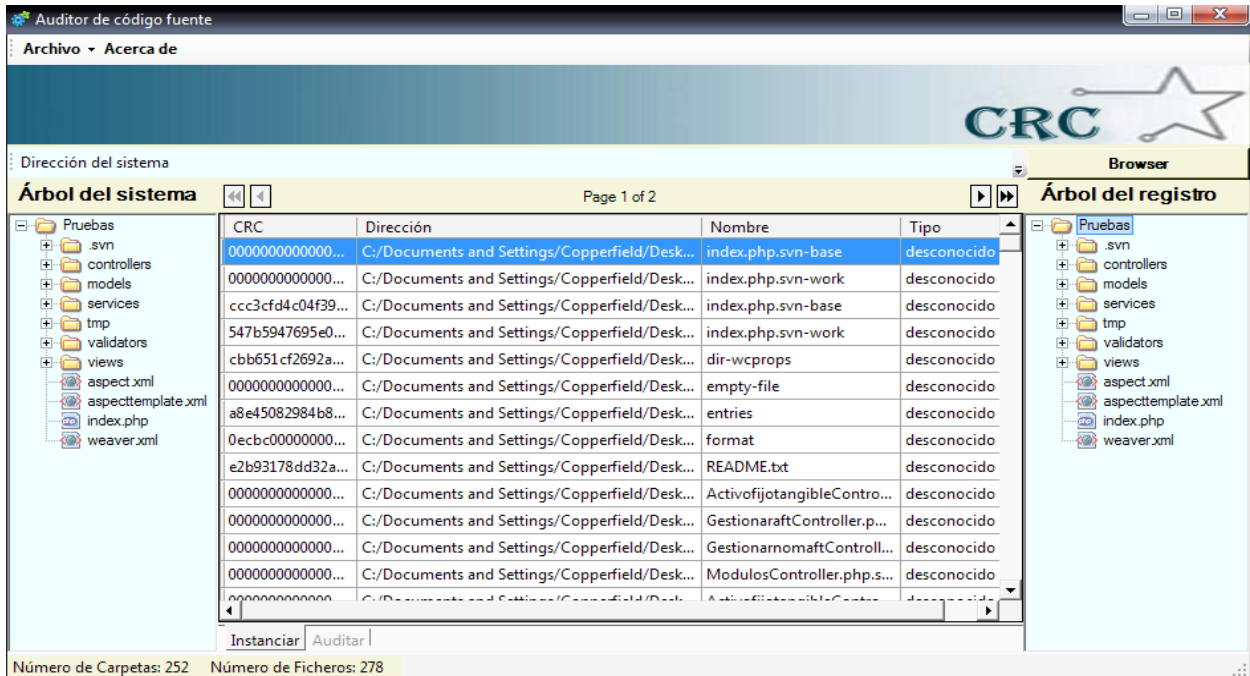


Figura 50: Interfaz que amerita al R4.2.1 Instanciar.

Fuente: Vargas D. y Hernández M. 2009.

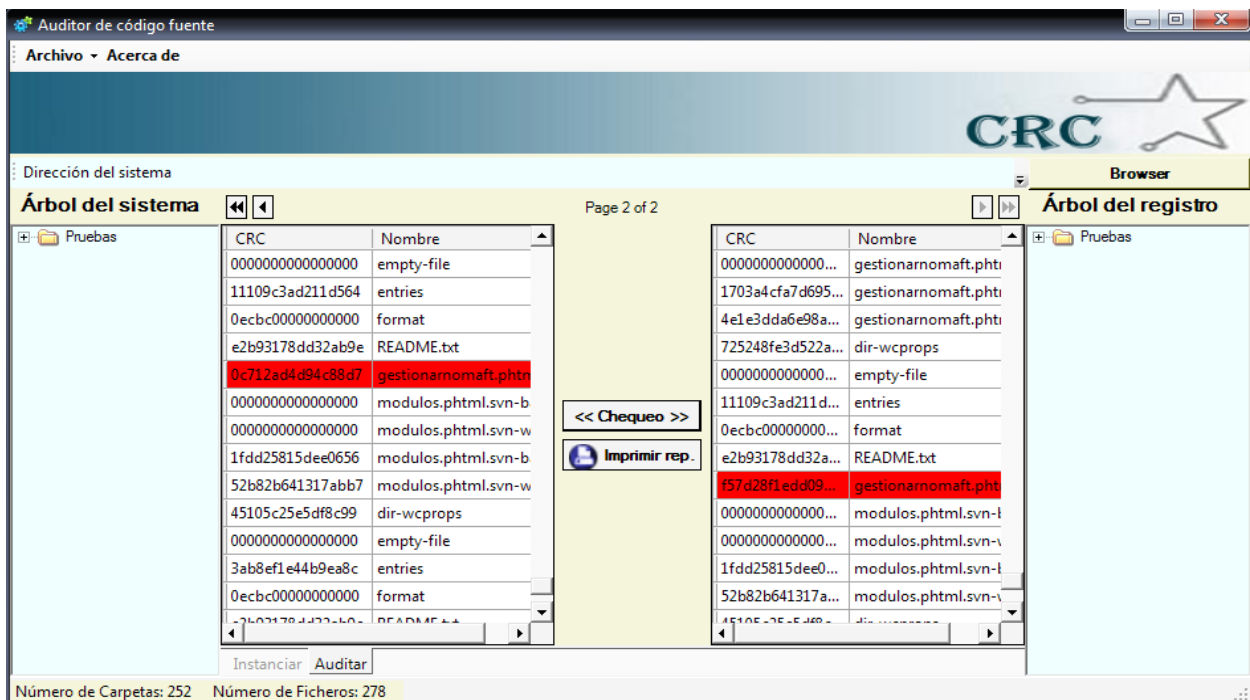


Figura 51: Interfaz que amerita a los R4.2.2 Auditar y R4.2.3 Chequear.

Fuente: Vargas D. y Hernández M. 2009.

Anexo 7: Prototipos de interfaz que ameritan a los requisitos funcionales del componente Generador de inversión de control.

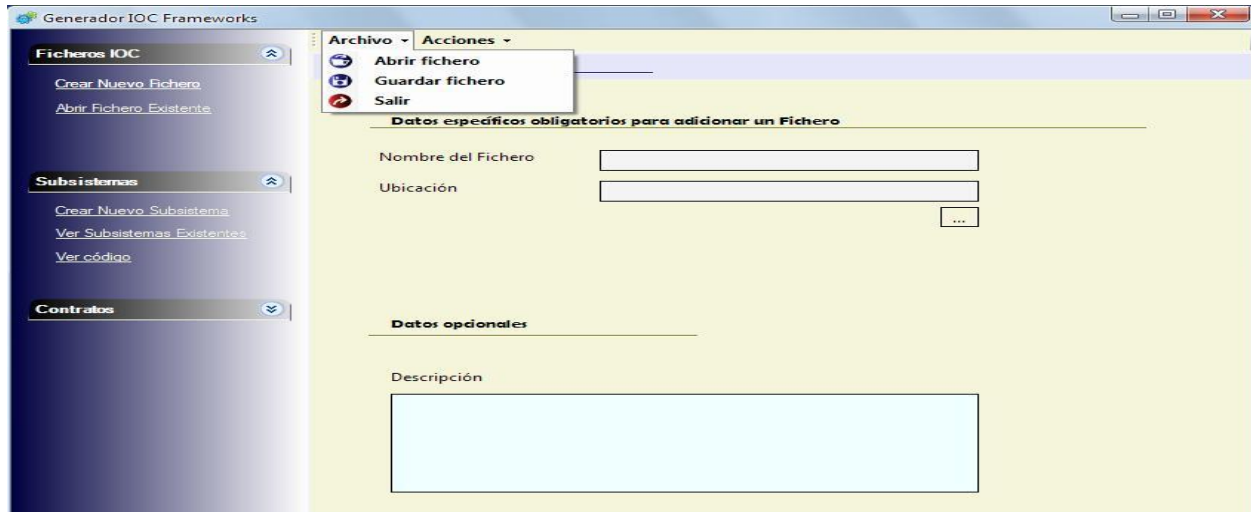


Figura 52: Interfaz que amerita al paquete de R. F. Gestionar fichero.

Fuente: Vargas D. y Hernández M. 2009.

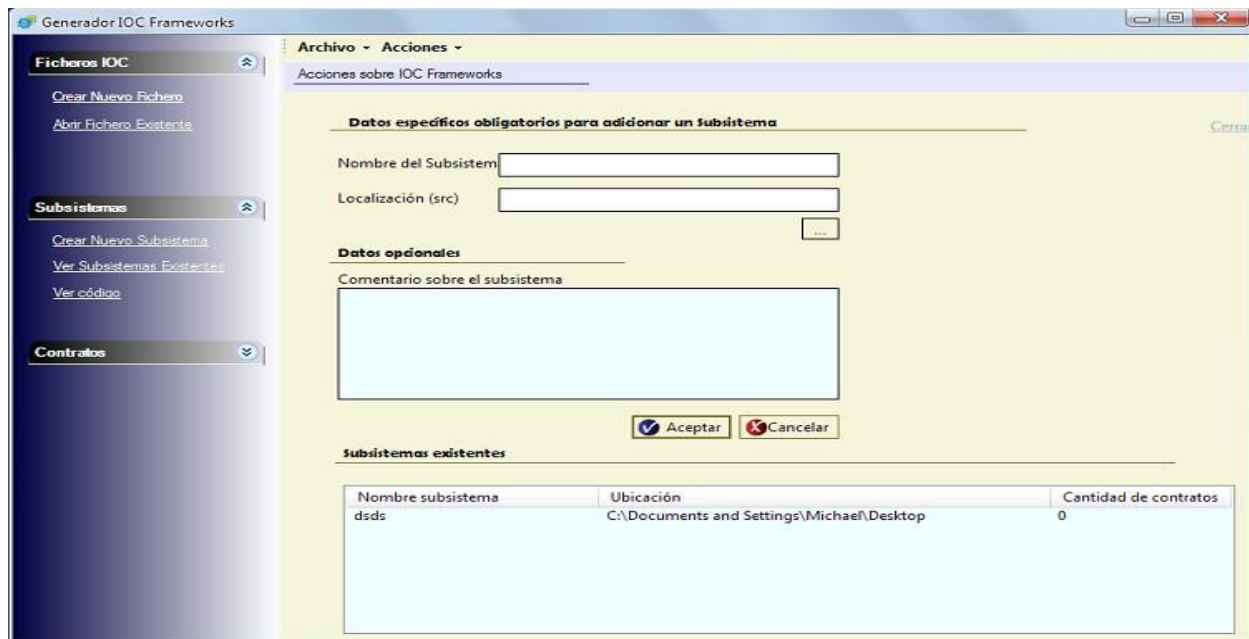


Figura 53: Interfaz que amerita al CRUD del R3.2 Gestionar subsistema.

Fuente: Vargas D. y Hernández M. 2009.

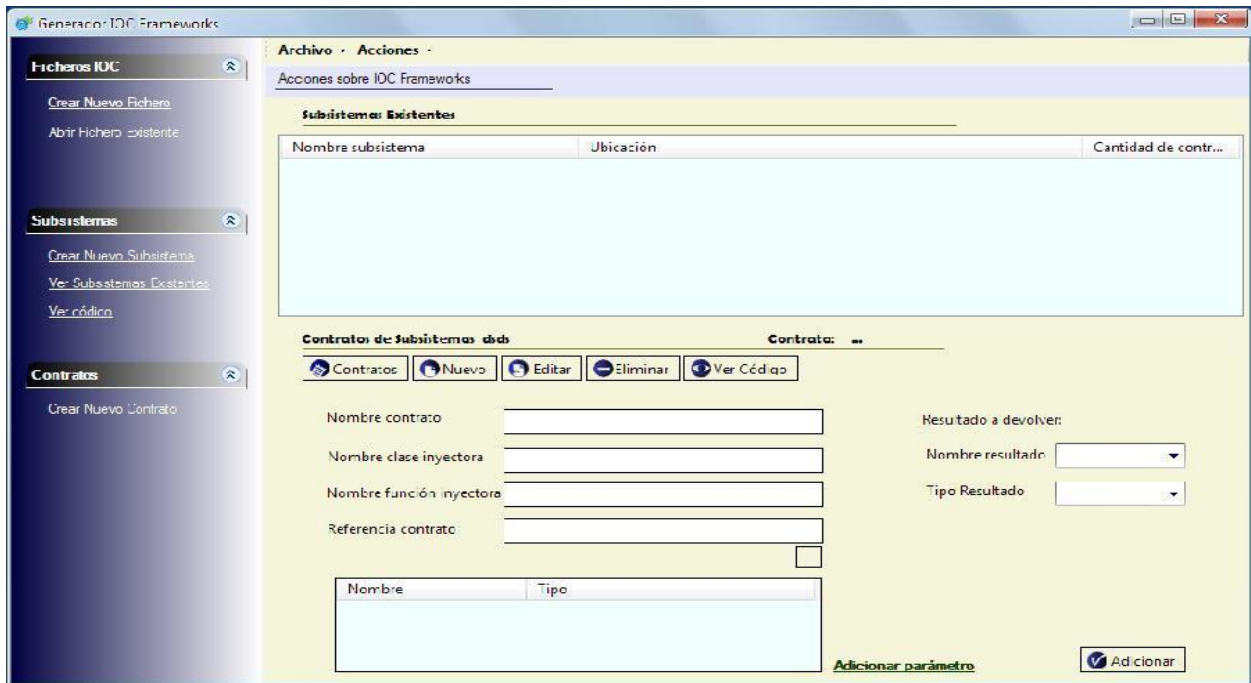


Figura 54: Interfaz que amerita al CRUD R3.3 Gestionar contrato.

Fuente: Vargas D. y Hernández M. 2009.

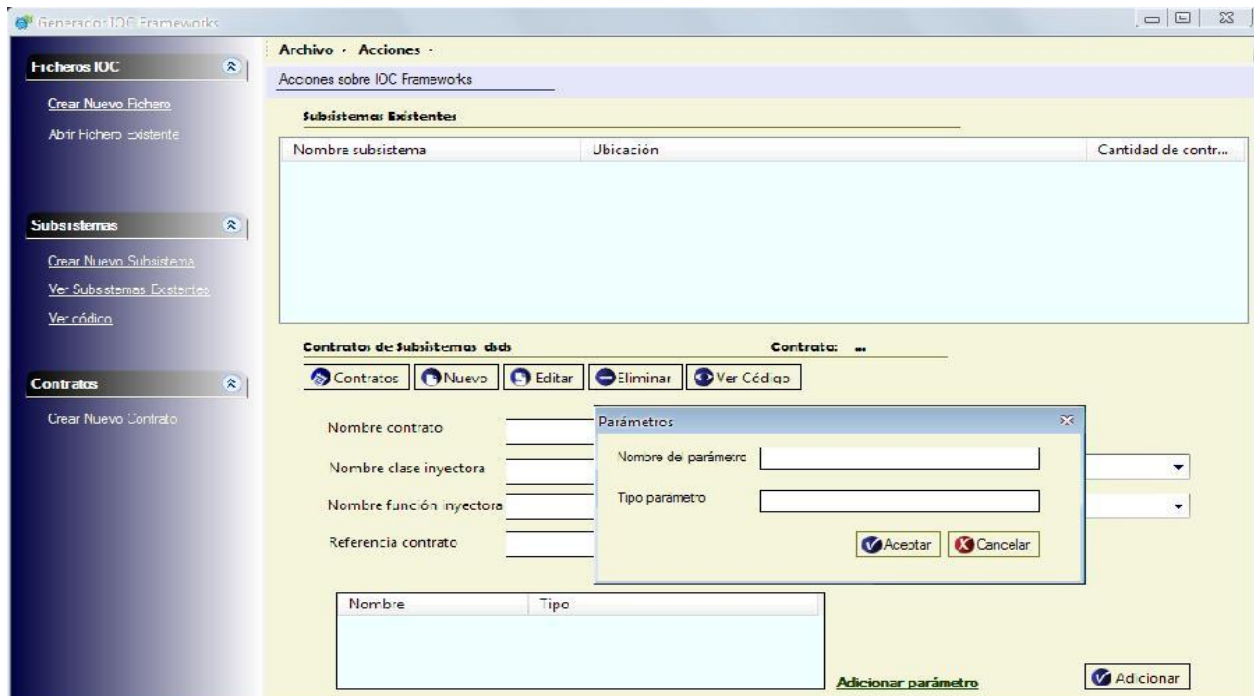


Figura 55: Interfaz que amerita al CRUD 3.3.1.1 Gestionar prototipo.

Fuente: Vargas D. y Hernández M. 2009.

Anexo 8: Diseño de clases del componente Generador de Aspectos.

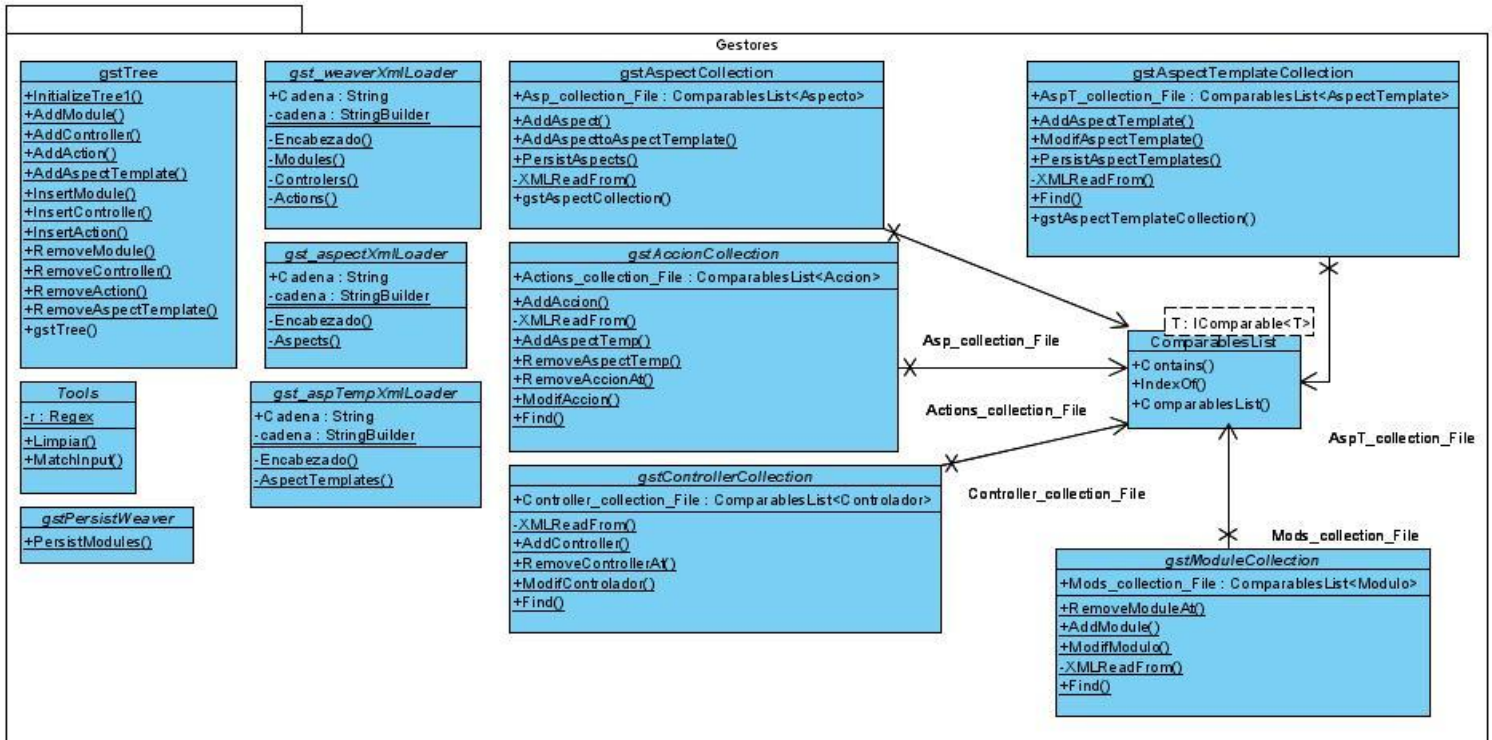


Figura 56: Diseño de clases del paquete de gestores que pertenece al paquete de lógica del componente Generador de Aspectos.

Fuente: Vargas D. y Hernández M. 2009.

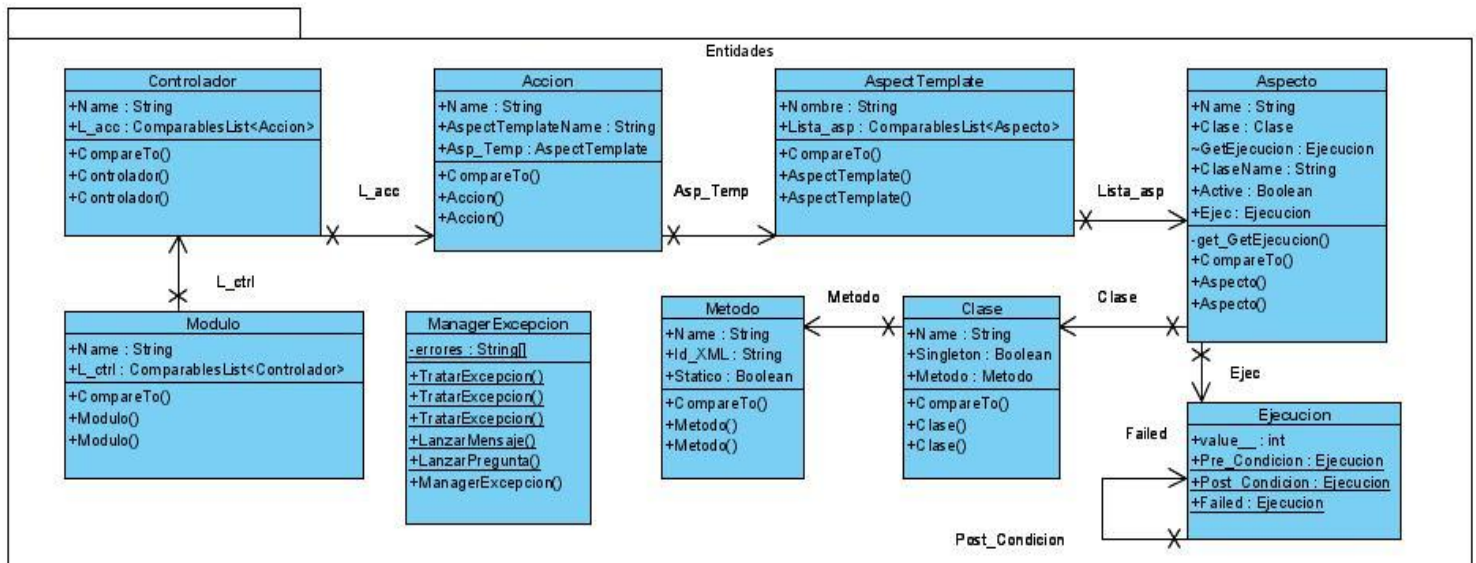


Figura 57: Diseño de clases del paquete de entidades que pertenece al paquete de lógica del componente Generador de Aspectos.

Fuente: Vargas D. y Hernández M. 2009.



Figura 58: Configuración de los paquetes de clases del componente Generador de Aspectos.

Fuente: Vargas D. y Hernández M. 2009.

Anexo 9: Diseño de clases del componente Generador de Excepciones.

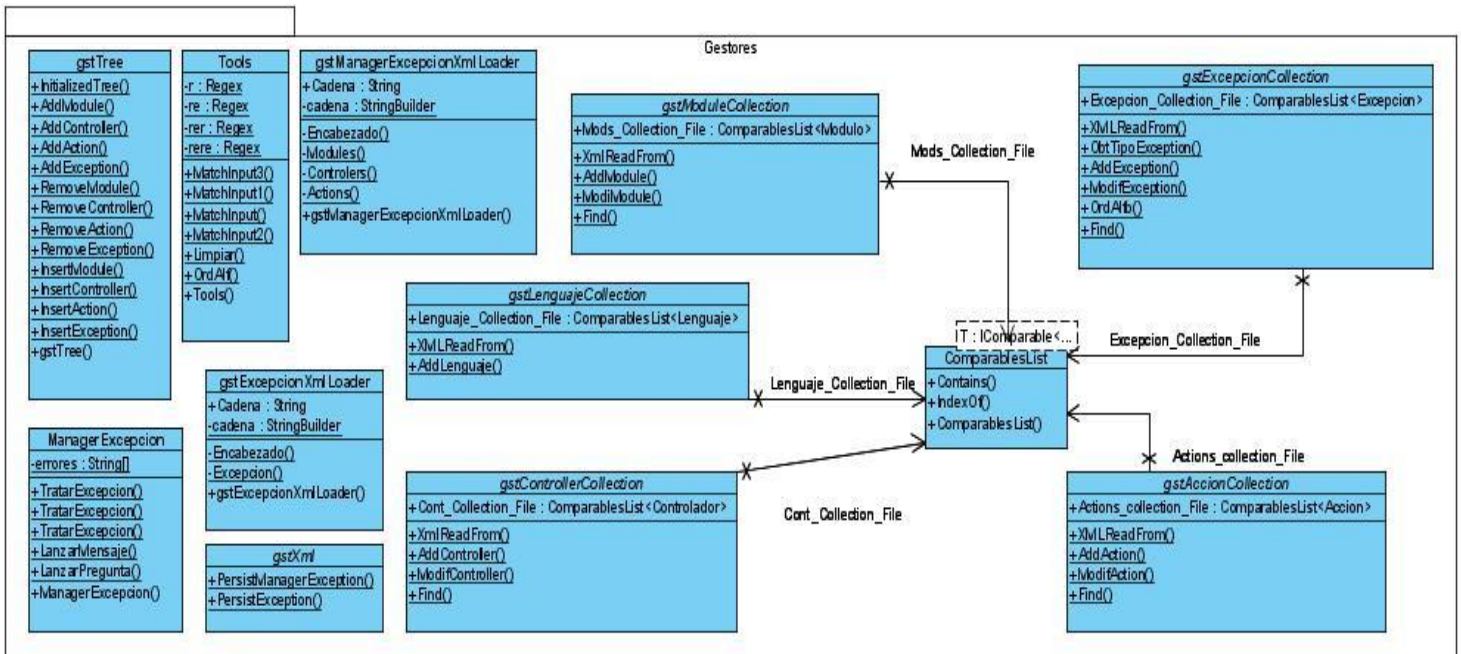


Figura 59: Diseño de clases del paquete de gestores que pertenece al paquete de lógica del componente Generador de Excepciones.

Fuente: Vargas D. y Hernández M. 2009.

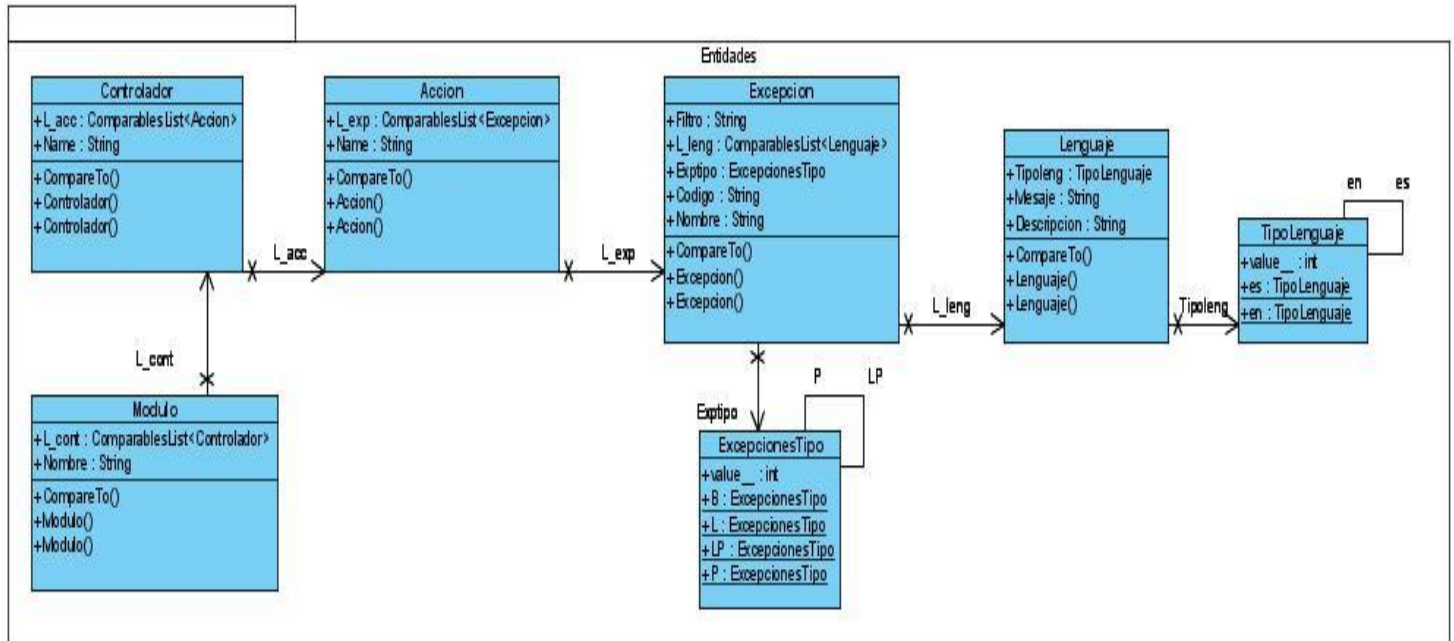


Figura 60: Diseño de clases del paquete de entidades que pertenece al paquete de lógica del componente Generador de Excepciones.

Fuente: Vargas D. y Hernández M. 2009.

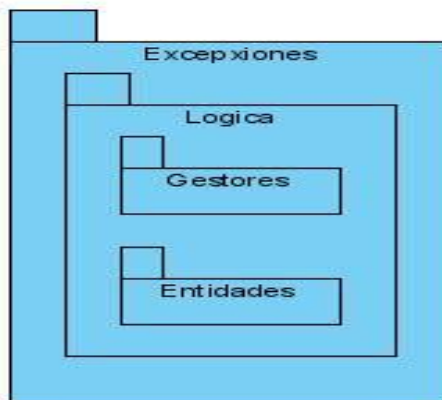


Figura 61: Configuración de los paquetes de clases del componente Generador de Excepciones.

Fuente: Vargas D. y Hernández M. 2009.

Anexo 10: Diseño de clases del componente Auditor de Código Fuente.

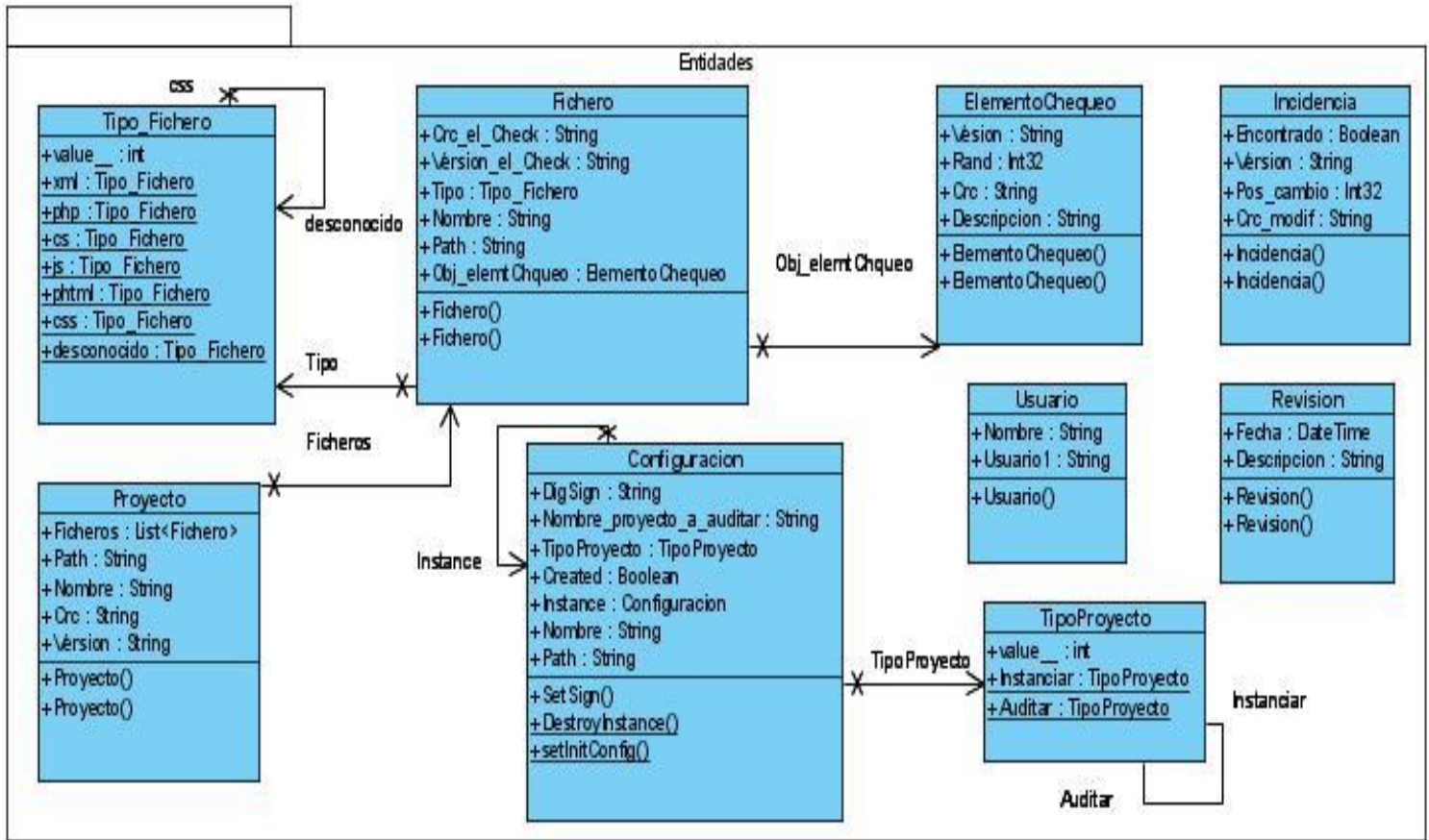


Figura 62: Diseño de clases del paquete de entidades que pertenece al paquete de lógica del componente Auditor de Código Fuente.

Fuente: Vargas D. y Hernández M. 2009.

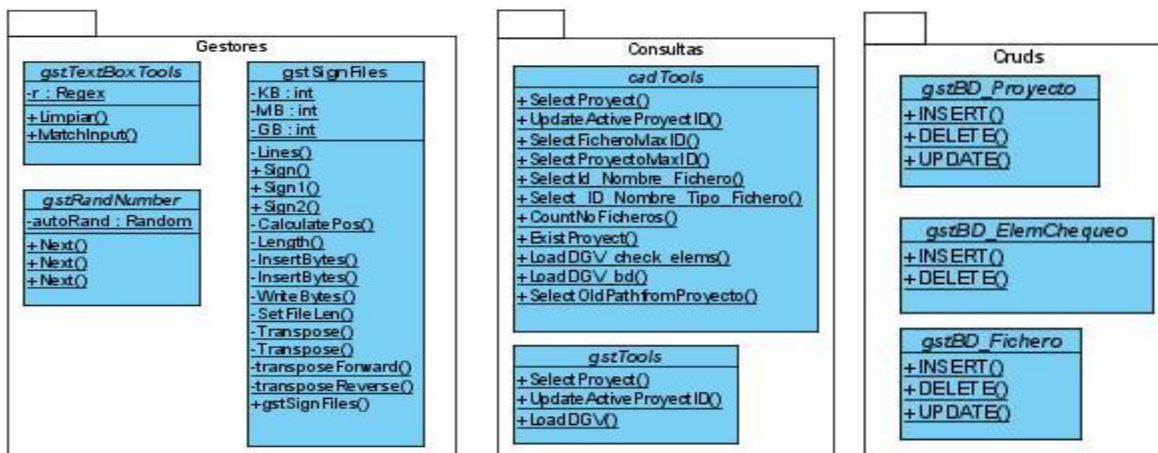


Figura 63: Diseño de clases de los paquetes de gestores que pertenece al paquete de lógica, consultas y CRUD que pertenecen al de CAD (Capa de Acceso a Datos) del componente Auditor de Código Fuente.

Fuente: Vargas D. y Hernández M. 2009.

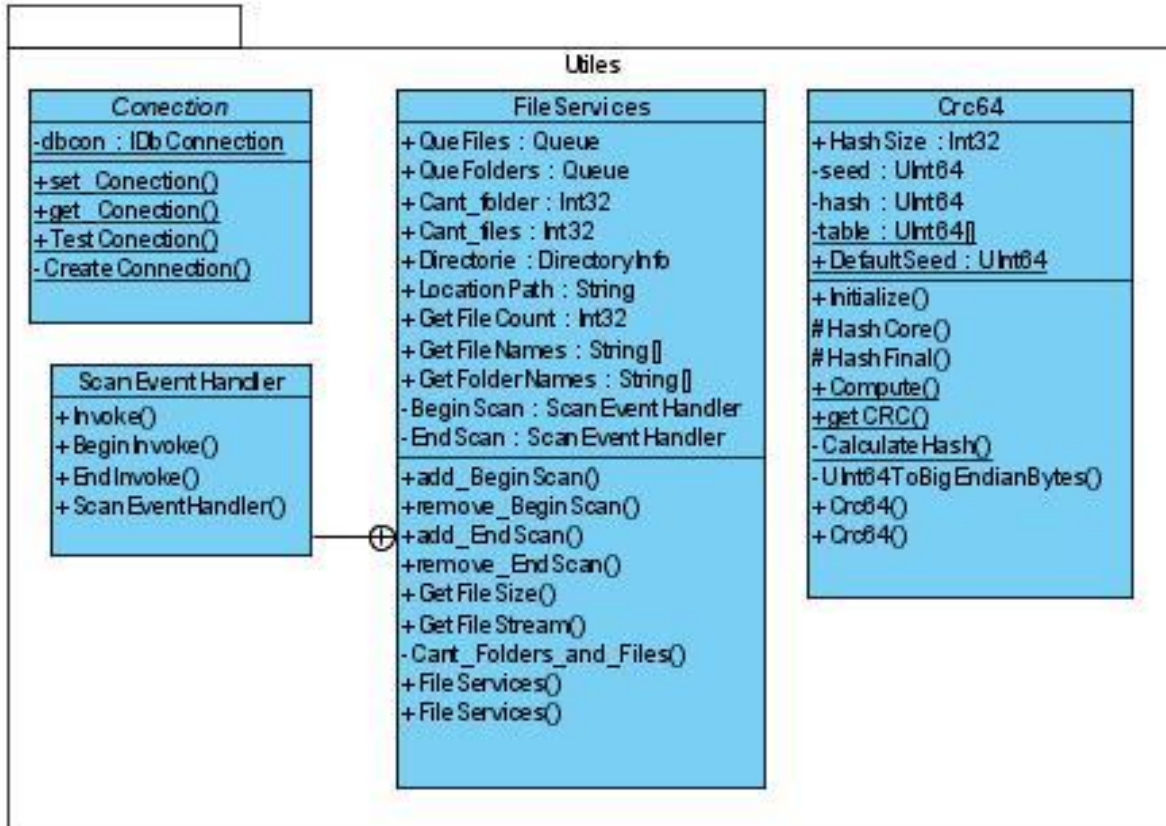


Figura 64: Diseño de clases del paquete útil que pertenece al componente Auditor de Código Fuente.

Fuente: Vargas D. y Hernández M. 2009.

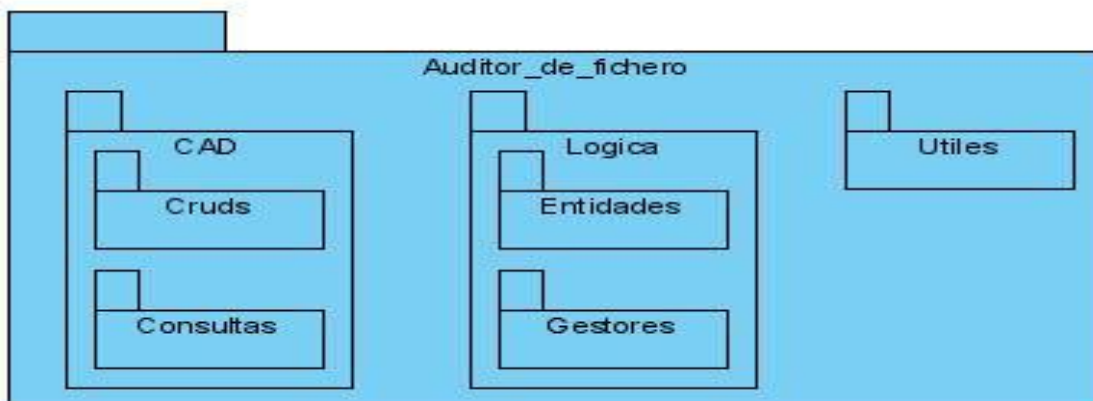


Figura 65: Configuración de los paquetes de clases del componente Auditor de Código Fuente.

Fuente: Vargas D. y Hernández M. 2009.

Anexo 11: Diseño de clases del componente Generador de Inversión de Control.

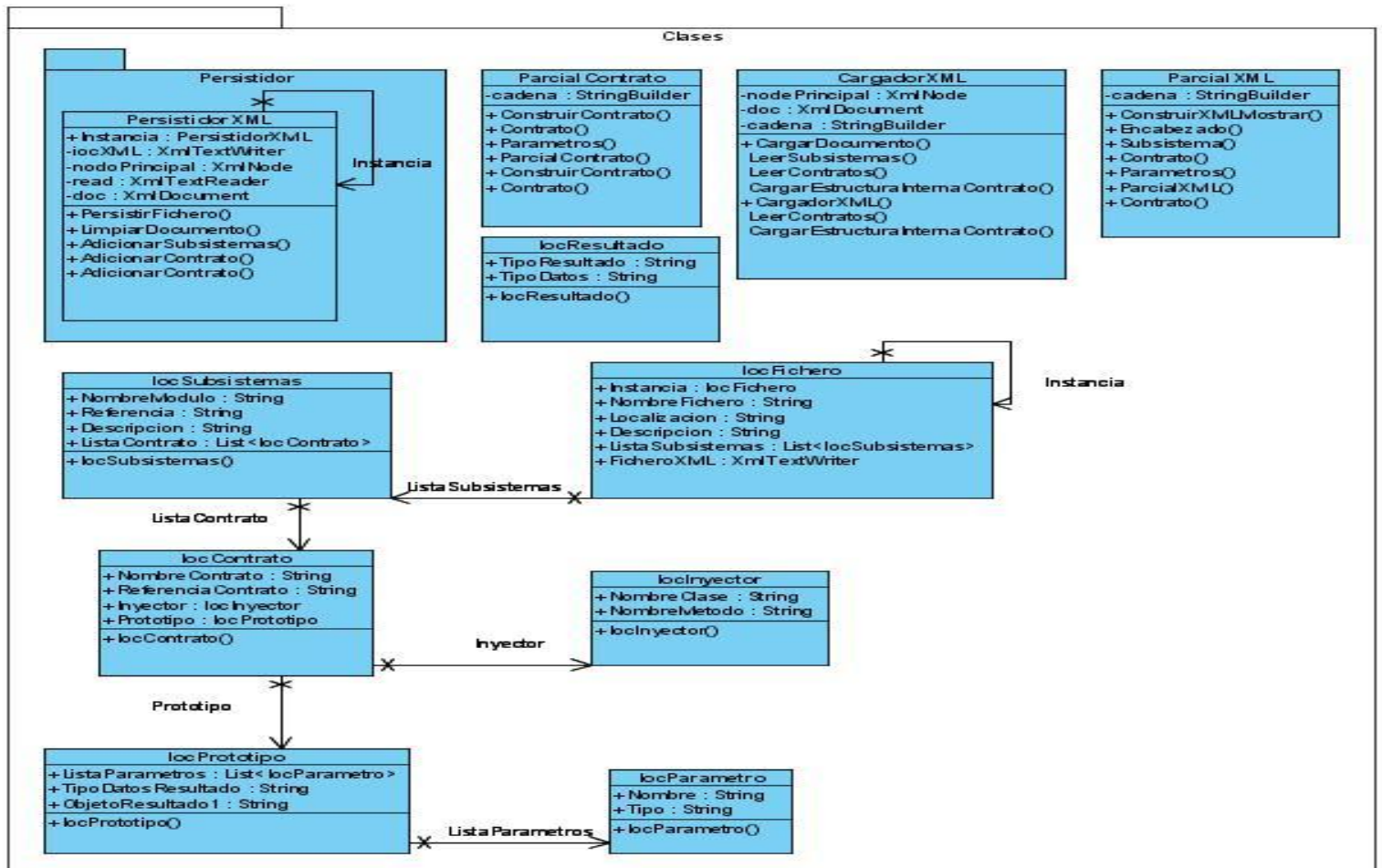


Figura 66: Diseño de clases del paquete persistidor y el paquete clases que pertenece al componente Generador de Inversión de Control.

Fuente: Vargas D. y Hernández M. 2009.

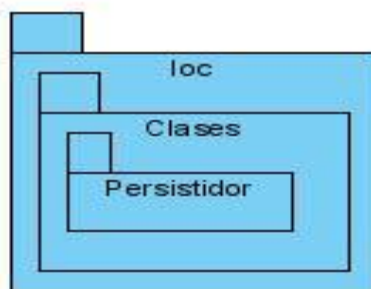


Figura 67: Configuración de los paquetes de clases del componente Generador de Inversión de Control.

Fuente: Vargas D. y Hernández M. 2009.

Anexo 12: Diseño de casos de pruebas del componente Generador de Aspectos.

Tabla 36: DCP de CRUD. del escenario Gestionar módulo perteneciente al componente Generador de aspectos.

DCP Adicionar Módulo

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el módulo IDE, en Generador de aspectos, en la interfaz Módulos.
- El módulo que se desea adicionar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

1: Adicionar módulo.	Se adiciona un nuevo módulo al sistema.	EP 1.1: Adicionar un nuevo módulo al sistema.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación la raíz y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos del módulo que se desea adicionar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Adicionar un módulo dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación la raíz y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos en el formulario de Adicionar módulos dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra la ventana que muestra el mensaje de información
		EP 1.3: Adicionar un módulo introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación la raíz y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos del módulo que se desea adicionar en el formulario introduciendo errores en los datos. - Se presiona el botón Aceptar.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación un módulo y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen o no los datos en el formulario. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar un nuevo módulo al sistema.	V(Módulo)	Para el botón Adicionar: Se adiciona el módulo y se guarda la información adicionada.	
EP 1.2	Adicionar un módulo dejando campos requeridos en blanco.	I(Vacío)	Para el botón Adicionar: Se deja en blanco el campo de texto y sale un mensaje que indica que los datos son incorrectos.	
EP 1.3	Adicionar un módulo introduciendo errores en los datos.	I (/ *- / - ' .)	Para el botón Adicionar El sistema no permite entrar caracteres inválidos y sale un mensaje que indica que los datos son incorrectos.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Eliminar Módulo

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el módulo IDE, en Generador de aspectos, en la interfaz Módulos.

- El módulo que se desea eliminar debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Eliminar módulo.	Se elimina el módulo seleccionado.	EP 1.1: Eliminar módulo.	<ul style="list-style-type: none"> - Se selecciona el módulo que se desea eliminar. - Se presiona el botón Eliminar o se selecciona el módulo en el árbol de navegación y se presiona el clic derecho y se elige la opción de Eliminar. - Se presiona el botón Aceptar del formulario que se presenta.
		EP 1.2: Cancelar.	<ul style="list-style-type: none"> - Se selecciona el módulo que se desea eliminar. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar módulo.	NA.	Se muestra un cartel de confirmación: "¿Está seguro que desea eliminar el módulo?". Se elimina el módulo seleccionado.	
EP 1.2	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DGP Modificar Módulo

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el módulo IDE, en Generador de aspectos, en la interfaz Módulos.
- El módulo que se desea modificar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Modificar módulo.	Se modifica un módulo seleccionado.	EP 1.1: Modificar módulo.	<ul style="list-style-type: none"> - Se selecciona el módulo que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos del módulo que se desea modificar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Modificar módulo dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se selecciona el módulo que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos en el formulario de Modificar módulos dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra el formulario.

		EP 1.3: Modificar módulo introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se selecciona el módulo que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas - Se introducen los datos con errores. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> -Se selecciona el módulo que se va a modificar. -Se presiona el botón izquierdo del mouse dos veces consecutivas. -Se introducen o no los datos en el formulario. Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar módulo.	V(Módulo)	Se modifica el módulo y se guarda la información modificada, se cierra la ventana.	
EP 1.2	Modificar módulo dejando campos requeridos en blanco.	I(Vacío)	Se deja en blanco el campo de texto y sale un mensaje que indica los datos son incorrectos.	
EP 1.3	Modificar módulo introduciendo errores en los datos.	I (/ *- / - ' .)	El sistema no permite entrar caracteres inválidos	

			y se mantiene en la ventana Modificar módulo.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Tabla 37: DCP de CRUD del escenario Gestionar controlador del componente Generador de aspectos.

DCP Adicionar Controlador

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el controlador IDE, en Generador de aspectos, en la interfaz Controlador.
- El controlador que se desea adicionar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Adicionar controlador.	Se adiciona un nuevo controlador al sistema.	EP 1.1: Adicionar un nuevo controlador al sistema.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación un módulo y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos del controlador que se desea adicionar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Adicionar un controlador dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación un módulo y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos en el formulario de Adicionar controlador

			dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra la ventana que muestra el mensaje de información
		EP 1.3: Adicionar un controlador introduciendo errores en los datos.	- Se presiona el botón Adicionar o se elige en el árbol de navegación un módulo y se presiona clic derecho y se elige la opción de Adicionar . - Se introducen los datos del controlador que se desea adicionar en el formulario introduciendo errores en los datos. - Se presiona el botón Aceptar .
		EP 1.4: Cancelar.	- Se presiona el botón Adicionar o se elige en el árbol de navegación un módulo y se presiona clic derecho y se elige la opción de Adicionar . - Se introducen o no los datos en el formulario. - Se presiona el botón Cancelar .

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar un nuevo controlador al sistema.	V(Controlador)	Para el botón Adicionar : Se adiciona el	

			controlador y se guarda la información adicional.	
EP 1.2	Adicionar un controlador dejando campos requeridos en blanco.	I(Vacío)	Para el botón Adicionar : Se deja en blanco el campo de texto y sale un mensaje que indica que los datos son incorrectos.	
EP 1.3	Adicionar un controlador introduciendo errores en los datos.	I (/ *- / - ' .)	Para el botón Adicionar El sistema no permite entrar caracteres inválidos y sale un mensaje que indica que los datos son incorrectos.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Eliminar Controlador

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el controlador IDE, en Generador de aspectos, en la interfaz Controlador.
- El controlador que se desea eliminar debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

1: Eliminar controlador.	Se elimina el controlador seleccionado.	EP 1.1: Eliminar controlador.	<ul style="list-style-type: none"> - Se selecciona el controlador que se desea eliminar. - Se presiona el botón Eliminar o se selecciona el controlador en el árbol de navegación y se presiona el clic derecho y se elige la opción de Eliminar. - Se presiona el botón Aceptar del formulario que se presenta.
		EP 1.2: Cancelar.	<ul style="list-style-type: none"> - Se selecciona el controlador que se desea eliminar. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar controlador.	NA.	Se muestra un cartel de confirmación: "¿Está seguro que desea eliminar el controlador?". Se elimina el controlador seleccionado.	
EP 1.2	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Modificar Controlador

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el controlador IDE, en Generador de aspectos, en la interfaz Controlador.
- El controlador que se desea modificar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Modificar controlador.	Se modifica un controlador seleccionado.	EP 1.1: Modificar controlador.	<ul style="list-style-type: none"> - Se selecciona el controlador que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos del controlador que se desea modificar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Modificar controlador dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se selecciona el controlador que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos en el formulario de Modificar controlador dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra el formulario.

		EP 1.3: Modificar controlador introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se selecciona el controlador que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas - Se introducen los datos con errores. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> -Se selecciona el controlador que se va a modificar. -Se presiona el botón izquierdo del mouse dos veces consecutivas. -Se introducen o no los datos en el formulario. Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar controlador.	V(Controlador)	Se modifica el controlador y se guarda la información modificada, se cierra la ventana.	
EP 1.2	Modificar controlador dejando campos requeridos en blanco.	I(Vacío)	Se deja en blanco el campo de texto y sale un mensaje	

			que indica los datos son incorrectos.	
EP 1.3	Modificar controlador introduciendo errores en los datos.	(*-/'.)	El sistema no permite entrar caracteres inválidos y se mantiene en la ventana Modificar controlador.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Tabla 38: DCP de CRUD del escenario Gestionar acción del componente Generador de aspectos.

DCP Adicionar Acción

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en la acción IDE, en Generador de aspectos, en la interfaz Acción.
- La acción que se desea adicionar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Adicionar acción.	Se adiciona una nueva acción al sistema.	EP 1.1: Adicionar un nuevo acción al sistema.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación un controlador y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos de la acción que se desea adicionar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Adicionar un acción dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación un controlador y se presiona

			<p>clic derecho y se elige la opción de Adicionar.</p> <ul style="list-style-type: none"> - Se introducen los datos en el formulario de Adicionar acción dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra la ventana que muestra el mensaje de información
		EP 1.3: Adicionar un acción introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación un controlador y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos de la acción que se desea adicionar en el formulario introduciendo errores en los datos. - Se presiona el botón Aceptar.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación un controlador y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen o no los datos en el formulario. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del	Resultado de la

			sistema	prueba
EP 1.1	Adicionar una nueva acción al sistema.	V(Acción)	Para el botón Adicionar : Se adiciona la acción y se guarda la información adicionada.	
EP 1.2	Adicionar una acción dejando campos requeridos en blanco.	I(Vacío)	Para el botón Adicionar : Se deja en blanco el campo de texto y sale un mensaje que indica que los datos son incorrectos.	
EP 1.3	Adicionar una acción introduciendo errores en los datos.	I (/ * - / ' .)	Para el botón Adicionar El sistema no permite entrar caracteres inválidos y sale un mensaje que indica que los datos son incorrectos.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DGP Eliminar Acción

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en la acción IDE, en Generador de aspectos, en la interfaz Acción.
- La acción que se desea eliminar debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

1: Eliminar acción.	Se elimina la acción seleccionada.	EP 1.1: Eliminar acción.	<ul style="list-style-type: none"> - Se selecciona la acción que se desea eliminar. - Se presiona el botón Eliminar o se selecciona la acción en el árbol de navegación y se presiona el clic derecho y se elige la opción de Eliminar. - Se presiona el botón Aceptar del formulario que se presenta.
		EP 1.2: Cancelar.	<ul style="list-style-type: none"> - Se selecciona la acción que se desea eliminar. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar acción.	NA.	Se muestra un cartel de confirmación: "¿Está seguro que desea eliminar la acción?". Se elimina la acción seleccionada.	
EP 1.2	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DGP Modificar Acción

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en la acción IDE, en Generador de aspectos, en la interfaz Acción.
- La acción que se desea modificar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Modificar acción.	Se modifica una acción seleccionada.	EP 1.1: Modificar acción.	<ul style="list-style-type: none"> - Se selecciona la acción que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos de la acción que se desea modificar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Modificar acción dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se selecciona la acción que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos en el formulario de Modificar acción dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.3: Modificar acción introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se selecciona la acción que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas - Se introducen los datos con errores. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> -Se selecciona la acción que se va a modificar. -Se presiona el botón izquierdo del mouse dos

			veces consecutivas. -Se introducen o no los datos en el formulario. Se presiona el botón Cancelar .
--	--	--	------------------------------------------------------------------------------------------------------------------

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar acción.	V(Acción)	Se modifica la acción y se guarda la información modificada, se cierra la ventana.	
EP 1.2	Modificar acción dejando campos requeridos en blanco.	I(Vacío)	Se deja en blanco el campo de texto y sale un mensaje que indica los datos son incorrectos.	
EP 1.3	Modificar acción introduciendo errores en los datos.	I (/ *- / - ' .)	El sistema no permite entrar caracteres inválidos y se mantiene en la ventana Modificar acción.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Tabla 39: DCP de CRUD del escenario Gestionar plantilla de aspectos del componente Generador de aspectos.

DCP Adicionar Plantilla de aspectos

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el plantilla de aspectos IDE, en Generador de aspectos, en la interfaz Plantilla de aspectos.
- El plantilla de aspectos que se desea adicionar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Adicionar plantilla de aspectos.	Se adiciona una nueva plantilla de aspectos al sistema.	EP 1.1: Adicionar una nueva plantilla de aspectos al sistema.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación una acción y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos de la plantilla de aspectos que se desea adicionar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Adicionar una plantilla de aspectos dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación una acción y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos en el formulario de Adicionar plantilla de aspectos dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra la ventana que muestra el mensaje de información

		EP 1.3: Adicionar una plantilla de aspectos introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación una acción y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos de la plantilla de aspectos que se desea adicionar en el formulario introduciendo errores en los datos. - Se presiona el botón Aceptar.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación una acción y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen o no los datos en el formulario. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar una nueva plantilla de aspectos al sistema.	V(Plantilla de aspectos)	Para el botón Adicionar : Se adiciona la plantilla de aspectos y se guarda la información adicionada.	
EP 1.2	Adicionar una plantilla de aspectos	I(Vacío)	Para el botón Adicionar : Se deja	

	dejando campos requeridos en blanco.		en blanco el campo de texto y sale un mensaje que indica que los datos son incorrectos.	
EP 1.3	Adicionar una plantilla de aspectos introduciendo errores en los datos.	(*-/-'.)	Para el botón Adicionar El sistema no permite entrar caracteres inválidos y sale un mensaje que indica que los datos son incorrectos.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Eliminar Plantilla de aspectos

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en la plantilla de aspectos IDE, en Generador de aspectos, en la interfaz Plantilla de aspectos.
- La plantilla de aspectos que se desea eliminar debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Eliminar plantilla de aspectos.	Se elimina la plantilla de aspectos seleccionada.	EP 1.1: Eliminar plantilla de aspectos.	<ul style="list-style-type: none"> - Se selecciona la plantilla de aspectos que se desea eliminar. - Se presiona el botón Eliminar o se selecciona la plantilla de aspectos en el árbol de navegación y se presiona el clic derecho y se elige la opción de Eliminar. - Se presiona el botón Aceptar del formulario que se presenta.
		EP 1.2: Cancelar.	- Se selecciona la plantilla

			de aspectos que se desea eliminar. - Se presiona el botón Cancelar .
--	--	--	--------------------------------------------------------------------------------

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar plantilla de aspectos.	NA.	Se muestra un cartel de confirmación: "¿Está seguro que desea eliminar la plantilla de aspectos?". Se elimina la plantilla de aspectos seleccionada.	
EP 1.2	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Modificar Plantilla de aspectos

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en la plantilla de aspectos IDE, en Generador de aspectos, en la interfaz Plantilla de aspectos.
- La plantilla de aspectos que se desea modificar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Modificar plantilla de aspectos.	Se modifica una plantilla de aspectos seleccionada.	EP 1.1: Modificar plantilla de aspectos.	<ul style="list-style-type: none"> - Se selecciona la plantilla de aspectos que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos de la plantilla de aspectos que se desea modificar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Modificar plantilla de aspectos dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se selecciona la plantilla de aspectos que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos en el formulario de Modificar plantilla de aspectos dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.3: Modificar plantilla de aspectos introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se selecciona la plantilla de aspectos que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas - Se introducen los datos con errores. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> -Se selecciona la plantilla de aspectos que se va a modificar. -Se presiona el botón izquierdo del mouse dos veces consecutivas. -Se introducen o no los datos en el formulario. Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar plantilla de aspectos.	V(Plantilla de aspectos)	Se modifica la plantilla de aspectos y se guarda la información modificada, se cierra la ventana.	
EP 1.2	Modificar plantilla de aspectos dejando campos requeridos en blanco.	I(Vacío)	Se deja en blanco el campo de texto y sale un mensaje que indica los datos son incorrectos.	
EP 1.3	Modificar plantilla de aspectos introduciendo errores en los datos.	I (/ *- / - ' .)	El sistema no permite entrar caracteres inválidos y se mantiene en la ventana Modificar plantilla de aspectos.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Tabla 40: DCP de CRUD del escenario Gestionar aspecto del componente Generador de aspectos.

DCP Adicionar Aspecto

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.

- Se debe encontrar en aspecto IDE, en Generador de aspectos, en la interfaz de Aspecto.
- El aspecto que se desea adicionar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Adicionar aspecto.	Se adiciona un nuevo aspecto al sistema.	EP 1.1: Adicionar un nuevo aspecto al sistema.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación una plantilla de aspectos y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos del aspecto que se desea adicionar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Adicionar un aspecto dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación una plantilla de aspecto y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos en el formulario de Adicionar aspectos dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra la ventana que muestra el mensaje de información

		EP 1.3: Adicionar un aspecto introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación una plantilla de aspectos y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos del aspecto que se desea adicionar en el formulario introduciendo errores en los datos. - Se presiona el botón Aceptar.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación una plantilla de aspectos y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen o no los datos en el formulario. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.
Ejecución	Campo de opciones	NA.	Pre-Condición, Post-Condición, Fallo.
Nombre de clase	Campo de texto	No	Combinación de letras y números.
Singleton	Campo de chequeo	NA.	Si o No.
Nombre de método	Campo de texto	No	Combinación de letras y números.
Id XML	Campo de texto	No	Combinación de letras y números.
Estático	Campo de chequeo	NA.	Si o No.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Ejecución	Nombre de clase	Singlet on	Nombre de método	Id XML	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar un nuevo aspecto al sistema.	V(Aspecto)	V(Seleccionado)	V(Clase)	V(Chequeado)	V(Método)	V(Id)	Para el botón Adicionar : Se adiciona el aspecto y se guarda la información adicionada.	
EP 1.2	Adicionar un aspecto dejando campos requeridos en blanco.	I(Vacío)	NA.	I(Vacío)	NA.	I(Vacío)	I(Vacío)	Para el botón Adicionar : Se deja en blanco el campo de texto y sale un mensaje que indica que los datos son incorrectos.	
EP 1.3	Adicionar un aspecto introduciendo errores en los datos.	I (/ *- / - ' .)	NA.	I (/ *- / - ' .)	NA.	I (/ *- / - ' .)	I (/ *- / - ' .)	Para el botón Adicionar El sistema no permite entrar caracteres inválidos y sale un mensaje que indica que los datos son incorrectos.	
EP 1.4	Cancelar.	NA.	NA.	NA.	NA.	NA.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Eliminar Aspecto

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el aspecto IDE, en Generador de aspectos, en la interfaz Aspectos.

- El aspecto que se desea eliminar debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Eliminar aspecto.	Se elimina el aspecto seleccionado.	EP 1.1: Eliminar aspecto.	- Se selecciona el aspecto que se desea eliminar. - Se presiona el botón Eliminar . - Se presiona el botón Aceptar del formulario que se presenta.
		EP 1.2: Cancelar.	- Se selecciona el aspecto que se desea eliminar. - Se presiona el botón Cancelar .

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar aspectos.	NA.	Se muestra un cartel de confirmación: "¿Está seguro que desea eliminar el aspectos?". Se elimina el aspecto seleccionado.	
EP 1.2	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Modificar Aspecto

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el aspecto IDE, en Generador de aspectos, en la interfaz Aspectos.
- El aspecto que se desea modificar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Modificar aspecto.	Se modifica un aspecto seleccionado.	EP 1.1: Modificar aspecto.	<ul style="list-style-type: none"> - Se selecciona el aspecto que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos del aspecto que se desea modificar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Modificar aspecto dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se selecciona el aspecto que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos en el formulario de Modificar aspecto dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.3: Modificar aspecto introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se selecciona el aspecto que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas - Se introducen los datos con errores. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.4: Cancelar.	-Se selecciona el aspecto que se va a modificar.

			-Se presiona el botón izquierdo del mouse dos veces consecutivas. -Se introducen o no los datos en el formulario. Se presiona el botón Cancelar .
--	--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.
Ejecución	Campo de opciones	NA.	Pre-Condición, Post-Condición, Fallo.
Nombre de clase	Campo de texto	No	Combinación de letras y números.
Singleton	Campo de chequeo	NA.	Si o No.
Nombre de método	Campo de texto	No	Combinación de letras y números.
Id XML	Campo de texto	No	Combinación de letras y números.
Estático	Campo de chequeo	NA.	Si o No.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Ejecución	Nombre de clase	Singleton	Nombre de método	Id XML	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar aspectos.	V(Aspecto)	V(Seleccionado)	V(Clase)	V(Chequeado)	V(Método)	V(Id)	Se modifica el aspecto y se guarda la información modificada, se cierra la ventana.	
EP 1.2	Modificar aspectos dejando campos requeridos	I(Vacío)	NA.	I(Vacío)	NA.	I(Vacío)	I(Vacío)	Se deja en blanco el campo de texto y sale un mensaje	

	en blanco.							que indica los datos son incorrectos.	
EP 1.3	Modificar aspectos introduciendo errores en los datos.	I (/ *- / - ' .)	NA.	I (/ *- / - ' .)	NA.	I (/ *- / - ' .)	I (/ *- / - ' .)	El sistema no permite entrar caracteres inválidos y se mantiene en la ventana Modificar aspectos.	
EP 1.4	Cancelar.	NA.	NA.	NA.	NA.	NA.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Anexo 13: Diseño de casos de pruebas del componente Generador de Excepciones.

- Generador de Excepciones.

Tabla 41: Paquete de DCP del escenario Gestionar ficheros perteneciente al componente generador de excepciones.

DCP Crear Ficheros.

1. Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el componente IDE, en Generador de excepciones.
- El fichero que desea crear debe ubicarlo en una carpeta que no tenga implícita ningún fichero del mismo tipo.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Crear ficheros.	Se crean nuevos ficheros al sistema.	EP 1.1: Crear ficheros al sistema.	<ul style="list-style-type: none"> - Se presiona el botón superior izquierdo del menú. - Se elige la opción

			<p>Crear del menú.</p> <ul style="list-style-type: none"> - Se elije la ubicación donde se van a crear los ficheros. - Se presiona el botón Aceptar.
		EP 1.2: Crear ficheros dejando campo requeridos en blanco.	NA.
		EP 1.3: Crear ficheros introduciendo errores en los datos.	NA.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se presiona el botón superior izquierdo del menú. - Se elije la opción Crear del menú. - Se elije o no la ubicación donde se van a crear los ficheros. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Crear ficheros al sistema.	NA.	Para la opción de Crear : Se crea o busca la ubicación donde se van a crear los ficheros, se crean y se cierra la ventana.	
EP 1.2	Crear ficheros dejando campo	NA.	NA.	

	requeridos en blanco.			
EP 1.3	Crear ficheros introduciendo errores en los datos.	NA.	NA.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Cargar Ficheros.

1. Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el componente IDE, en Generador de excepciones.
- El fichero que desea cargar debe existir en la carpeta de la ubicación elegida.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Cargar ficheros.	Se cargan ficheros existentes al sistema.	EP 1.1: Cargar ficheros al sistema.	- Se presiona el botón superior izquierdo del menú. - Se elige la opción Cargar del menú. - Se elige la ubicación donde están ubicados los ficheros existentes. - Se presiona el botón Aceptar .
		EP 1.2: Cargar ficheros dejando campo requeridos en blanco.	NA.
		EP 1.3: Cargar ficheros introduciendo errores en los datos.	NA.
		EP 1.4: Cancelar.	- Se presiona el botón superior izquierdo del menú. - Se elige la opción Cargar del menú. - Se elige o no la ubicación de donde se va a cargar los ficheros.

			- Se presiona el botón Cancelar .
--	--	--	------------------------------------------

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Cargar ficheros existentes al sistema.	NA.	Para la opción de Cargar : Se busca la ubicación donde se encuentran los ficheros existentes, se cargar y se cierra la ventana.	
EP 1.2	Cargar ficheros dejando campo requeridos en blanco.	NA.	NA.	
EP 1.3	Cargar ficheros introduciendo errores en los datos.	NA.	NA.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DGP Guardar Ficheros.

1. Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el componente IDE, en Generador de excepciones.

➤ Se debe de haber cargado o creado los ficheros.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Guardar ficheros.	Se guarda los ficheros del sistema.	EP 1.1: Guardar ficheros del sistema.	- Se presiona el botón superior izquierdo del menú. - Se elije la opción Guardar del menú. - Se elije la ubicación donde serán ubicados los ficheros. - Se presiona el botón Aceptar .
		EP 1.2: Guardar ficheros dejando campo requeridos en blanco.	NA.
		EP 1.3: Cargar ficheros introduciendo errores en los datos.	NA.
		EP 1.4: Cancelar.	- Se presiona el botón superior izquierdo del menú. - Se elije la opción Guardar del menú. - Se elije o no la ubicación de donde se va a guardar los ficheros. - Se presiona el botón Cancelar .

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba

EP 1.1	Guardar ficheros existentes al sistema.	NA.	Para la opción de Guardar : Se busca la ubicación donde se guardaran los ficheros, se guarda y se cierra la ventana.	
EP 1.2	Guardar ficheros dejando campo requeridos en blanco.	NA.	NA.	
EP 1.3	Guardar ficheros introduciendo errores en los datos.	NA.	NA.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Buscar.

1. Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el componente IDE, en Generador de excepciones.
- Se debe de haber cargado o creado los ficheros.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Buscar.	Se busca algún concepto (Módulo, Controlador, Acción, Excepción).	EP 1.1: Buscar conceptos.	<ul style="list-style-type: none"> - Se presiona clic derecho sobre el árbol de navegación. - Se elije la opción Buscar del menú. - Se elije el concepto a buscar. - Se presiona el botón Buscar.

		EP 1.2: Buscar conceptos dejando campo requeridos en blanco.	<ul style="list-style-type: none"> - Se elije la opción Buscar del menú - Se introduce los datos en el formulario de buscar dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra la ventana que muestra el mensaje de información.
		EP 1.3: Cargar ficheros introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se elije la opción Buscar del menú - Se introduce los datos en el formulario de buscar introduciendo datos con errores. - Se presiona el botón Aceptar o se cierra la ventana que muestra el mensaje de información.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se elije la opción Buscar del menú - Se introducen o no los datos en el formulario. - Se cierra el formulario.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Buscar conceptos.	V(Módulo, Controlador, Acción, Excepción)	Para el botón de Buscar : Se busca el concepto y se muestra la ubicación del mismo en el árbol de navegación	

			conjuntamente con sus nodos hijos.	
EP 1.2	Buscar conceptos dejando campo requeridos en blanco.	I(Vacío)	Para el botón de Buscar : Se deja en blanco el campo de texto y sale un cartel que indica que los datos son incorrectos.	
EP 1.3	Buscar conceptos introduciendo errores en los datos.	I (/ *- / - ' .)	Para el botón de Buscar : Se deja en blanco el campo de texto y sale un cartel que indica que los datos son incorrectos.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Tabla 42: DCP de CRUD del escenario Gestionar módulo del componente Generador de excepciones.

DCP Adicionar Módulo

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el módulo IDE, en Generador de excepciones, en la interfaz Módulos.
- El módulo que se desea adicionar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

<p>1: Adicionar módulo.</p>	<p>Se adiciona un nuevo módulo al sistema.</p>	<p>EP 1.1: Adicionar un nuevo módulo al sistema.</p>	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación la raíz y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos del módulo que se desea adicionar en el formulario. - Se presiona el botón Aceptar.
		<p>EP 1.2: Adicionar un módulo dejando campos requeridos en blanco.</p>	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación la raíz y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos en el formulario de Adicionar módulos dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra la ventana que muestra el mensaje de información
		<p>EP 1.3: Adicionar un módulo introduciendo errores en los datos.</p>	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación la raíz y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos del módulo que se desea adicionar en el formulario introduciendo errores en los datos. - Se presiona el botón Aceptar.
		<p>EP 1.4: Cancelar.</p>	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación un módulo y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen o no los datos en el formulario. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar un nuevo módulo al sistema.	V(Módulo)	Para el botón Adicionar : Se adiciona el módulo y se guarda la información adicionada.	
EP 1.2	Adicionar un módulo dejando campos requeridos en blanco.	I(Vacío)	Para el botón Adicionar : Se deja en blanco el campo de texto y sale un mensaje que indica que los datos son incorrectos.	
EP 1.3	Adicionar un módulo introduciendo errores en los datos.	I (/ *- / - ' .)	Para el botón Adicionar El sistema no permite entrar caracteres inválidos y sale un mensaje que indica que los datos son incorrectos.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Eliminar Módulo

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el módulo IDE, en Generador de excepciones, en la interfaz Módulos.
- El módulo que se desea eliminar debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Eliminar módulo.	Se elimina el módulo seleccionado.	EP 1.1: Eliminar módulo.	- Se selecciona el módulo que se desea eliminar. - Se presiona el botón Eliminar o se selecciona el módulo en el árbol de navegación y se presiona el clic derecho y se elige la opción de Eliminar . - Se presiona el botón Aceptar del formulario que se presenta.
		EP 1.2: Cancelar.	- Se selecciona el módulo que se desea eliminar. - Se presiona el botón Cancelar .

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar módulo.	NA.	Se muestra un cartel de confirmación: "¿Está seguro que desea eliminar el	

			módulo?". Se elimina el módulo seleccionado.	
EP 1.2	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DGP Modificar Módulo

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el módulo IDE, en Generador de excepciones, en la interfaz Módulos.
- El módulo que se desea modificar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Modificar módulo.	Se modifica un módulo seleccionado.	EP 1.1: Modificar módulo.	<ul style="list-style-type: none"> - Se selecciona el módulo que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos del módulo que se desea modificar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Modificar módulo dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se selecciona el módulo que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos en el formulario de Modificar módulos dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra el formulario.

		EP 1.3: Modificar módulo introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se selecciona el módulo que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas - Se introducen los datos con errores. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> -Se selecciona el módulo que se va a modificar. -Se presiona el botón izquierdo del mouse dos veces consecutivas. -Se introducen o no los datos en el formulario. Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar módulo.	V(Módulo)	Se modifica el módulo y se guarda la información modificada, se cierra la ventana.	
EP 1.2	Modificar módulo dejando campos requeridos en blanco.	I(Vacío)	Se deja en blanco el campo de texto y sale un mensaje que indica los datos son incorrectos.	
EP 1.3	Modificar módulo	I (/ *- / - ' .)	El sistema no	

	introduciendo errores en los datos.		permite entrar caracteres inválidos y se mantiene en la ventana Modificar módulo.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Tabla 43: DCP de CRUD del escenario Gestionar controlador del componente Generador de excepciones.

DCP Adicionar Controlador

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el controlador IDE, en Generador de excepciones, en la interfaz Controlador.
- El controlador que se desea adicionar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Adicionar controlador.	Se adiciona un nuevo controlador al sistema.	EP 1.1: Adicionar un nuevo controlador al sistema.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación un módulo y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos del controlador que se desea adicionar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Adicionar un controlador dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación un módulo y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos en el formulario de

			Adicionar controlador dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra la ventana que muestra el mensaje de información
		EP 1.3: Adicionar un controlador introduciendo errores en los datos.	- Se presiona el botón Adicionar o se elige en el árbol de navegación un módulo y se presiona clic derecho y se elige la opción de Adicionar . - Se introducen los datos del controlador que se desea adicionar en el formulario introduciendo errores en los datos. - Se presiona el botón Aceptar .
		EP 1.4: Cancelar.	- Se presiona el botón Adicionar o se elige en el árbol de navegación un módulo y se presiona clic derecho y se elige la opción de Adicionar . - Se introducen o no los datos en el formulario. - Se presiona el botón Cancelar .

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar un nuevo controlador al	V(Controlador)	Para el botón Adicionar : Se	

	sistema.		adiciona el controlador y se guarda la información adicionada.	
EP 1.2	Adicionar un controlador dejando campos requeridos en blanco.	I(Vacío)	Para el botón Adicionar : Se deja en blanco el campo de texto y sale un mensaje que indica que los datos son incorrectos.	
EP 1.3	Adicionar un controlador introduciendo errores en los datos.	I (*-/-'.)	Para el botón Adicionar El sistema no permite entrar caracteres inválidos y sale un mensaje que indica que los datos son incorrectos.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Eliminar Controlador

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el controlador IDE, en Generador de excepciones, en la interfaz Controlador.
- El controlador que se desea eliminar debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

1: Eliminar controlador.	Se elimina el controlador seleccionado.	EP 1.1: Eliminar controlador.	<ul style="list-style-type: none"> - Se selecciona el controlador que se desea eliminar. - Se presiona el botón Eliminar o se selecciona el controlador en el árbol de navegación y se presiona el clic derecho y se elige la opción de Eliminar. - Se presiona el botón Aceptar del formulario que se presenta.
		EP 1.2: Cancelar.	<ul style="list-style-type: none"> - Se selecciona el controlador que se desea eliminar. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar controlador.	NA.	Se muestra un cartel de confirmación: "¿Está seguro que desea eliminar el controlador?". Se elimina el controlador seleccionado.	
EP 1.2	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Modificar Controlador

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el controlador IDE, en Generador de excepciones, en la interfaz Controlador.
- El controlador que se desea modificar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Modificar controlador.	Se modifica un controlador seleccionado.	EP 1.1: Modificar controlador.	<ul style="list-style-type: none"> - Se selecciona el controlador que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos del controlador que se desea modificar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Modificar controlador dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se selecciona el controlador que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos en el formulario de Modificar controlador dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra el formulario.

		EP 1.3: Modificar controlador introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se selecciona el controlador que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas - Se introducen los datos con errores. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> -Se selecciona el controlador que se va a modificar. -Se presiona el botón izquierdo del mouse dos veces consecutivas. -Se introducen o no los datos en el formulario. Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar controlador.	V(Controlador)	Se modifica el controlador y se guarda la información modificada, se cierra la ventana.	
EP 1.2	Modificar controlador dejando campos requeridos en blanco.	I(Vacío)	Se deja en blanco el campo de texto y sale un mensaje que indica los datos	

			son incorrectos.	
EP 1.3	Modificar controlador introduciendo errores en los datos.	(/*/-/'.)	El sistema no permite entrar caracteres inválidos y se mantiene en la ventana Modificar controlador.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Tabla 44: DCP de CRUD del escenario Gestionar acción del componente Generador de excepciones.

DCP Adicionar Acción

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en la acción IDE, en Generador de excepciones, en la interfaz Acción.
- La acción que se desea adicionar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Adicionar acción.	Se adiciona una nueva acción al sistema.	EP 1.1: Adicionar un nuevo acción al sistema.	- Se presiona el botón Adicionar o se elige en el árbol de navegación un controlador y se presiona clic derecho y se elige la opción de Adicionar . - Se introducen los datos de la acción que se desea adicionar en el formulario. - Se presiona el botón Aceptar .
		EP 1.2: Adicionar un acción dejando campos requeridos en blanco.	- Se presiona el botón Adicionar o se elige en el árbol de navegación un controlador y se presiona clic derecho y se elige la opción de Adicionar . - Se introducen los datos

			<p>en el formulario de Adicionar acción dejando al menos un campo requerido en blanco.</p> <ul style="list-style-type: none"> - Se presiona el botón Aceptar o se cierra la ventana que muestra el mensaje de información
		EP 1.3: Adicionar un acción introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación un controlador y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos de la acción que se desea adicionar en el formulario introduciendo errores en los datos. - Se presiona el botón Aceptar.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación un controlador y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen o no los datos en el formulario. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar una nueva	V(Acción)	Para el botón	

	acción al sistema.		Adicionar: Se adiciona la acción y se guarda la información adicionada.	
EP 1.2	Adicionar una acción dejando campos requeridos en blanco.	I(Vacío)	Para el botón Adicionar: Se deja en blanco el campo de texto y sale un mensaje que indica que los datos son incorrectos.	
EP 1.3	Adicionar una acción introduciendo errores en los datos.	I (/ * - / - ' .)	Para el botón Adicionar El sistema no permite entrar caracteres inválidos y sale un mensaje que indica que los datos son incorrectos.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Eliminar Acción

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en la acción IDE, en Generador de excepciones, en la interfaz Acción.
- La acción que se desea eliminar debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

1: Eliminar acción.	Se elimina la acción seleccionada.	EP 1.1: Eliminar acción.	<ul style="list-style-type: none"> - Se selecciona la acción que se desea eliminar. - Se presiona el botón Eliminar o se selecciona la acción en el árbol de navegación y se presiona el clic derecho y se elige la opción de Eliminar. - Se presiona el botón Aceptar del formulario que se presenta.
		EP 1.2: Cancelar.	<ul style="list-style-type: none"> - Se selecciona la acción que se desea eliminar. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar acción.	NA.	Se muestra un cartel de confirmación: "¿Está seguro que desea eliminar la acción?". Se elimina la acción seleccionada.	
EP 1.2	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Modificar Acción

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en la acción IDE, en Generador de excepciones, en la interfaz Acción.
- La acción que se desea modificar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Modificar acción.	Se modifica una acción seleccionada.	EP 1.1: Modificar acción.	<ul style="list-style-type: none"> - Se selecciona la acción que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos de la acción que se desea modificar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Modificar acción dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se selecciona la acción que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos en el formulario de Modificar acción dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.3: Modificar acción introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se selecciona la acción que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas - Se introducen los datos con errores. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> -Se selecciona la acción que se va a modificar. -Se presiona el botón izquierdo del mouse dos veces consecutivas. -Se introducen o no los

			datos en el formulario. Se presiona el botón Cancelar .
--	--	--	----------------------------------------------------------------

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar acción.	V(Acción)	Se modifica la acción y se guarda la información modificada, se cierra la ventana.	
EP 1.2	Modificar acción dejando campos requeridos en blanco.	I(Vacío)	Se deja en blanco el campo de texto y sale un mensaje que indica los datos son incorrectos.	
EP 1.3	Modificar acción introduciendo errores en los datos.	I (/ *- / - ' .)	El sistema no permite entrar caracteres inválidos y se mantiene en la ventana Modificar acción.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Tabla 45: DCP de CRUD del escenario Gestionar excepción del componente Generador de excepciones.

DCP Adicionar Excepción

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en la excepción IDE, en Generador de excepciones, en la interfaz Excepción.
- La excepción que se desea adicionar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Adicionar excepción.	Se adiciona una nueva excepción al sistema.	EP 1.1: Adicionar una nueva excepción al sistema.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación una acción y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos de la excepción que se desea adicionar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Adicionar una plantilla de aspectos dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación una acción y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos en el formulario de Adicionar excepción dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra la ventana que muestra el mensaje de información

		EP 1.3: Adicionar una plantilla de aspectos introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación una acción y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos de excepción que se desea adicionar en el formulario introduciendo errores en los datos. - Se presiona el botón Aceptar.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación una acción y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen o no los datos en el formulario. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.
Código	Campo de texto	No	Combinación de letras y números.
Filtro	Campo de texto	Si	Combinación de letras y números.
Tipo de Ejecución	Campo de opción	NA.	B, L, LP, P

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Código	Filtro	Tipo de ejecución	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar	V(Excep	V(Códi	V(Fil	V(Selecci	Para el botón	

	una nueva excepción al sistema.	ción)	go)	tro)	onado)	Adicionar: Se adiciona la excepción y se guarda la información adicionada.	
EP 1.2	Adicionar una excepción dejando campos requeridos en blanco.	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	Para el botón Adicionar: Se deja en blanco el campo de texto y sale un mensaje que indica que los datos son incorrectos.	
EP 1.3	Adicionar una excepción introduciendo errores en los datos.	I (/ *- / - ' .)	I (/ *- / - ' .)	I (/ *- / - ' .)	I (/ *- / - ' .)	Para el botón Adicionar El sistema no permite entrar caracteres inválidos y sale un mensaje que indica que los datos son incorrectos.	
EP 1.4	Cancelar.	NA.	NA.	NA.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Eliminar Excepción

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en la excepción IDE, en Generador de excepciones, en la interfaz Excepciones.
- La excepción que se desea eliminar debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

1: Eliminar excepción.	Se elimina excepción seleccionada.	EP 1.1: Eliminar excepción.	-Se selecciona la excepción que se desea eliminar. -Se presiona el botón Eliminar o se selecciona la excepción en el árbol de navegación y se presiona el clic derecho y se elige la opción de Eliminar . -Se presiona el botón Aceptar del formulario que se presenta.
		EP 1.2: Cancelar.	-Se selecciona la excepción que se desea eliminar. -Se presiona el botón Cancelar .

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar plantilla de aspectos.	NA.	Se muestra un cartel de confirmación: "¿Está seguro que desea eliminar la excepción?". Se elimina la excepción seleccionada.	
EP 1.2	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DGP Modificar Excepción

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en la excepción IDE, en Generador de excepciones, en la interfaz Excepción.
- La excepción que se desea modificar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Modificar excepción.	Se modifica una excepción seleccionada.	EP 1.1: Modificar excepción.	- Se selecciona excepción que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos de la excepción que se desea modificar en el formulario. - Se presiona el botón Aceptar .
		EP 1.2: Modificar excepción dejando campos requeridos en blanco.	-Se selecciona la excepción que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos en el formulario de Modificar excepción de aspectos dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.3: Modificar excepción introduciendo errores en los datos.	-Se selecciona la excepción que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas - Se introducen los datos con errores. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.4: Cancelar.	-Se selecciona la excepción que se va a modificar. -Se presiona el botón

			izquierdo del mouse dos veces consecutivas. -Se introducen o no los datos en el formulario. Se presiona el botón Cancelar .
--	--	--	------------------------------------------------------------------------------------------------------------------------------------------

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras y números.
Código	Campo de texto	No	Combinación de letras y números.
Filtro	Campo de texto	Si	Combinación de letras y números.
Tipo de Ejecución	Campo de opción	NA.	B, L, LP, P

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Código	Filtro	Tipo de ejecución	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar la excepción.	V(Excepción)	V(Código)	V(Filtro)	V(Seleccionado)	Se modifica la excepción y se guarda la información modificada, se cierra la ventana.	
EP 1.2	Modificar excepción dejando campos requeridos en blanco.	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	Se deja en blanco el campo de texto y sale un mensaje que indica los datos son incorrectos.	
EP 1.3	Modificar excepción introduciendo errores en los datos.	I (*-/-'.)	I (*-/-'.)	I (*-/-'.)	I (*-/-'.)	El sistema no permite entrar caracteres inválidos y se mantiene en la	

						ventana Modificar plantilla de aspectos.	
EP 1.4	Cancelar.	NA.	NA.	NA.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Tabla 46: DCP de CRUD del escenario Gestionar lenguaje del componente Generador de excepciones.

DCP Adicionar Lenguaje

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en lenguaje IDE, en Generador de excepciones, en la interfaz de Lenguaje.
- El lenguaje que se desea adicionar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Adicionar lenguaje.	Se adiciona un nuevo lenguaje al sistema.	EP 1.1: Adicionar un nuevo lenguaje al sistema.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación una excepción y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos del aspecto que se desea adicionar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Adicionar un lenguaje dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar o se elige en el árbol de navegación una excepción y se presiona clic derecho y se elige la opción de Adicionar. - Se introducen los datos en el formulario de Adicionar lenguaje dejando al menos un campo

			requerido en blanco. - Se presiona el botón Aceptar o se cierra la ventana que muestra el mensaje de información
		EP 1.3: Adicionar un lenguaje introduciendo errores en los datos.	- Se presiona el botón Adicionar o se elige en el árbol de navegación una excepción y se presiona clic derecho y se elige la opción de Adicionar . - Se introducen los datos del lenguaje que se desea adicionar en el formulario introduciendo errores en los datos. - Se presiona el botón Aceptar .
		EP 1.4: Cancelar.	- Se presiona el botón Adicionar o se elige en el árbol de navegación una excepción y se presiona clic derecho y se elige la opción de Adicionar . - Se introducen o no los datos en el formulario. - Se presiona el botón Cancelar .

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Tipo de lenguaje	Campo de opción	No	Es, en
Mensaje	Campo de texto	No	Combinación de letras y números.
Descripción	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Tipo de Lenguaje	Mensaje	Descripción	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar un nuevo lenguaje al sistema.	V(Selección)	V(Mensaje)	V(Descripción)	Para el botón Adicionar : Se adiciona el lenguaje y se guarda la información adicionada.	
EP 1.2	Adicionar un lenguaje dejando campos requeridos en blanco.	NA.	I(Vacío)	I(Vacío)	Para el botón Adicionar : Se deja en blanco el campo de texto y sale un mensaje que indica que los datos son incorrectos.	
EP 1.3	Adicionar un lenguaje introduciendo errores en los datos.	NA.	I (/ * - / - ' .)	I (/ * - / - ' .)	Para el botón Adicionar El sistema no permite entrar caracteres inválidos y sale un mensaje que indica que los datos son incorrectos.	
EP 1.4	Cancelar.	NA.	NA.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Eliminar Lenguaje

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el lenguaje IDE, en Generador de Excepciones, en la interfaz Lenguaje.
- El lenguaje que se desea eliminar debe existir en el sistema.
- Se debe haber cargado o creado los ficheros

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

1: Eliminar lenguaje.	Se elimina el lenguaje seleccionado.	EP 1.1: Eliminar lenguaje.	<ul style="list-style-type: none"> - Se selecciona el lenguaje que se desea eliminar. - Se presiona el botón Eliminar. - Se presiona el botón Aceptar del formulario que se presenta.
		EP 1.2: Cancelar.	<ul style="list-style-type: none"> - Se selecciona el lenguaje que se desea eliminar. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar lenguaje.	NA.	Se muestra un cartel de confirmación: "¿Está seguro que desea eliminar el lenguaje?". Se elimina el lenguaje seleccionado.	
EP 1.2	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Modificar Lenguaje

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.

- Se debe encontrar en el lenguaje IDE, en Generador de Excepciones, en la interfaz Lenguaje.
- El lenguaje que se desea modificar no debe existir en el sistema.
- Se debe haber cargado o creado los ficheros.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Modificar lenguaje.	Se modifica un lenguaje seleccionado.	EP 1.1: Modificar lenguaje.	<ul style="list-style-type: none"> - Se selecciona el lenguaje que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos del lenguaje que se desea modificar en el formulario. - Se presiona el botón Aceptar.
		EP 1.2: Modificar lenguaje dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se selecciona el lenguaje que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas. - Se introducen los datos en el formulario de Modificar lenguaje dejando al menos un campo requerido en blanco. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.3: Modificar lenguaje introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se selecciona el lenguaje que se va a modificar. - Se presiona el botón izquierdo del mouse dos veces consecutivas - Se introducen los datos con errores. - Se presiona el botón Aceptar o se cierra el formulario.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> -Se selecciona el lenguaje que se va a modificar. -Se presiona el botón izquierdo del mouse dos veces consecutivas. -Se introducen o no los

			datos en el formulario. Se presiona el botón Cancelar .
--	--	--	-------------------------------------------------------------------

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Tipo de lenguaje	Campo de opción	No	Es, en
Mensaje	Campo de texto	No	Combinación de letras y números.
Descripción	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Tipo de Lenguaje	Mensaje	Descripción	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar lenguaje.	V(Seleccionado)	V(Mensaje)	V(Descripción)	Se modifica el lenguaje y se guarda la información modificada, se cierra la ventana.	
EP 1.2	Modificar lenguaje dejando campos requeridos en blanco.	NA.	I(Vacío)	I(Vacío)	Se deja en blanco el campo de texto y sale un mensaje que indica los datos son incorrectos.	
EP 1.3	Modificar lenguaje introduciendo errores en los datos.	NA.	I (/ * - / - ' .)	I (/ * - / - ' .)	El sistema no permite entrar caracteres inválidos y se mantiene en la ventana Modificar lenguaje.	
EP 1.4	Cancelar.	NA.	NA.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Anexo 14: Diseño de casos de pruebas del Generador de Inversión de Control.

- Generador de Inversión de Control.

Tabla 47: Paquete de DCP del escenario Gestionar ficheros perteneciente al componente generador de inversión de control.

DCP Crear Fichero.

1. Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el componente IDE, en Generador de Inversión de Control.
- El fichero que desea crear debe ubicarlo en una carpeta que no tenga implícita ningún fichero del mismo tipo.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Crear fichero.	Se crea un nuevo fichero al sistema.	EP 1.1: Crear fichero al sistema.	- Se presiona la acción Crear nuevo fichero. - Se elije la ubicación donde se va a crear el fichero. - Se presiona el botón Aceptar.
		EP 1.2: Crear fichero dejando campos requeridos en blanco.	- Se presiona la acción Crear nuevo fichero. - Se elije la ubicación donde se va a crear el fichero. - Se introducen los datos, dejando al menos uno de los datos requeridos en blanco. - Se presiona el botón Aceptar.
		EP 1.3: Crear ficheros introduciendo errores en los datos.	- Se presiona la acción Crear nuevo fichero. - Se elije la ubicación donde se va a crear el fichero. - Se introducen los datos con errores

			- Se presiona el botón Aceptar .
		EP 1.4: Cancelar.	- Se presiona la acción Crear nuevo fichero . - Se elige la ubicación donde se va a crear el fichero o no. Se introducen los datos o no. - Se presiona el botón Cancelar .

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto.	No	Combinación de letras.
Ubicación	Campo de texto.	No	Dirección fijada.
Descripción	Campo de texto.	Si.	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Ubicación	Descripción	Respuesta del sistema	Resultado de la prueba
EP 1.1	Crear fichero al sistema.	V(Fichero)	NA.	V(Descripción)	Para la opción de Crear : Se crea y busca la ubicación donde se va a crear el fichero, se crea y se cierra la ventana.	
EP 1.2	Crear fichero dejando campo requeridos en blanco.	I(Vacío)	I(Vacío)	I(Vacío)	Se deja en blanco el campo de texto y sale un mensaje que indica los datos son incorrectos.	
EP 1.3	Crear ficheros introduciendo	I (/ *- / - ' .)	NA.	I (/ *- / - ' .)	El sistema no permite entrar	

	errores en los datos.				caracteres inválidos y se mantiene en la ventana de crear fichero.	
EP 1.4	Cancelar.	NA.	NA.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DGP Cargar Fichero.

1. Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el componente IDE, en Generador de Inversión de Control.
- El fichero que desea cargar debe existir en la carpeta de la ubicación elegida.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Cargar fichero.	Se carga el fichero existente al sistema.	EP 1.1: Cargar fichero al sistema.	- Se elige la acción Abrir fichero existente en el lateral izquierdo. - Se elige la ubicación donde está ubicado el fichero existente. - Se presiona el botón Aceptar .
		EP 1.2: Cargar fichero dejando campo requeridos en blanco.	NA.
		EP 1.3: Cargar fichero introduciendo errores en los datos.	NA.
		EP 1.4: Cancelar.	- Se elige la acción Abrir fichero existente en el lateral izquierdo. - Se elige la ubicación o no donde está ubicado el fichero existente. - Se presiona el botón Cancelar .

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Cargar ficheros existentes al sistema.	NA.	Para la acción Abrir fichero existente: Se busca la ubicación donde se encuentra el fichero existente, se carga y se cierra la ventana.	
EP 1.2	Cargar fichero dejando campo requeridos en blanco.	NA.	NA.	
EP 1.3	Cargar ficheros introduciendo errores en los datos.	NA.	NA.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Guardar Fichero.

1. Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el componente IDE, en Generador de Inversión de Control.
- Se debe de haber cargado o creado el fichero.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Guardar fichero.	Se guarda el fichero del sistema.	EP 1.1: Guardar fichero del sistema.	<ul style="list-style-type: none"> - Se presiona el botón superior izquierdo del menú (Archivo). - Se elije la opción Guardar fichero. - Se elije la ubicación donde será ubicado el fichero. - Se presiona el botón Aceptar.
		EP 1.2: Guardar fichero dejando campos requeridos en blanco.	NA.
		EP 1.3: Cargar ficheros introduciendo errores en los datos.	NA.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se presiona el botón superior izquierdo del menú (Archivo). - Se elije la opción Guardar fichero. - Se elije la ubicación o no donde será ubicado el fichero. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Guardar el fichero existente al sistema.	NA.	Para la opción de Guardar fichero : Se busca la	

			ubicación donde se guardara el fichero, se guarda y se cierra la ventana.	
EP 1.2	Guardar el fichero dejando campos requeridos en blanco.	NA.	NA.	
EP 1.3	Guardar el fichero introduciendo errores en los datos.	NA.	NA.	
EP 1.4	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Tabla 48: DCP de CRUD del escenario Gestionar subsistema del componente Generador de inversión de control.

DCP Adicionar Subsistema

1. Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el subsistema IDE, en Generador de Inversión de Control.
- El subsistema que se desea adicionar no debe existir en el sistema.
- Se debe haber cargado o creado el fichero.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Adicionar subsistema.	Se adiciona un nuevo subsistema al sistema.	EP 1.1: Adicionar un nuevo subsistema al sistema.	<ul style="list-style-type: none"> - Se presiona la acción Crear nuevo fichero en el lateral izquierdo del formulario. - Se introducen los datos del subsistema que se desea adicionar en el formulario. - Se presiona el botón Aceptar.

		EP 1.2: Adicionar un subsistema dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se presiona la acción Crear nuevo fichero en el lateral izquierdo del formulario. - Se introducen los datos del subsistema dejando al menos un campo requerido en blanco de los que se desean adicionar en el formulario. - Se presiona el botón Aceptar.
		EP 1.3: Adicionar un subsistema introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se presiona la acción Crear nuevo fichero en el lateral izquierdo del formulario. - Se introducen datos con errores del subsistema en el formulario. - Se presiona el botón Aceptar.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se presiona la acción Crear nuevo fichero en el lateral izquierdo del formulario. - Se introducen los datos del subsistema o no que se desea adicionar en el formulario. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras.
Ubicación	Campo de texto	No	Dirección fijada.
Descripción	Campo de texto	No	Combinación de letras y números.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Ubicación	Descripción	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar un nuevo subsistema al sistema.	V(subsistema)	NA.	V(Descripción)	Para la acción Crear nuevo fichero : Se adiciona el subsistema y se guarda la información adicionada.	
EP 1.2	Adicionar un subsistema dejando campos requeridos en blanco.	I(Vacío)	No.	I(Vacío)	Para la acción Crear nuevo fichero : Se deja en blanco el campo de texto y sale un mensaje que indica que los datos son incorrectos.	
EP 1.3	Adicionar un subsistema introduciendo errores en los datos.	I (/ * - / - ' .)	NA.	I (/ * - / - ' .)	Para la acción Crear nuevo fichero . El sistema no permite entrar caracteres inválidos y sale un mensaje que indica que los datos son incorrectos.	
EP 1.4	Cancelar.	NA.	NA.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Eliminar Subsistema

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el subsistema IDE, en Generador de Inversión de Control, en la interfaz subsistema.
- El subsistema que se desea eliminar debe existir en el sistema.
- Se debe haber cargado o creado el fichero.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Eliminar subsistema.	Se elimina el subsistema seleccionado.	EP 1.1: Eliminar subsistema.	<ul style="list-style-type: none"> - Se selecciona el subsistema que se desea eliminar. - Se presiona el botón Eliminar. - Se presiona el botón Aceptar del formulario que se presenta.
		EP 1.2: Cancelar.	<ul style="list-style-type: none"> - Se selecciona el subsistema que se desea eliminar. - Se presiona el botón Cancelar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar subsistema.	NA.	Se elimina el subsistema seleccionado. Sale un cartel que indica: El subsistema ha sido eliminado satisfactoriamente.	
EP 1.2	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Tabla 49: DCP de CRUD del escenario Gestionar contrato del componente Generador de inversión de control.

DCP Adicionar Contrato

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en contrato IDE, en Generador de Inversión de Control, en la interfaz de contrato.
- El contrato que se desea adicionar no debe existir en el sistema.
- Se debe haber cargado o creado el fichero.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Adicionar contrato.	Se adiciona un nuevo contrato al sistema.	EP 1.1: Adicionar un nuevo contrato al sistema.	- Se presiona el botón Contrato . - Se introducen los datos del contrato que se desea adicionar en el formulario. - Se presiona el botón Aceptar .
		EP 1.2: Adicionar un contrato dejando campos requeridos en blanco.	- Se presiona el botón Contrato . - Se introducen los datos del contrato que se desea adicionar y se deja un campo requerido en blanco en el formulario. - Se presiona el botón Aceptar .
		EP 1.3: Adicionar un contrato introduciendo errores en los datos.	- Se presiona el botón Contrato . - Se introducen los datos con errores del contrato que se desea adicionar en el formulario. - Se presiona el botón Aceptar .
		EP 1.4: Cancelar.	- Se presiona el botón Contrato . - Se introducen los datos o no del contrato que se

			desea adicionar en el formulario. - Se presiona el botón Cancelar .
--	--	--	-------------------------------------------------------------------------------

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras.
Clase inyectora	Campo de texto	No	Combinación de letras.
Nombre función inyectora	Campo de texto	No	Combinación de letras.
Referencia contrato.	Campo de texto	No	Dirección fijada.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Clase inyectora	Nombre función inyectora	Referencia contrato.	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar un nuevo contrato al sistema.	V(Contrato)	V(Clase inyectora)	V(Clase función inyectora)	NA.	Para el botón Adicionar : Se adiciona el contrato y se guarda la información adicionada.	
EP 1.2	Adicionar un contrato dejando campos requeridos en blanco.	I(Vacío)	I(Vacío)	I(Vacío)	No.	Para el botón Adicionar : Se deja en blanco el campo de texto y sale un mensaje que indica que los datos son incorrectos.	
EP 1.3	Adicionar un contrato introduciendo errores en los datos.	I (/ *- / - ' .)	I (/ *- / - ' .)	I (/ *- / - ' .)	NA.	Para el botón Adicionar El sistema no permite entrar caracteres inválidos y sale un mensaje que indica que los datos son incorrectos.	

EP 1.4	Cancelar.	NA.	NA.	NA.	NA.	Se cancela la operación.	
--------	-----------	-----	-----	-----	-----	--------------------------	--

Fuente: Vargas D. y Hernández M. 2009.

DCP Eliminar Contrato

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el contrato IDE, en Generador de Inversión de Controles, en la interfaz Contrato.
- El contrato que se desea eliminar debe existir en el sistema.
- Se debe haber cargado o creado el fichero.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Eliminar contrato.	Se elimina el contrato seleccionado.	EP 1.1: Eliminar contrato.	- Se selecciona el contrato que se desea eliminar. - Se presiona el botón Eliminar . - Se presiona el botón Aceptar del formulario que se presenta.
		EP 1.2: Cancelar.	- Se selecciona el contrato que se desea eliminar. - Se presiona el botón Cancelar .

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar contrato.	NA.	Se elimina el contrato seleccionado. Se muestra un mensaje que dice: "el contrato a sido eliminado correctamente".	
EP 1.2	Cancelar.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Modificar Contrato

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el contrato IDE, en Generador de Inversión de Control, en la interfaz Contrato.
- El contrato que se desea modificar no debe existir en el sistema.
- Se debe haber cargado o creado el fichero.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Modificar contrato.	Se modifica un contrato seleccionado.	EP 1.1: Modificar contrato.	- Se selecciona el contrato que se va a modificar. - Se aprieta el botón Editar . - Se introducen los datos nuevos. - Se aprieta el botón Modificar .
		EP 1.2: Modificar contrato dejando campos requeridos en blanco.	- Se selecciona el contrato que se va a modificar. - Se aprieta el botón Editar . - Se introducen los datos nuevos, dejando algún campo requerido en blanco. - Se aprieta el botón

			Modificar.
		EP 1.3: Modificar contrato introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se selecciona el contrato que se va a modificar. - Se aprieta el botón Editar. - Se introducen los datos nuevos con errores. - Se aprieta el botón Modificar.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se selecciona el contrato que se va a modificar. - Se aprieta el botón Editar. - Se introducen los datos nuevos o no. <p>Se presiona el botón Cancelar.</p>

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Nombre	Campo de texto	No	Combinación de letras.
Clase inyectora	Campo de texto	No	Combinación de letras.
Nombre función inyectora	Campo de texto	No	Combinación de letras.
Referencia contrato.	Campo de texto	No	Dirección fijada.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Clase inyectora	Nombre función inyectora	Referencia contrato.	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar un nuevo contrato al sistema.	V(Contrato)	V(Clase inyectora)	V(Clase función inyectora)	NA.	Para el botón Editar : Se introducen los datos nuevos del contrato y se guarda la información introducida.	
EP 1.2	Modificar un contrato dejando	I(Vacío)	I(Vacío)	I(Vacío)	No.	Para el botón Editar : Se deja en blanco el campo de texto y sale un mensaje que	

	campos requeridos en blanco.					indica que los datos son incorrectos.	
EP 1.3	Modificar un contrato introduciendo errores en los datos.	I (/ *- / - ' .)	I (/ *- / - ' .)	I (/ *- / - ' .)	NA.	Para el botón Editar El sistema no permite entrar caracteres inválidos y sale un mensaje que indica que los datos son incorrectos.	
EP 1.4	Cancelar.	NA.	NA.	NA.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Anexo 15: Diseño de casos de pruebas del componente Auditor de Código Fuente.

➤ Auditor de Código Fuente

Tabla 50: Paquete de DCP del escenario Gestionar ficheros perteneciente al componente Auditor de código fuente.

DCP Crear Proyecto.

1. Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el componente IDE, en Auditor de Código Fuente.
- El proyecto que desea crear será guardado en la base de datos.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Crear proyecto.	Se crea un nuevo proyecto al sistema.	EP 1.1: Crear proyecto al sistema.	<ul style="list-style-type: none"> - Se presiona la acción Nuevo proyecto. - Se elige la opción de Instanciar. - Se introducen en el

			nuevo formulario los datos. - Se presiona el botón Terminar .
		EP 1.2: Crear proyecto dejando campos requeridos en blanco.	- Se presiona la acción Nuevo proyecto . - Se elije la opción de Instanciar . - Se introducen en el nuevo formulario los datos, dejando alguno requerido en blanco. - Se presiona el botón Terminar .
		EP 1.3: Crear ficheros introduciendo errores en los datos.	- Se presiona la acción Nuevo proyecto . - Se elije la opción de Instanciar . - Se introducen en el nuevo formulario los datos con errores. - Se presiona el botón Terminar .
		EP 1.4: Cancelar.	- Se presiona la acción Nuevo proyecto . - Se elije la opción de Instanciar . - Se introducen en el nuevo formulario los datos o no - Se presiona el botón Cancelar .

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Instanciar	Campo de chequeo.	No	Si o No (uno de los dos).
Nombre	Campo de texto	No	Combinación de letras.
Firma	Campo de texto	No	Combinación de letras y números.
Incluir firma	Campo de chequeo.	No	Si o No.
Ubicación	Campo de texto	No	Dirección fijada.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Instanciar	Nombre	Firma	Incluir firma	Ubicación	Respuesta del sistema	Resultado de la prueba
EP 1.1	Crear proyecto al sistema.	NA.	V(Proyecto)	V(Firma)	NA.	NA.	Para la opción de Crear : Se crea y busca la ubicación donde se encuentra los ficheros del sistema.	
EP 1.2	Crear proyecto dejando campo requeridos en blanco.	NA.	I(Vacío)	I(Vacío)	NA.	NA.	Se deja en blanco el campo de texto y sale un mensaje que indica los datos son incorrectos.	
EP 1.3	Crear proyecto introduciendo errores en los datos.	NA.	I (/ * - / - ' .)	I (/ * - / - ' .)	NA.	NA.	El sistema no permite entrar caracteres inválidos y se mantiene en la ventana de crear proyecto.	
EP 1.4	Cancelar.	NA.	NA.	NA.	NA.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Cargar Proyecto.

1. Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el componente IDE, en Auditor de Código Fuente.
- El proyecto que desea cargar debe existir en la carpeta de la ubicación elegida.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Cargar proyecto.	Se carga el proyecto existente al sistema.	EP 1.1: Cargar proyecto al sistema.	<ul style="list-style-type: none"> - Se presiona la acción Nuevo proyecto. - Se elije la opción de Auditar. - Se introducen en el nuevo formulario los datos. - Se presiona el botón Terminar.
		EP 1.2: Cargar proyecto dejando campo requeridos en blanco.	<ul style="list-style-type: none"> - Se presiona la acción Nuevo proyecto. - Se elije la opción de Auditar. - Se introducen en el nuevo formulario los datos dejando alguno requerido en blanco. - Se presiona el botón Terminar.
		EP 1.3: Cargar proyecto introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se presiona la acción Nuevo proyecto. - Se elije la opción de Auditar. - Se introducen en el nuevo formulario los datos con errores. - Se presiona el botón Terminar.
		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se presiona la acción Nuevo proyecto. - Se elije la opción de Auditar. - Se introducen en el nuevo formulario los datos o no. - Se presiona el botón Terminar.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
Auditar	Campo de chequeo.	NA.	Si o No.
Seleccionar proyecto	Campo de texto.	No	Dirección fijada.
Ubicación	Campo de texto.	No	Dirección fijada.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Auditar	Seleccionar proyecto	Ubicación	Respuesta del sistema	Resultado de la prueba
EP 1.1	Cargar proyecto existente al sistema.	NA.	NA.	NA.	Para la acción Cargar proyecto existente: Se busca el nombre del proyecto, además se busca la ubicación donde se encuentra el proyecto existente, se carga y se cierra la ventana.	
EP 1.2	Cargar proyecto dejando campo requeridos en blanco.	NA.	NA.	NA.	NA.	
EP 1.3	Cargar proyecto introduciendo errores en los datos.	NA.	NA.	NA.	NA.	
EP 1.4	Cancelar.	NA.	NA.	NA.	Se cancela la operación.	

Fuente: Vargas D. y Hernández M. 2009.

DCP Chequear Ficheros

1. Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- Se debe encontrar en el contrato IDE, en Auditor de Código Fuente, en la interfaz Auditar.
- Para chequear los ficheros debe de haber realizado la opción de Auditar.
- Se debe haber cargado el proyecto.

1.1 Requisitos a probar.

Nombre del requisito	Descripción General	Escenarios de pruebas	Flujo del escenario
1: Chequear ficheros.	Chequear los ficheros del proyecto cargado.	EP 1.1: Chequear	-Se presiona el botón Chequear.
		EP 1.2: Cancelar.	- No se realiza la operación.

Fuente: Vargas D. y Hernández M. 2009.

1.1.1 Descripción de variable.

Nombre del campo	Clasificación	Puede ser nulo	Descripción
NA.	NA.	NA.	NA.

Fuente: Vargas D. y Hernández M. 2009.

1.1.2 Juego de datos a probar.

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Chequear ficheros	NA.	Se chequea los ficheros del proyecto cargado y si no existen modificaciones saldrá un mensaje de que el proceso fue satisfactorio, sino se marcaran de rojo los ficheros modificados y saldrá un mensaje indicando la	

			cantidad.	
EP 1.2	Cancelar.	NA.	No se realiza la operación.	

Fuente: Vargas D. y Hernández M. 2009.

Anexo 16: Instrumento de medición de la métrica de Tamaño de Clases (TC).

Tabla 51: Rango de valores para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de implementación y Reutilización) relacionados con la métrica TC.

	Categoría	Criterio
Responsabilidad	Baja	< =Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.
Complejidad implementación	Baja	< =Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.
Reutilización	Baja	> 2*Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	<= Prom.

Fuente: Vargas D. y Hernández M. 2009.

Tabla 52: Resultados de la evaluación de la métrica TC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).

Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
Modulo	6	Media	Media	Media
Controlador	6	Media	Media	Media
Accion	6	Media	Media	Media
AspectTemplate	6	Media	Media	Media
Aspecto	7	Media	Media	Media
Ejecucion	0	Baja	Baja	Alta
Clase	6	Media	Media	Media
Metodo	6	Media	Media	Media
ManagerException	6	Media	Media	Media
gstModuleCollection	5	Media	Media	Media
gstControllerCollection	5	Media	Media	Media
gstAccionCollection	7	Media	Media	Media

gstAspectTemplateCollection	6	Media	Media	Media
gstAspectCollection	5	Media	Media	Media
gstWearverXmlLoader	4	Baja	Baja	Alta
gst_aspectXmlLoader	2	Baja	Baja	Alta
gst_aspTempXmlLoader	2	Baja	Baja	Alta
gstPersistWeaver	1	Baja	Baja	Alta
Tools	2	Baja	Baja	Alta
gstTree	13	Alta	Alta	Baja
ComparableList	3	Baja	Baja	Alta
Controlador	6	Media	Media	Media
Accion	6	Media	Media	Media
Excepcion	6	Media	Media	Media
Lenguaje	6	Media	Media	Media
Tipo_Lenguaje	0	Baja	Baja	Alta
ExcepcionesTipo	0	Baja	Baja	Alta
gstModuleCollection	4	Baja	Baja	Alta
gstControllerCollection	4	Baja	Baja	Alta
gstAccionCollection	4	Baja	Baja	Alta
Modulo	6	Media	Media	Media
gstExcepcionCollection	6	Media	Media	Media
gstLenguajeCollection	2	Baja	Baja	Alta
gstManagerExcepcionXmlLoader	5	Media	Media	Media
gstExcepcionXmlLoader	3	Baja	Baja	Alta
gstXml	2	Baja	Baja	Alta
gstTree	14	Alta	Alta	Baja
ManagerExcepcion	6	Media	Media	Media
ComparableList	3	Baja	Baja	Alta
Persistidor	5	Media	Media	Media
Parcial_Contrato	6	Media	Media	Media
IoCResultado	1	Baja	Baja	Alta
CargadorXml	7	Media	Media	Media
ParcialXml	7	Media	Media	Media
IoCFichero	0	Baja	Baja	Alta
IoCSubsistema	1	Baja	Baja	Alta
IoCContrato	1	Baja	Baja	Alta
IoCInyector	1	Baja	Baja	Alta
IoCPrototipo	1	Baja	Baja	Alta
IoCParametro	1	Baja	Baja	Alta
Proyecto	2	Baja	Baja	Alta
Fichero	2	Baja	Baja	Alta
Css	0	Baja	Baja	Alta
Configuracion	3	Baja	Baja	Alta
Revision	2	Baja	Baja	Alta
TipoProyecto	0	Baja	Baja	Alta
Incidencia	2	Baja	Baja	Alta
Usuario	1	Baja	Baja	Alta

ElementoChequeo	2	Baja	Baja	Alta
gstSingFiles	15	Alta	Alta	Baja
gstTextBoxTools	2	Baja	Baja	Alta
gstRandNumber	3	Baja	Baja	Alta
CadTools	11	Alta	Alta	Baja
gstTools	3	Baja	Baja	Alta
gstBD_Proyecto	3	Baja	Baja	Alta
gstBD_ElemChequeo	2	Baja	Baja	Alta
gstBD_Fichero	3	Baja	Baja	Alta
Conection	4	Baja	Baja	Alta
ScanEventHandler	4	Baja	Baja	Alta
FileServices	9	Alta	Alta	Baja
Crc64	9	Alta	Alta	Baja

Fuente: Vargas D. y Hernández M. 2009.

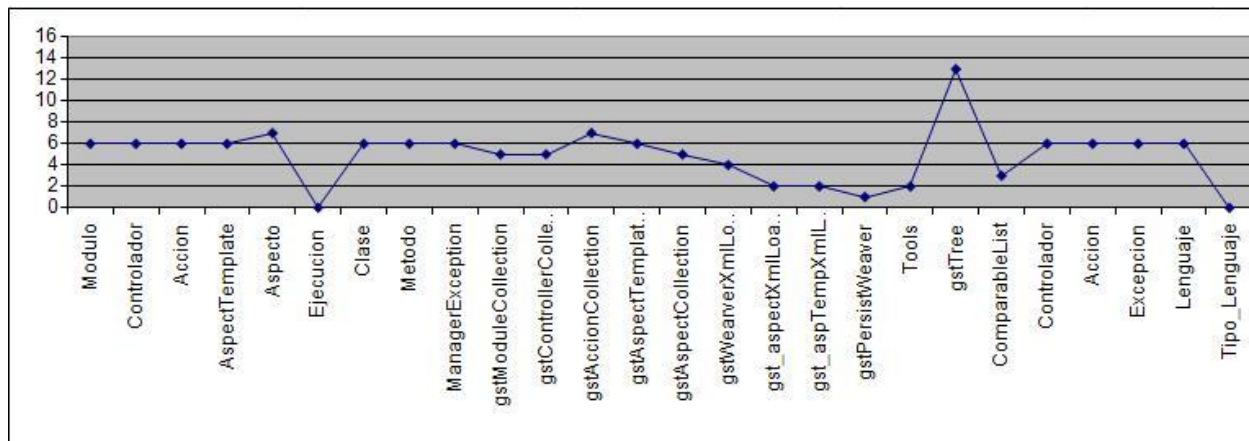


Figura 68: Gráfica de los resultados de la evaluación de la métrica TC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización), parte 1.

Fuente: Vargas D. y Hernández M. 2009.

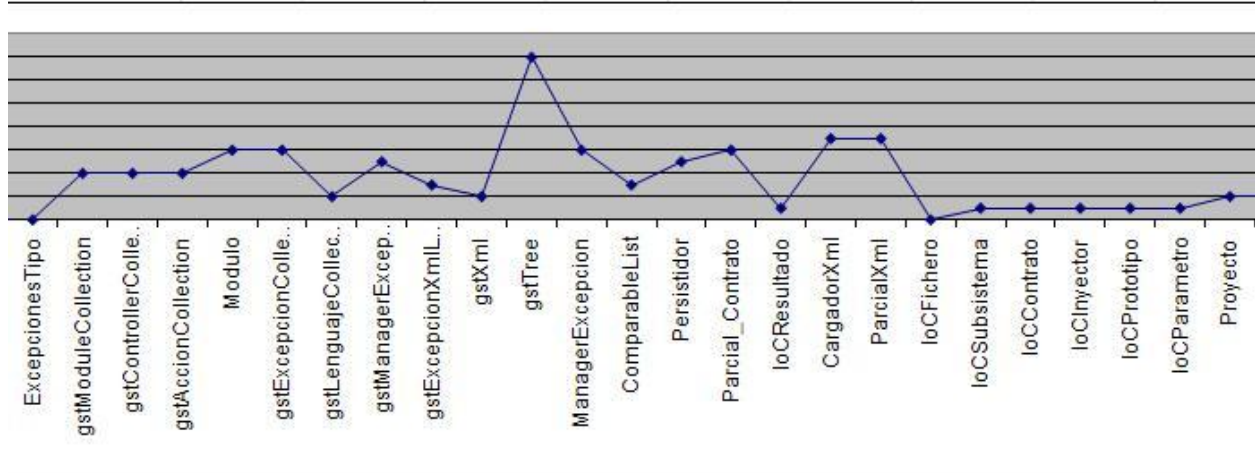


Figura 69: Gráfica de los resultados de la evaluación de la métrica TC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización), parte 2.

Fuente: Vargas D. y Hernández M. 2009.

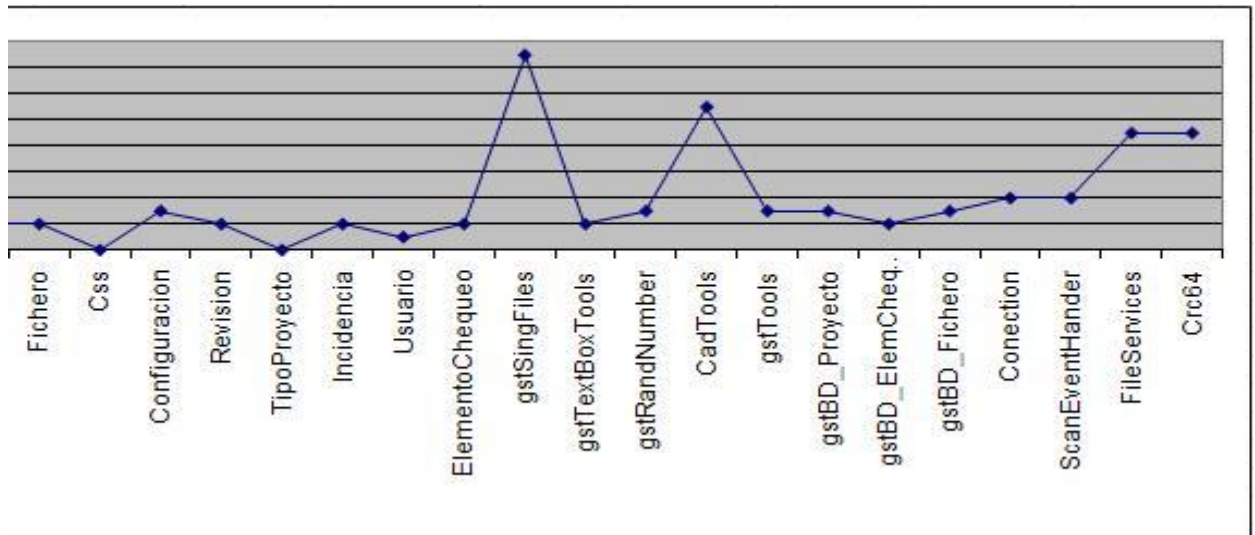


Figura 70: Gráfica de los resultados de la evaluación de la métrica TC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización), parte 3.

Fuente: Vargas D. y Hernández M. 2009.

Anexo 17: Instrumento de medición de la métrica Respuesta para una clase (RPC).

Tabla 53: Rango de valores de para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas) relacionados con la métrica RPC.

	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2

	Categoría	Criterio
Complejidad Mant.	Baja	\leq Prom.
	Media	Entre Prom. y $2 \cdot$ Prom.
	Alta	$> 2 \cdot$ Prom.

	Categoría	Criterio
Reutilización	Baja	$>2 \cdot$ Prom.
	Media	Entre Prom. y $2 \cdot$ Prom.
	Alta	\leq Prom.

	Categoría	Criterio
Cantidad de Pruebas	Baja	\leq Prom.
	Media	Entre Prom. y $2 \cdot$ Prom.
	Alta	$> 2 \cdot$ Prom.

Fuente: Vargas D. y Hernández M. 2009.

Tabla 54: Resultados de la evaluación de la métrica RPC y su influencia en los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas).

Clase	Cantidad de Relaciones de Uso	Acoplamiento	Complejidad Mant.	Reutilización	Cantidad de Pruebas
Modulo	1	Bajo	Media	Media	Media
Controlador	1	Bajo	Media	Media	Media
Accion	1	Bajo	Media	Media	Media
AspectTemplate	1	Bajo	Media	Media	Media
Aspecto	2	Medio	Alta	Baja	Alta
Ejecucion	0	Ninguno	Baja	Alta	Baja
Clase	1	Bajo	Media	Media	Media
Metodo	0	Ninguno	Baja	Alta	Baja

ManagerException	0	Ninguno	Baja	Alta	Baja
gstModuleCollection	1	Bajo	Media	Media	Media
gstControllerCollection	1	Bajo	Media	Media	Media
gstAccionCollection	1	Bajo	Media	Media	Media
gstAspectTemplateCollection	1	Bajo	Media	Media	Media
gstAspectCollection	1	Bajo	Media	Media	Media
gstWearverXmlLoader	0	Ninguno	Baja	Alta	Baja
gst_aspectXmlLoader	0	Ninguno	Baja	Alta	Baja
gst_aspTempXmlLoader	0	Ninguno	Baja	Alta	Baja
gstPersistWeaver	0	Ninguno	Baja	Alta	Baja
Tools	0	Ninguno	Baja	Alta	Baja
gstTree	0	Ninguno	Baja	Alta	Baja
ComparableList	0	Ninguno	Baja	Alta	Baja
Controlador	1	Bajo	Media	Media	Media
Accion	1	Bajo	Media	Media	Media
Excepcion	2	Medio	Alta	Baja	Alta
Lenguaje	1	Bajo	Media	Media	Media
Tipo_Lenguaje	0	Ninguno	Baja	Alta	Baja
ExcepcionesTipo	0	Ninguno	Baja	Alta	Baja
gstModuleCollection	1	Bajo	Media	Media	Media
gstControllerCollection	1	Bajo	Media	Media	Media
gstAccionCollection	1	Bajo	Media	Media	Media
Modulo	1	Bajo	Media	Media	Media
gstExcepcionCollection	1	Bajo	Media	Media	Media
gstLenguajeCollection	1	Bajo	Media	Media	Media
gstManagerExcepcionXmlLoader	0	Ninguno	Baja	Alta	Baja
gstExcepcionXmlLoader	0	Ninguno	Baja	Alta	Baja
gstXml	0	Ninguno	Baja	Alta	Baja
gstTree	0	Ninguno	Baja	Alta	Baja
ManagerExcepcion	0	Ninguno	Baja	Alta	Baja
ComparableList	0	Ninguno	Baja	Alta	Baja
Persistidor	0	Ninguno	Baja	Alta	Baja
Parcial_Contrato	0	Ninguno	Baja	Alta	Baja
IoCResultado	0	Ninguno	Baja	Alta	Baja
CargadorXml	0	Ninguno	Baja	Alta	Baja
ParcialXml	0	Ninguno	Baja	Alta	Baja
IoCFichero	1	Bajo	Media	Media	Media
IoCSubsistema	1	Bajo	Media	Media	Media

IoCContrato	2	Medio	Alta	Baja	Alta
IoCInyector	0	Ninguno	Baja	Alta	Baja
IoCPrototipo	1	Bajo	Media	Media	Media
IoCParametro	0	Ninguno	Baja	Alta	Baja
Proyecto	1	Bajo	Media	Media	Media
Fichero	2	Medio	Alta	Baja	Alta
Css	0	Ninguno	Baja	Alta	Baja
Configuracion	1	Bajo	Media	Media	Media
Revision	0	Ninguno	Baja	Alta	Baja
TipoProyecto	0	Ninguno	Baja	Alta	Baja
Incidencia	0	Ninguno	Baja	Alta	Baja
Usuario	0	Ninguno	Baja	Alta	Baja
ElementoChequeo	0	Ninguno	Baja	Alta	Baja
gstSingFiles	0	Ninguno	Baja	Alta	Baja
gstTextBoxTools	0	Ninguno	Baja	Alta	Baja
gstRandNumber	0	Ninguno	Baja	Alta	Baja
CadTools	0	Ninguno	Baja	Alta	Baja
gstTools	0	Ninguno	Baja	Alta	Baja
gstBD_Proyecto	0	Ninguno	Baja	Alta	Baja
gstBD_ElemChequeo	0	Ninguno	Baja	Alta	Baja
gstBD_Fichero	0	Ninguno	Baja	Alta	Baja
Conexion	0	Ninguno	Baja	Alta	Baja
ScanEventHandler	0	Ninguno	Baja	Alta	Baja
FileServices	1	Bajo	Media	Media	Media
Crc64	0	Ninguno	Baja	Alta	Baja

Fuente: Vargas D. y Hernández M. 2009.

Anexo 18: Instrumento de medición de la métrica Árbol de profundidad de Herencia (APH).

Tabla 55: Resultados de la evaluación de la métrica APH.

Clase	Clase padre	Niveles del arbol de herencia
Modulo	ComparablesList	2
Controlador	ComparablesList	2
Accion	ComparablesList	2
AspectTemplate	ComparablesList	2
Aspecto	ComparablesList	2
Clase	ComparablesList	2
Metodo	ComparablesList	2
ComparablesList		1
Modulo	ComparablesList	2

Controlador	ComparablesList	2
Accion	ComparablesList	2
Excepcion	ComparablesList	2
Lenguaje	ComparablesList	2
ComparablesList		1

Fuente: Vargas D. y Hernández M. 2009.

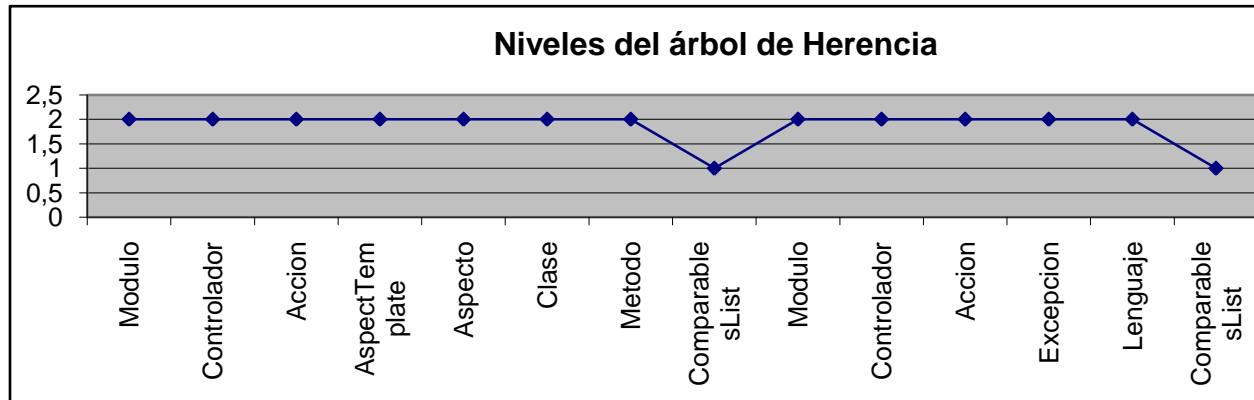


Figura 71: Gráfica de los resultados de la evaluación de la métrica APH.

Fuente: Vargas D. y Hernández M. 2009.

Anexo 19: Instrumento de medición de la métrica Número de Descendientes (NDD).

	Categoría	Criterio
Reutilización	Baja	\leq Prom.
	Media	Entre Prom. y $2 \times$ Prom.
	Alta	$> 2 \times$ Prom.
Abstracción	Indefinida	> 5
	Afectada	Entre 2 y 5
	Definida	≤ 2
Cohesión	Baja	> 5
	Media	Entre 2 y 5
	Alta	≤ 2
Cantidad de Pruebas	Baja	≤ 2
	Media	Entre 2 y 5
	Alta	> 5

Figura 72: Rango de valores para la evaluación técnica de los atributos de calidad (Reutilización, Abstracción del diseño, Nivel de Cohesión y Cantidad de Pruebas) relacionados con la métrica NDD.

Fuente: Vargas D. y Hernández M. 2009.

Tabla 56: Resultados de la evaluación de la métrica NDD y su influencia en los atributos de calidad (Reutilización, Abstracción del diseño, Nivel de Cohesión y Cantidad de Pruebas).

Clase	Clase padre	Número de descendientes	Reutilización	Abstracción de la clase base	Cohesión de la JC	Cantidad de Pruebas
Modulo	ComparablesList	0	Baja	Definida	Alta	Baja
Controlador	ComparablesList	0	Baja	Definida	Alta	Baja
Accion	ComparablesList	0	Baja	Definida	Alta	Baja
AspectTemplate	ComparablesList	0	Baja	Definida	Alta	Baja
Aspecto	ComparablesList	0	Baja	Definida	Alta	Baja
Clase	ComparablesList	0	Baja	Definida	Alta	Baja
Metodo	ComparablesList	0	Baja	Definida	Alta	Baja
ComparablesList		7	Alta	Indefinida	Baja	Alta
Modulo	ComparablesList	0	Baja	Definida	Alta	Baja
Controlador	ComparablesList	0	Baja	Definida	Alta	Baja
Accion	ComparablesList	0	Baja	Definida	Alta	Baja
Excepcion	ComparablesList	0	Baja	Definida	Alta	Baja
Lenguaje	ComparablesList	0	Baja	Definida	Alta	Baja
ComparablesList		5	Alta	Afectada	Media	Media

Fuente: Vargas D. y Hernández M. 2009.

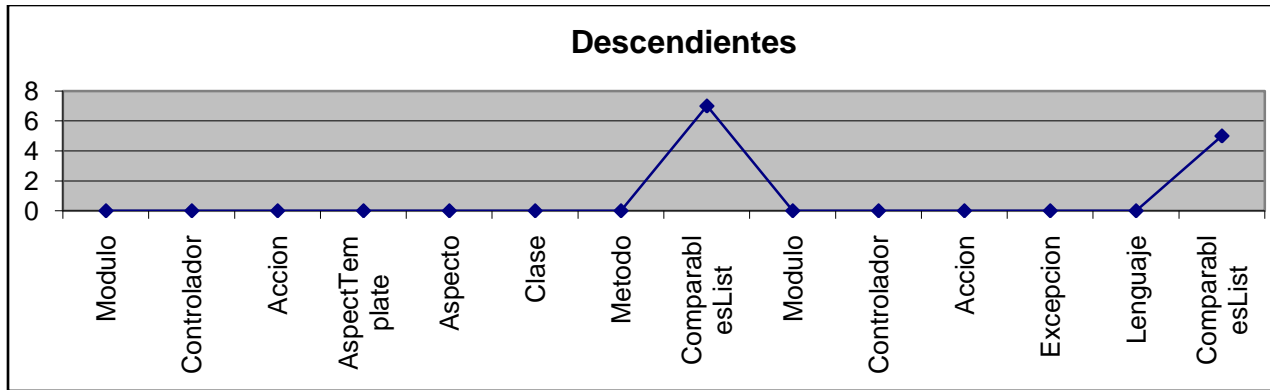


Figura 73: Representación de los resultados obtenidos al aplicar los instrumentos que evalúan la métrica NDD.

Fuente: Vargas D. y Hernández M. 2009.

Anexo 20: Instrumento de medición de la métrica Número de Operaciones Redefinidas para una Sub-Clase (NOR).

	Categoría	Criterio
Complejidad de Mantenimiento	Baja	0
	Media	Entre 0 y 2
Cantidad de Pruebas	Baja	0
	Media	Entre 0 y 2
Violación de la Abstracción representada por la superclase	Baja	0
	Media	Entre 0 y 2

Figura 74: Rango de valores para la evaluación técnica de los atributos de calidad (Abstracción del diseño, Cantidad de pruebas, Complejidad de Mantenimiento) relacionados con la métrica NOR.

Fuente: Vargas D. y Hernández M. 2009.

Tabla 57: Resultados de la evaluación de la métrica NOR y su influencia en los atributos de calidad (Abstracción del diseño, Cantidad de pruebas, Complejidad de Mantenimiento).

Clase	Clase padre	No de Operaciones	Calidad del	Complejidad de Mant.	Cantidad de Pruebas	Violación de ARS

		redefinidas	Diseño			
Modulo	ComparablesList	0	Buena	Baja	Baja	Baja
Controlador	ComparablesList	0	Buena	Baja	Baja	Baja
Accion	ComparablesList	0	Buena	Baja	Baja	Baja
AspectTemplate	ComparablesList	0	Buena	Baja	Baja	Baja
Aspecto	ComparablesList	0	Buena	Baja	Baja	Baja
Clase	ComparablesList	0	Buena	Baja	Baja	Baja
Metodo	ComparablesList	0	Buena	Baja	Baja	Baja
ComparablesList		0	Buena	Baja	Baja	Baja
Modulo	ComparablesList	0	Buena	Baja	Baja	Baja
Controlador	ComparablesList	0	Buena	Baja	Baja	Baja
Accion	ComparablesList	0	Buena	Baja	Baja	Baja
Excepcion	ComparablesList	0	Buena	Baja	Baja	Baja
Lenguaje	ComparablesList	0	Buena	Baja	Baja	Baja
ComparablesList		0	Buena	Baja	Baja	Baja

Fuente: Vargas D. y Hernández M. 2009.

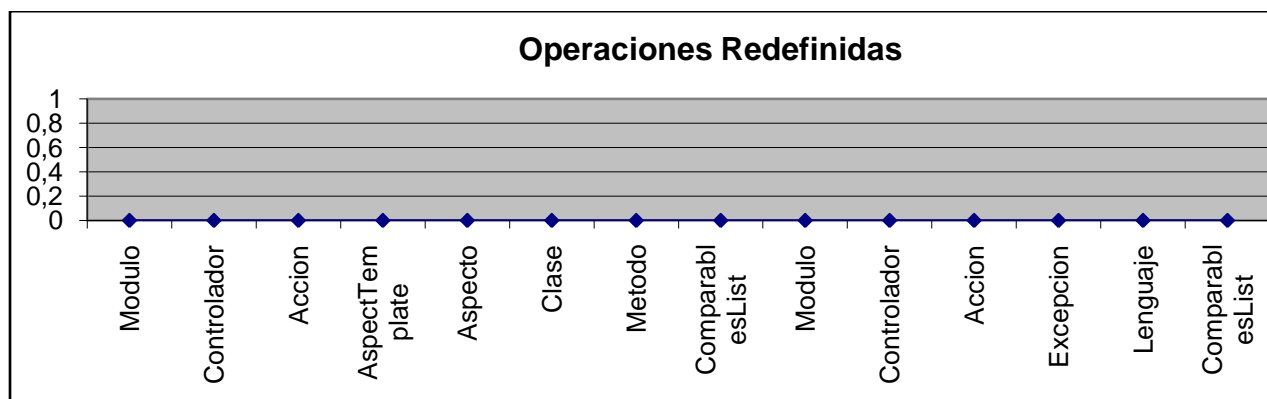


Figura 75: Representación de los resultados obtenidos al aplicar los instrumentos que evalúan la métrica NOR.

Fuente: Vargas D. y Hernández M. 2009.