



Universidad de las Ciencias Informáticas

FACULTAD 5

Título: Desarrollo de un Software que implemente un modelo de madurez para SOA.

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor: Alain Arturo Chacón Casas.

Tutor: Ing. Arturo Cesar Arias Orizondo.

Agradecimientos

Agradecer a todas las personas que de alguna manera me han ayudado sería casi un capítulo más de la tesis, trataré de ser lo más explícito posible, con el fin de que todos quepan en una cuartilla.

Inicialmente, quisiera poder agradecerle a la persona más importante en mi vida, a mi querida madre. A ti, muchas gracias por darme la vida, y en el transcurso de la misma servirme de guía y ejemplo. A toda mi familia, sin excepción de ninguno, a mis tíos, mis primos, a mis dos hermanas Delia y Karina, a mis adorables abuelas, en fin a todos, muchas gracias por sus consejos y apoyo. Agradecer, a esa persona malcriada y majadera, pero a la vez dulce, preocupada, atenta y solidaria, que me ha dado de forma incondicional su amor y cariño, a ti Liena, muchas gracias. A mi tutor, el profesor Arturo, por todo el conocimiento que me ha dado, por nunca decirme que no cuando necesitaba su ayuda, por siempre estar a mi lado en todos los eventos en donde se presentara la tesis y defender a capa y espada el trabajo realizado. Muchas gracias por todo y creo que siempre estaré en deuda con usted. A todos mis profesores, por transmitirme la sabiduría y el

conocimiento necesario para poder llegar a ser realidad este día. A ustedes dondequiera que estén muchas gracias.

A todas mis amistades, gracias por todo lo que de una forma u otra han podido hacer por mí. Discúlpenme que no especifique sus nombres, pero me tomaría mucho tiempo el poder hacerlo, a cada uno de ustedes muchas gracias.

Declaración de autoría

Declaro que soy el único autor del presente trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Alain Arturo Chacón Casas.

Ing. Arturo Cesar Arias Orizondo.

DATOS DE CONTACTO

Síntesis del Tutor: Especialidad de graduación: Ingeniero Informático | Categoría docente: asistente | Facultad 5, Universidad de las Ciencias Informáticas (UCI) | Años de experiencia en el tema: cuatro| Años de graduado: ocho| email: arturo@uci.cu

Resumen

Hoy en día las empresas presentan con conjunto de necesidades, entre las que se pueden mencionar, el logro de diferencias competitivas, lograr agilidad en su negocio y el desarrollo de servicios mediante la utilización de las nuevas tecnologías. Estas necesidades pueden suprimirse haciendo uso de una Arquitectura Orientada a Servicios (SOA por sus siglas en inglés), pero el proceso de adopción de la misma se ha tornado un tanto complejo. Este proceso de adopción se puede planificar empleando algún modelo de madurez para SOA.

El presente trabajo pretende el desarrollo de una aplicación informática que permita la gestión y aplicación de un modelo de madurez para SOA, dicho modelo fue propuesto a través de una investigación por el centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE) perteneciente a la Universidad de las Ciencias Informáticas (UCI).

Índice de Contenido

Índice de Figuras	10
Índice de Tablas.....	11
Índice de Anexos	13
Introducción	14
Capítulo#1 Fundamentación Teórica	18
Introducción	18
Conceptos asociados al objeto de estudio del problema	18
Arquitectura Empresarial (AE)	18
SOA.....	19
Complejidad de las Iniciativas Empresariales basadas en SOA y sus fracasos	19
Modelo de madurez.....	20
Descripción del modelo de madurez propuesto por el CDAE	21
Estructura de datos para representar el modelo.....	21
Análisis de aplicaciones existentes.....	22
Opina	22
PollDaddy.....	23
EncuestaTick	24
Necesidad del desarrollo de una nueva aplicación	24
Ventajas y desventajas de las aplicaciones de escritorio y web.....	25
Herramientas, tecnologías y metodología a utilizar	26
Lenguaje de programación.....	26
Entorno de Desarrollo Integrado	26
Librería Swing.....	27
Sistema Gestor de Bases de Datos PostgreSQL.....	27
JDBC.....	28
Metodología del Desarrollo del Software	29
Herramientas CASE	29

UML.....	30
Conclusiones parciales del capítulo.....	31
Capítulo#2 Exploración y Planificación	32
Introducción	32
Reglas del negocio.....	32
Roles de Usuarios	32
Exploración	32
Historias de Usuarios	33
Arquitectura del sistema (arquitectura dividida en tres capas).....	38
Planificación.....	40
Estimación del tiempo por las historias de usuarios	41
Plan de Iteraciones	41
Iteración 1	42
Iteración 2	42
Iteración 3	42
Plan de duración de las iteraciones.....	42
Plan de Entregas	43
Diseño del Sistema.....	44
Tarjeta CRC.....	44
Patrones de diseño.....	47
Patrones GRASP (Patrones de Software para la asignación General de Responsabilidades).....	47
Conclusiones parciales del capítulo	50
Capítulo#3 Implementación y prueba.....	51
Introducción	51
Iteraciones en el desarrollo del sistema.....	51
Iteración 1	51
Tareas de las historias de usuarios pertenecientes a la primera iteración.....	52
Iteración 2	59

Tareas de las Historias Usuarios pertenecientes a la segunda iteración	59
Iteración 3	61
Tareas de las Historias Usuarios pertenecientes a la tercera iteración	61
Pruebas	63
Pruebas de Aceptación.....	64
Conclusiones parciales del capítulo.....	68
Conclusiones generales del trabajo realizado	69
Referencia Bibliográficas	70
Anexos.....	72

Índice de Figuras

Figura 1 Áreas de medidas del modelo de madurez propuesto por el CDAE	21
Figura 2 Representación del modelo propuesto en la estructura árbol	22
Figura 3 Gestor de encuestas Opina.....	22
Figura 4 Interfaz de Opina.....	23
Figura 5 Gestor de encuestas PollDaddy	23
Figura 6 Gestor de encuestas Encuesta Tick	24
Figura 7 Prototipo de interfaz para la historia de usuario Gestionar Modelo	34
Figura 8 Prototipo de interfaz al aplicar una evaluación.....	35
Figura 9 Prototipo de interfaz para la historia de usuario Consultar evaluación	37
Figura 10 Arquitectura dividida en tres capas	38
Figura 11 Diseño estructural de la base de datos	40
Figura 12 Diagrama de clase en donde se evidencia la utilización del patrón Creador.....	48
Figura 13 Diagrama de clase en donde se evidencia la utilización del patrón Controlador	49

Índice de Tablas

Tabla 1 Ventajas y desventajas de aplicaciones web y escritorio	25
Tabla 2 Principales clases para el uso de JDBC	29
Tabla 3 Historia de usuario Gestionar Modelo	34
Tabla 4 Historia de usuario Aplicar evaluación	35
Tabla 5 Historia de usuario Consultar evaluación	37
Tabla 6 Estimación de tiempo por historia de usuario	41
Tabla 7 Plan de duración de las iteraciones	43
Tabla 8 Plan de entrega	43
Tabla 9 Tarjeta CRC de la clase Principal	45
Tabla 10 Tarjeta CRC de la clase Consulta	46
Tabla 11 Tarjeta CRC de la clase Conexión	46
Tabla 12 Historias de usuarios abordadas en la primera iteración	51
Tabla 13 Primera tarea asociada a la historia de usuario Gestionar Modelo	52
Tabla 14 Segunda tarea asociada a la historia de usuario Gestionar Modelo	52
Tabla 15 Tercera tarea asociada a la historia de usuario Gestionar Modelo	52
Tabla 16 Primera tarea asociada a la historia de usuario Gestionar Área de Medida	53
Tabla 17 Segunda tarea asociada a la historia de usuario Gestionar Área de Medida	53
Tabla 18 Tercera tarea asociada a la historia de usuario Gestionar Área de Medida	53
Tabla 19 Primera tarea asociada a la historia de usuario Gestionar Dimensión	54
Tabla 20 Segunda tarea asociada a la historia de usuario Gestionar Dimensión	54
Tabla 21 Tercera tarea asociada a la historia de usuario Gestionar Dimensión	54
Tabla 22 Primera tarea asociada a la historia de usuario Gestionar Categoría	55
Tabla 23 Segunda tarea asociada a la historia de usuario Gestionar Categoría	55
Tabla 24 Tercera tarea asociada a la historia de usuario Gestionar Categoría	55
Tabla 25 Tabla 24 Primera tarea asociada a la historia de usuario Gestionar Atributo Cualitativo	56
Tabla 26 Segunda tarea asociada a la historia de usuario Gestionar Atributo Cualitativo	56
Tabla 27 Tercera tarea asociada a la historia de usuario Gestionar Atributo Cualitativo	56
Tabla 28 Primera tarea asociada a la historia de usuario Gestionar Pregunta	57
Tabla 29 Segunda tarea asociada a la historia de usuario Gestionar Pregunta	57
Tabla 30 Tercera tarea asociada a la historia de usuario Gestionar Pregunta	57
Tabla 31 Primera tarea asociada a la historia de usuario Gestionar Valor	58
Tabla 32 Segunda tarea asociada a la historia de usuario Gestionar Valor	58
Tabla 33 Primera tarea asociada a la historia de usuario Gestionar Peso	58
Tabla 34 Segunda tarea asociada a la historia de usuario Gestionar Peso	59
Tabla 35 Historias de Usuarios abordadas en la segunda iteración	59

Tabla 36 Primera tarea asociada a la historia de usuario Realizar Evaluación.....	59
Tabla 37 Segunda tarea asociada a la historia de usuario Realizar Evaluación.....	60
Tabla 38 Tercera tarea asociada a la historia de usuario Realizar Evaluación.....	60
Tabla 39 Primera tarea asociada a la historia de usuario Consultar Evaluación.....	60
Tabla 40 Historias de Usuarios abordadas en la tercera iteración.....	61
Tabla 41 Primera tarea asociada a la historia de usuario Gestionar Usuario.....	61
Tabla 42 Segunda tarea asociada a la historia de usuario Gestionar Usuario.....	61
Tabla 43 Tercera tarea asociada a la historia de usuario Gestionar Usuario.....	62
Tabla 44 Primera tarea asociada a la historia de usuario Autenticar Usuario.....	62
Tabla 45 Segunda tarea asociada a la historia de usuario Autenticar Usuario.....	62
Tabla 46 Primera tarea asociada a la historia de usuario Gestionar Nivel.....	63
Tabla 47 Segunda tarea asociada a la historia de usuario Gestionar Nivel.....	63
Tabla 48 Prueba de aceptación sobre la historia de usuario Gestionar Modelo.....	66
Tabla 49 Prueba de aceptación sobre la historia de usuario Realizar Evaluación.....	67
Tabla 50 Prueba de aceptación sobre la historia de usuario Consultar Evaluación.....	67

Índice de Anexos

Anexo 1 Historia de usuario Gestionar Área de Medida	72
Anexo 2 Historia de usuario Gestionar Dimensión.....	72
Anexo 3 Historia de usuario Gestionar Categoría	73
Anexo 4 Historia de usuario Gestionar Atributo Cualitativo	73
Anexo 5 Historia de usuario Gestionar Pregunta	73
Anexo 6 Historia de usuario Gestionar Valor	74
Anexo 7 Historia de usuario Gestionar Peso	74
Anexo 8 Historia de usuario Gestionar Usuario.....	74
Anexo 9 Historia de usuario Autenticar	75
Anexo 10 Historia de usuario Gestionar Nivel	75
Anexo 11 Interfaz principal.....	76
Anexo 12 Interfaz gestionar peso	77
Anexo 13 Interfaz modificar área de medida.....	77
Anexo 14 Interfaz modificar dimensión	78

Introducción

Las empresas necesitan poder interconectar los procesos, personas e información, aumentar los canales de interacción con el cliente y mejorar su experiencia y satisfacción, lograr diferencias competitivas y agilidad del negocio, desarrollar servicios a través de la utilización efectiva de nuevas tecnologías, así como controlar la complejidad de las regulaciones.

La falta de integración entre sus componentes, aplicaciones y datos, hace difícil obtener una respuesta rápida y efectiva ante los cambios que afectan de forma natural a los negocios. La inflexibilidad genera costos, reduce la capacidad de respuesta ante los clientes, compromete el cumplimiento con la normativa legal y afecta negativamente a la productividad de los empleados. En fin, una deficiente integración es uno de los problemas más importantes a los que las organizaciones deben hacer frente para mantener su competitividad y garantizar su crecimiento.

La Arquitectura Orientada a Servicios, por sus siglas en inglés (SOA) supone una estrategia general de organización de los elementos de infraestructura tecnológica, de forma que una colección abigarrada de sistemas distribuidos y aplicaciones complejas se pueda transformar en una red de recursos integrados, simplificada y sumamente flexible. SOA permite cambios más rápidos en los procesos, lo que brinda ventaja competitiva al crear nuevas aplicaciones dinámicas, ayuda a la automatización de los Procesos de Negocio, permite la integración de aplicaciones, la modernización incremental de Sistemas Legados, así como la consolidación de la información.

Si bien una adopción de SOA bien planificada y ejecutada puede mejorar la capacidad de respuesta de las organizaciones, no todos los esfuerzos de orientación a servicios han resultado satisfactorios. El desarrollo de un programa SOA ha resultado complejo y en muchas ocasiones no tributa a los beneficios esperados. Buena parte de las iniciativas han fracasado y se identifica al factor humano como una de las principales causas. Es por ello que del modo en que se adopta este paradigma sigue siendo un factor determinante.

La adopción de SOA puede planificarse empleando algún modelo de madurez. Estos han surgido para ayudar a las organizaciones a saber hasta dónde han avanzado en el camino de adopción de SOA. Forman la base para comunicar y extender capacidades; ayudan en la construcción de planes. Los modelos de madurez para SOA constituyen un método para evaluar y medir la madurez de la adopción de SOA en las organizaciones, además describen las capacidades de una arquitectura SOA efectiva y en particular brinda una visión de dónde se debe mejorar. Dado que SOA en sí es todavía un conjunto de

buenas prácticas emergentes, el principal problema con estos modelos es que todos ellos tienen diferentes puntos de vista sobre la cuestión de la madurez de SOA. Para ellos todas las dimensiones que evalúan la madurez tienen el mismo peso y no son evidentes las relaciones de dependencia que entre estas existen, pues algunas áreas necesitan tener cierto grado de madurez para garantizar el avance de otras, lo que en muchos casos se suma a generar una mayor confusión sobre SOA.

Como respuesta a esta situación, el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE) perteneciente a la Universidad de las Ciencias Informáticas (UCI), a partir de una investigación realizada en estos temas, ha definido la estructura que debe tener un nuevo modelo de madurez para SOA que resuelva algunas deficiencias de modelos anteriores.

Dicha estructura está enfocada a definir un modelo flexible, que permita adaptarse con facilidad a las características de las organizaciones donde se aplica. Define los elementos que evalúan aspectos relevantes de una iniciativa SOA, así como un mecanismo para determinar el nivel de madurez de la iniciativa asociado al cálculo de indicadores. Lo hasta ahora definido, consiste en la concepción estructural de una base de conocimientos. La información almacenada en el modelo constituye un conjunto de buenas prácticas asociadas al desarrollo de una iniciativa SOA y que pueden ser empleadas para tomar decisiones acertadas en el proceso de adopción de SOA.

El CDAE, no cuenta con ninguna herramienta informática que posibilite el proceso de creación del modelo de madurez para SOA en su investigación a partir de la estructura definida. Además, de no contar, con una herramienta que permita la implantación del modelo de madurez para SOA definido en alguna organización.

Por lo anterior se define el siguiente **problema**: ¿Cómo gestionar con flexibilidad el modelo de madurez para SOA propuesto por el CDAE?

Para solucionar el problema planteado se define como **objetivo general** del presente trabajo de diploma: Construir una solución de software que permita gestionar con flexibilidad el modelo de madurez para SOA definido por el CDAE.

La siguiente **idea a defender** plantea una posible solución al problema: La gestión del modelo de madurez para SOA definido por el CDAE será más flexible si se realiza a través de una solución de software.

El **objeto de estudio**, se enmarca en las soluciones informáticas para la evaluación de indicadores.

Se define como **campo de acción**, la automatización del modelo de madurez para SOA definido por el CDAE.

Objetivos específicos:

- Realizar un estudio del arte sobre las aplicaciones que permitan gestionar el desarrollo e implantación de encuestas.
- Realizar el levantamiento de requerimientos y objetivos de la aplicación informática.
- Diseñar los prototipos de interfaz y la arquitectura del software.
- Realizar un análisis y diseño detallado del sistema.
- Realizar la implementación del sistema.
- Validar la solución de software propuesta.

Tareas de la investigación:

- Estudio de los procedimientos para la gestión de un modelo de madurez para SOA.
- Realización de búsquedas en diferentes bibliografías acerca de la existencia de sistemas informatizados que realicen la gestión de sistemas evaluadores de indicadores como los sistemas encuestas.
- Analizar las funcionalidades que debe brindar la aplicación informática.
- Selección y estudio de metodologías y herramientas posibles a utilizar para la gestión del sistema.
- Escribir en formato digital todo el proceso investigativo del desarrollo del trabajo como resultado de la experiencia obtenida.
- Diseñar e implementar el software para la gestión del modelo de madurez para SOA definido por el CDAE.
- Evaluar el software desarrollado a través del cliente.

Para la realización de estas tareas se contó con el apoyo de los siguientes métodos científicos de la investigación. Estos se dividen en dos grupos: los métodos teóricos y los empíricos.

Métodos Teóricos

Consulta Bibliográfica: Este método permitió consultar bibliografía referente al proceso de creación de aplicaciones informáticas de escritorio, para tener un mejor conocimiento del tema. Se revisaron además modelos de madurez existentes para identificar las características y atributos necesarios a incluir en el modelo.

Analítico – Sintético: Este método permitió analizar teorías y documentos sobre los modelos de madurez para SOA, profundizar en sus características fundamentales y ventajas, que servirán de ayuda para el desarrollo de la investigación y llegar a conclusiones.

Histórico-Lógico: Se utilizó para realizar un estudio sobre las aplicaciones informáticas que permitieran el desarrollo y aplicación del modelo de madurez propuesto por el CDAE.

Métodos Empíricos

Entrevista: Este método jugó un papel fundamental en la definición del problema a resolver de la investigación.

Como **aporte práctico** se genera un sistema informático que permite gestionar de manera flexible, el proceso de creación de un modelo de madurez para SOA en constante evolución y adaptación.

Estructura en capítulos del documento

En el presente documento se aborda la investigación cuyo diseño metodológico se ha descrito. Su desarrollo comienza con un capítulo que describe la fundamentación teórica de dicha investigación, el cual incluye un estudio del estado del arte de los sistemas gestores de encuestas, y una descripción de las herramientas disponibles y los criterios que se tuvieron en cuenta para seleccionar las que se utilizaron en el desarrollo de la solución propuesta. Se incluye un capítulo que describe el proceso de construcción de la solución; el cual comienza con la descripción, priorización y planificación de las historias de usuario que describen las funcionalidades a implementar; pasando luego a detallar la arquitectura de la solución y sus principales características. Se concluye con un capítulo en el cual se describen los métodos utilizados para validar la solución desarrollada.

Capítulo#1 Fundamentación Teórica

Introducción

En el presente capítulo se presentan los elementos teóricos que sirven de base a la investigación del problema planteado. Se estudian las soluciones de software que sirvan de ayuda para gestionar la creación de modelos de madurez para SOA a través de un estado del arte correspondiente a estos, dado que para aplicar una evaluación a alguna institución sobre la adopción de SOA mediante un modelo de madurez es necesario responder un número de preguntas anteriormente establecidas en el modelo. La solución informática propuesta, tendrá funcionalidades que se comportarán muy similares a las principales funcionalidades de un sistema de encuesta, por lo que el estudio del arte estará encaminado a las Aplicaciones de Encuestas, así como a los Generadores de encuestas disponibles. Además, el presente capítulo propone un estudio de las tecnologías, metodología y herramientas a utilizar en el desarrollo de la solución.

Conceptos asociados al objeto de estudio del problema

Arquitectura Empresarial (AE)

El concepto de arquitectura empresarial tiene su origen en el año de 1987 con la publicación del artículo de J. Zachman en el Diario IBM Systems, titulado «Un marco para la arquitectura de sistemas de información.» [1]. Surge la AE como disciplina que describe la forma en que la tecnología y la información soportan las operaciones del negocio y aportan beneficios directos a este [2]. Para ello, identifica y modela los componentes funcionales de una organización (los sistemas de información, procesos, unidades organizacionales y las personas de una organización) y cómo se relacionan operando como un todo [3]. La AE provee mecanismos que permiten la evolución de la Arquitectura, asegurando la gobernabilidad del programa, la reutilización de los activos creados y asegurando la coherencia con las metas de la organización [4].

La AE nace como una disciplina que evoluciona desde modelos administrativos y de gestión, como la teoría organizacional y la teoría de sistemas. En una organización corresponde a la forma de representar de manera integral la empresa, permitiendo cubrir y considerar todos y cada uno de los elementos que la conforman.

SOA

SOA supone una estrategia general de organización de los elementos de infraestructura tecnológica, permitiendo que una colección de sistemas dispersos y aplicaciones complejas, se puedan transformar en una red simplificada, flexible y gobernada de recursos integrados, convirtiéndose en un medio idóneo para lograr la interoperabilidad. La creación de servicios y aplicaciones compuestas, que pueden existir con independencia de las tecnologías subyacentes y su capacidad para ser empleados en contextos diferentes, favorece que la mayor parte posible de la funcionalidad del conjunto de las infraestructuras tecnológicas de una organización sea reutilizable, generando mayor eficiencia en las operaciones de infraestructura tecnológica [5].

Antes de SOA: En las arquitecturas de infraestructura tecnológica tradicionales, las actividades del proceso de negocios, las aplicaciones y los datos con frecuencia están encerrados en "silos" independientes e incompatibles que son caros de mantener y dejan a los usuarios la necesidad de navegar entre redes, aplicaciones y bases de datos independientes para realizar tareas de negocios completas[6].

Después de SOA: Con una SOA, los usuarios ya no tienen que iniciar sesión en varios sistemas, buscar los datos relevantes e integrar los resultados manualmente. Los datos de las actividades de los procesos de negocios se entregan como un servicio integrado, en una sola aplicación, en una sola pantalla, con un solo inicio de sesión[6].

SOA es una nueva arquitectura de software que al aplicarse se obtiene un sistema informático que resuelve los problemas de integración en la empresa. Constituye un lenguaje de modelado de software que nos permite representar los conceptos del negocio que responden a los procesos de la empresa. Es un paradigma o estilo arquitectónico que se basa en la creación de un conjunto de servicios, de diferente granularidad, entre los procesos de negocio y las aplicaciones.

Complejidad de las Iniciativas Empresariales basadas en SOA y sus fracasos

La adopción de SOA en toda la organización puede tomar años y los beneficios pueden tardar más tiempo para apreciarse en su totalidad [7]. Por esto, aun cuando es evidente que un programa que no logra obtener el ROI (Retorno de Inversión) esperado puede considerarse como fallido, en los programas SOA no puede esperarse tanto para notar que el programa no marcha bien [8]. Determinar el fracaso o el éxito

de una Iniciativa Empresarial basada en SOA probablemente sea mucho más difícil que para otros tipos de proyectos.

Una SOA fallida puede definirse como el fracaso del conjunto de prácticas que las organizaciones adoptan y que no tienen el impacto esperado referentes a la mejora en la interoperabilidad, simplificación del ambiente de infraestructura tecnológica, reducción de costos, incremento en la velocidad del desarrollo de aplicaciones, creación de mayor flexibilidad y en cambio aumenta la complejidad del ambiente tecnológico, aumentan los costos, no se hace mucho para salvar las incompatibilidades entre las diferentes plataformas o se ata la organización a un proveedor de tecnología específico [8].

Fuentes provenientes del sector empresarial ubican el fracaso de las iniciativas SOA en el rango del 50% al 70 % [9]. Un estudio hecho a 20 compañías por el Burton Group en 2008, mostró que el fracaso fue del 50% y del resto, un 30% no se consideró fracasada la iniciativa pero tampoco exitosa [10].

Para muchas organizaciones ha resultado demasiado complejo ejecutar con éxito estas iniciativas, en general: por la falta de experiencia, metodologías, habilidades SOA, ausencia de gobierno, fallo para involucrar al negocio, subestimar o ignorar el impacto del cambio, falta de liderazgo técnico y sobre las personas; falta de un fuerte patrocinio ejecutivo con poder de decisión, pobre gestión del alcance, de los requerimientos y de la comunicación entre los múltiples equipos que intervienen en la iniciativa. Además, ha influido negativamente crear expectativas irreales sobre la base de una planificación errónea, insuficiente inversión de tiempo y dinero, limitando los recursos necesarios para la iniciativa. No sustentar el desarrollo de SOA sobre la base de casos de negocio concretos y de impacto para la organización, débil gestión de la contratación sobre empresas consultoras que apoyan la iniciativa y no contar con métricas para evaluar el impacto de la iniciativa en el negocio o indicadores de desempeño para medir su progreso [11, 12].

Modelo de madurez

La complejidad inherente al desarrollo de una Iniciativa Empresarial basada en SOA, puede ser manejada por los modelos de Madurez, gracias el enfoque evolutivo que ofrecen. Estos permiten planificar una hoja de ruta a partir de medidas objetivas del estado inicial de la organización y del que esta desea obtener [5, 13]. Cuando una organización desarrolla o utiliza un modelo de madurez de SOA, debe buscar una lista de características o capacidades que su arquitectura podría exhibir, asociándolas a obtener las soluciones

que requiere el negocio. El objetivo de esto es identificar los elementos que faltan o que requieren ser adaptados en la arquitectura.

Descripción del modelo de madurez propuesto por el CDAE

El modelo de madurez para SOA propuesto por el CDAE, básicamente está estructurado por cuatro áreas de medida orientadas a reflejar las capacidades que la organización crea para desarrollar y sustentar la arquitectura[26] (ver figura 1).

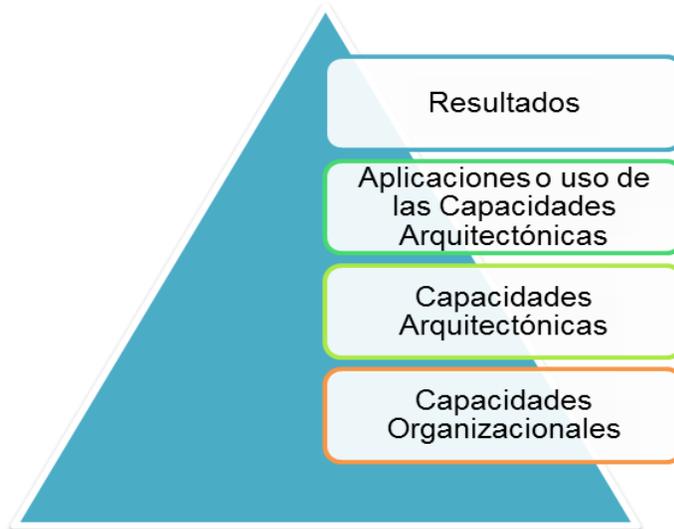


Figura 1 Áreas de medidas del modelo de madurez propuesto por el CDAE

El modelo de madurez propuesto por el CDAE está estructurado por una serie de elementos comprendidos entre sí. Para una mejor comprensión del mismo hace necesario la representación del mismo en alguna estructura de dato que sea lo suficientemente comprensible para cualquier persona.

Estructura de datos para representar el modelo

Para la representación del modelo de madurez para SOA propuesto por CDAE se decide utilizar la estructura de datos denominada árbol. Esta estructura ofrece las facilidades necesarias para hacer el trabajo más flexible y tener una mejor visión de la estructura del modelo propuesto (ver figura 2).

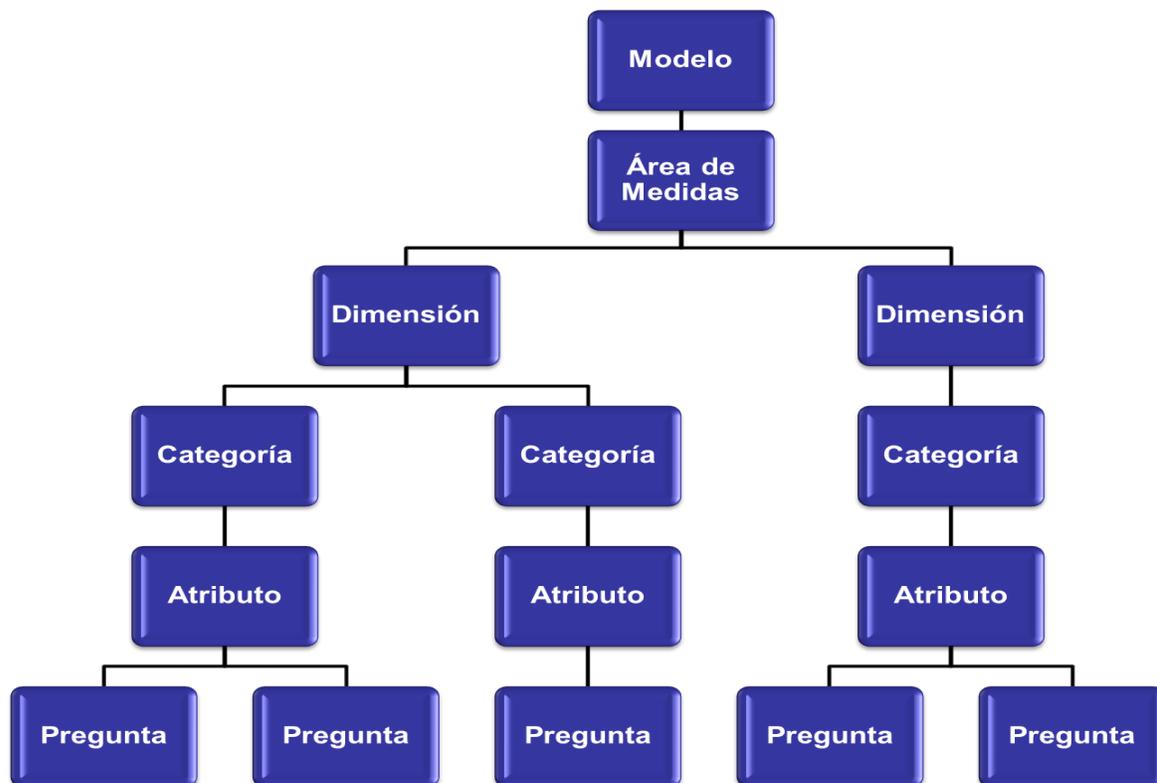


Figura 2 Representación del modelo propuesto en la estructura árbol

Análisis de aplicaciones existentes

En nuestros días se pueden encontrar diversas aplicaciones que representen soluciones afines a programas para el desarrollo y utilización de encuestas. En este epígrafe se realiza un análisis sobre las aplicaciones informáticas más utilizadas a nivel mundial para la gestión de encuestas, que aportarán conocimiento para la realización del presente trabajo, dado que al aplicarse algún modelo de madurez de SOA en alguna institución el proceso de evaluación se comporta muy similar al proceso que se desarrolla cuando se realiza alguna encuesta digital.

Opina



Figura 3 Gestor de encuestas Opina

Esta aplicación, permite modelar cuestionarios y/o encuestas. Opina ha sido desarrollada para la Universidad de Sevilla y quedan reservados todos los derechos. Además, posibilita la realización desde cualquier punto con acceso a internet (web) de cuestionarios, gestionar usuarios (encuestados), realizar modificaciones, así como parametrizar y configurar todas las opciones de las que consta.

La interfaz de Opina está basada en escritorios. Estos escritorios intentan recrear a los conocidos escritorios de los Sistemas Operativos. Estos escritorios están compuestos de los siguientes elementos principales:

- ✓ Sistema de navegación por escritorios.
- ✓ Sistema de menús.
- ✓ Área de trabajo.



Figura 4 Interfaz de Opina

Opina gestiona distintos tipos de preguntas entre las que se encuentran las numéricas, las de elección múltiple, las preguntas abiertas. Esta aplicación exporta los resultados obtenidos en la encuesta en formato Microsoft Excel.

PollDaddy



Figura 5 Gestor de encuestas PollDaddy

Entre las bondades de PollDaddy se encuentran el permitir crear encuestas personalizadas, la interfaz de creación de encuestas es muy intuitiva, las encuestas son visualmente bonitas, permite escoger los temas

visuales de las encuestas o crear nuevos temas, ofrece la posibilidad de escoger entre 20 tipos diferentes de estilos, además permite insertar preguntas de diferentes tipos (selección múltiple con única o múltiple respuesta, preguntas de información personal, de información de contacto, generación de preguntas abiertas, entre otras múltiples posibilidades), tiene un excelente tutorial paso a paso, permite incrustar la encuesta en un sitio web o en un acceso directo por url¹, los reportes de resultados y estadísticas son presentados en tiempo real además de ser muy completos y pueden ser mostrados o mantenerlos privados según sea la elección.

EncuestaTick



Figura 6 Gestor de encuestas Encuesta Tick

Para su utilización es necesario registrarse en www.encuestatick.com o www.portaldeencuestas.com, esta aplicación brinda tres posibilidades de crear una encuesta:

- ✓ Desde cero.
- ✓ Editando una plantilla ya prediseñada.
- ✓ Duplicando una encuesta propia y hacer las modificaciones necesarias.

Estas encuestas, pueden mandarse por correo o publicarlas en algún sitio web mediante un link. Las respuestas se guardan al instante y se puede empezar a ver resultados, estos resultados mostrarán los porcentajes de las respuestas, dando la posibilidad de filtrar los resultados por pregunta, además de ver gráficos y estadísticas por pregunta o grupo de pregunta, y descargarlo en pdf².

Necesidad del desarrollo de una nueva aplicación

Luego de un detenido y profundo análisis se llega a la conclusión de que las aplicaciones informáticas estudiadas representan una fuente de valor para la investigación y sirven de sostén para la elaboración de la misma, aportando conocimiento sobre el tipo de pregunta que se debería de gestionar en el modelo y el tipo de respuestas que deberían dárseles. No obstante, no se toman como soluciones ya que para la

¹ Localizador de recursos uniforme.

² Formato de documento portátil, es un formato de almacenamiento de documentos.

gestión y aplicación de un modelo de madurez para SOA se requieren de otras funcionalidades las cuales no pueden ser desarrolladas por estos sistemas. Además, como fue anteriormente descrito, para la visualización de la estructura del modelo de madurez para SOA propuesto por el CDAE se hace uso de la estructura de dato denominada árbol. Lo cual constituye un aspecto fundamental en la solución para una mejor comprensión del modelo de madurez propuesto y estos sistemas no permiten dicha funcionalidad.

Ventajas y desventajas de las aplicaciones de escritorio y web

Aplicación Web	Aplicación de Escritorio
Ventajas	
<ul style="list-style-type: none"> ➤ Se puede acceder a la aplicación desde cualquier estación en la red. ➤ No hay problemas de incompatibilidad entre versiones, porque todos trabajan con la misma. ➤ Se centralizan los procesos. ➤ No necesita instalaciones adicionales sólo una computadora con sistema operativo y navegador instalados. ➤ Es independiente de versión o fabricante del sistema operativo. 	<ul style="list-style-type: none"> ➤ Tiempo de respuesta más rápido. ➤ Pueden ser más robustas, incorporando variados modelos de seguridad y diseño. ➤ Es mejor opción cuando es necesario realizar numerosos cálculos. ➤ Centralización de los datos, tanto la aplicación como los datos se encuentran en un mismo lugar y no están en lugares diferentes. ➤ El procesamiento es interno a la máquina y la aplicación no transfiere todo el tiempo información hacia y desde el servidor, lo que tributa a la privacidad.
Desventajas	
<ul style="list-style-type: none"> ➤ Requiere conexión a la red. ➤ Se pierde tiempo de desarrollo haciéndola compatible con los distintos navegadores. 	<ul style="list-style-type: none"> ➤ Falta de portabilidad de la aplicación a diferentes sistemas operativos. ➤ Retrasos a la hora de realizar actualizaciones o correcciones al programa.

Tabla 1 Ventajas y desventajas de aplicaciones web y escritorio

Teniendo en cuenta las necesidades del cliente, la elección del desarrollo de una aplicación de escritorio es la respuesta más factible al proceso de gestionar de manera flexible la información referente al nuevo

modelo de madurez para SOA. Este tipo de aplicación promete un trabajo eficiente y acorde a las tecnologías con que cuenta el CDAE. Un aspecto esencial que tributa a la selección de esta aplicación es que su proceso de desarrollo es mucho menor que el proceso de desarrollo de una aplicación web y al contar con un estrecho margen de tiempo para su implementación es necesario su utilización. El sistema debe de realizar una gran cantidad de cálculos por lo que esta condición refuerza la adopción de una aplicación de escritorio y además se consideró la condición de que al sistema solamente accederá una sola persona lo que no necesariamente debe estar conectado a la red.

Herramientas, tecnologías y metodología a utilizar

Lenguaje de programación

Se elige al lenguaje Java como lenguaje de programación ya que el mismo elimina la complejidad de los lenguajes como "C" y "C++", da paso al contexto de los lenguajes orientados a objetos. Maneja la memoria de la computadora, por lo que el programador no debe de preocuparse por apuntadores, memoria que no se esté utilizando. Java realiza todo estas acciones sin necesidad que el programador se lo indique. Tiene ciertas políticas que evitan que se puedan codificar virus con este lenguaje. Como el código compilado de Java (conocido como bytecode) es interpretado, un programa compilado de Java puede ser utilizado por cualquier computadora que tenga instalado el intérprete de Java. Este código es interpretado por diferentes computadoras de igual manera, solamente hay que instalar un intérprete para cada plataforma. De esa manera Java logra ser un lenguaje que no depende de una arquitectura computacional disponible. Además, en Java no se requiere que se compilen todas las clases de un programa para que este funcione. Al realizar una modificación a una clase, Java se encarga de realizar un Dynamic Binding o un Dynamic Loading para encontrar las clases. Por último, debido a la Máquina Virtual de Java (JVM) las aplicaciones desarrolladas en Java funcionan en Linux, Windows, Mac OS, y en cualquier sistema operativo para el cual exista una JVM, garantizando que no se dependa de un único sistema operativo para el funcionamiento de nuestra solución.

Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado (IDE siglas en inglés) es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Estas herramientas pueden estar pensadas para su utilización en un único lenguaje de programación o para varios. Como IDE se elige

Netbeans, ya que el mismo, soporta el desarrollo de todos los tipos de aplicaciones Java. Este IDE es software libre y gratuito, intuitivo y muy fácil de utilizar, y en particular es rápido para aplicaciones de escritorio. Brindando todas las características anteriormente mencionadas, una gran ventaja al programador en el desarrollo de la solución propuesta.

Librería Swing

La librería Swing de Java proporciona todos los componentes necesarios para la construcción de la interfaz gráfica de usuario. Los componentes Swing frente a los componentes AWT (llamados componentes pesados) presentan la ventaja de ser independientes de la plataforma, por eso reciben el nombre de componentes ligeros. Los componentes Swing son más numerosos que los AWT (por ejemplo, fichas, paneles con scroll, árboles). Swing basa sus componentes en la arquitectura modelo-vista-controlador (MVC).[16]

Los componentes Swing se pueden anidar unos dentro de otros y permiten crear interfaces más atractivas y con mayor funcionalidad que los componentes AWT. [16]

Por estas razones ha sido escogida para la creación de la interfaz gráfica de la presente aplicación, permitiendo realizar interfaces amigables³ y altamente funcionales.

Sistema Gestor de Bases de Datos PostgreSQL

PostgreSQL es un Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos. Es un gestor de bases de datos de código abierto, brinda un control de concurrencia multi-versión (MVCC por sus siglas en inglés) que permite trabajar con grandes volúmenes de datos; soporta gran parte de la sintaxis SQL y cuenta con un extenso grupo de enlaces con lenguajes de programación, entre el que se encuentra Java. Posee características significativas del motor de datos, entre las que se pueden incluir las subconsultas, los valores por defecto, las restricciones a valores en los campos (constraints) y los disparadores (triggers). El código fuente se encuentra disponible para todos sin costo alguno. Posee una integridad referencial e interfaces nativas para lenguajes como ODBC, JDBC, C, C++, PHP, PERL, TCL, ECPG, PYTHON y RUBY. Funciona en todos los sistemas operativos Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. Por todas estas características se decide su utilización en la solución planteada.

³ Entendible y fácil de usar.

JDBC

Java Database Connectivity, más conocida por sus siglas JDBC, es una API⁴ que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

Un manejador de conexiones hacia un modelo de base de datos en particular es un conjunto de clases que implementan las interfaces Java y que utilizan los métodos de registro para declarar los tipos de localizadores a base de datos url que pueden manejar. Para utilizar una base de datos particular, el usuario ejecuta su programa junto con la biblioteca de conexión apropiada al modelo de su base de datos, y accede a ella estableciendo una conexión, para ello provee el localizador a la base de datos y los parámetros de conexión específicos. A partir de allí puede realizar con cualquier tipo de tareas con la base de datos a las que tenga permiso: consulta, actualización, creación, modificación y borrado de tablas, ejecución de procedimientos almacenados en la base de datos, entre otras tareas.

⁴ Interfaz de programación de aplicaciones.

Clase	Descripción
DriverManager	Para cargar un driver.
Connection	Para establecer conexiones con las bases de datos.
Statement	Para Ejecutar sentencias SQL y enviarlas a las BD.
ResultSet	Para almacenar el resultado de la consulta.

Tabla 2 Principales clases para el uso de JDBC

Metodología del Desarrollo del Software

Una metodología de desarrollo de software es una serie de técnicas, herramientas y procedimientos que deben seguirse para desarrollar un software. Son utilizadas principalmente para establecer las etapas de un proyecto, decidir que tareas se llevarán a cabo en cada etapa y para controlar y gestionar un proyecto. Sirven de sostén para la creación de software.

XP (Extreme Programming) es la metodología seleccionada ya que sus características favorecen el desarrollo de la solución. Entre las cuales están: la realimentación continua entre el cliente y el equipo de desarrollo, la producción de versiones del sistema de manera rápida, el diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto y además, soporta el cambio de requerimientos.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software[19].

El ciclo de vida ideal al emplear la metodología XP consta de 6 fases[20], de las cuales se abordarán en el presente trabajo las primeras cuatro fases: exploración, planificación, iteración y producción.

Herramientas CASE

Las herramientas CASE representan una forma que permite modelar los procesos de negocios de las empresas y desarrollar los sistemas de información gerenciales[23]. Sin importar su arquitectura, tales herramientas deben poseer propiedades como las siguientes:

- Una interfaz gráfica y textual, que le permita al usuario manejar los objetos de diseño.
- Contar con un diccionario de datos, a fin de rastrear y controlar los objetos diseñados.

- Disponer de un conjunto de herramientas que permitan chequear las reglas del diseño y analizar su lógica.

La herramienta escogida tras una amplia labor investigativa fue Visual Paradigm, ya que es una de las herramientas CASE más completas que existe en el mercado actualmente, además de que la universidad posee la licencia para su utilización. Visual Paradigm soporta el ciclo de vida completo del desarrollo de software[24]: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML⁵ ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Además, contiene un conjunto de ayudas para el desarrollo de software, así como que permite la elaboración de todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación de diferentes extensiones.

UML

Utilizar un lenguaje de modelado permite verificar efectivamente el funcionamiento adecuado que el modelo debe poseer, pues existen menos medios de verificación, lo cual hace al modelo más sencillo que la propia programación.

El Lenguaje Unificado de Modelado (Unified Modeling Language - UML, por sus siglas en inglés) es un lenguaje estándar que permite especificar, visualizar, construir y documentar todos los elementos que forman un sistema de software[25] orientado a objetos. Por lo que su propósito consiste en elaborar los artefactos de un sistema a través de las distintas etapas de su ciclo de vida, principalmente durante el análisis y el diseño del mismo. El modelado mediante este lenguaje permite establecer un conjunto de requerimientos y estructuras necesarias para plasmar un sistema de software previo a la escritura del código. El tiempo invertido en el desarrollo de la arquitectura se minimiza, la trazabilidad y documentación del proyecto se realiza de una forma ordenada y guiada por los casos de uso. Pero lo más importante es la notable efectividad y productividad que se consigue en labores de diseño arquitectónico y mantenimiento, haciendo uso de UML frente a la realización de las mismas tareas, en ausencia de modelos. Por lo que se considera beneficiosa su utilización en la solución al problema.

⁵ Lenguaje Unificado de Modelado.

Conclusiones parciales del capítulo

En el presente capítulo, se ha plasmado un análisis de los conceptos y aspectos fundamentales asociados a la investigación. Se realizó un estudio del modelo de madurez para SOA propuesto por el CDAE para una mejor comprensión del mismo. Concluido una vez dicho estudio se define la manera en que se representará dicho modelo de madurez para que sea comprensible por cualquier persona. Partiendo del planteamiento, que al aplicar una evaluación a alguna empresa, haciendo uso de un modelo de madurez para SOA, el proceso que se desarrolla es muy similar al proceso que se desarrolla cuando se realiza alguna encuesta digital, se realizó un estudio de algunas aplicaciones que permitan la gestión y aplicación de encuestas. Este estudio brindó conocimientos que serán aplicados en el desarrollo de la solución propuesta. Además, se seleccionaron unas series de herramientas las cuales ofrecieran ventajas de su utilización para el grupo de desarrollo; y se definió la metodología de desarrollo a emplear.

Capítulo#2 Exploración y Planificación

Introducción

En el presente capítulo se especifican las reglas con que contará el negocio, así como los roles de usuarios que intervendrán en el sistema. Además, se abordan las fases de Exploración y Planificación propias de la metodología de desarrollo utilizada en la implementación del sistema informático, en las cuales se exponen las historias de usuario seccionada en iteraciones, los planes de entrega y la duración de las iteraciones durante el desarrollo del sistema.

Reglas del negocio

- La evaluación de una empresa deberá ser orientada por la dirección del centro.
- La empresa para evaluarse deberá presentar su nombre.
- La empresa para consultar alguna evaluación anterior deberá especificar la fecha en que obtuvo dicha evaluación.
- Solo tendrán acceso a la aplicación los usuarios con privilegio de administrador.

Roles de Usuarios

El administrador es el único rol existente, este tendrá acceso a todas las funcionalidades que propicia el sistema.

Exploración

La metodología de desarrollo XP comienza con la fase de exploración. Durante esta etapa se realiza el proceso de identificación de las historias de usuario, estas constituyen uno de los artefactos más importante que se generan en la metodología, pues son la forma en que se especifican los requisitos del sistema. El ciclo de vida de XP enfatiza en el carácter iterativo e incremental del desarrollo. Una iteración de desarrollo es un período de tiempo en el que se realiza un conjunto de funcionalidades determinadas, que en el caso de XP corresponden a un conjunto de historias de usuarios.

Historias de Usuarios

La historia de usuario, se escriben desde la perspectiva del cliente, aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas [27]. El contenido de estas debe ser concreto y sencillo [28]. Se realiza una por cada característica principal del sistema, es utilizada para hacer estimaciones de tiempo y para el plan de lanzamientos, reemplazan un gran documento de requisitos. Las historias de usuarios quedan estructuradas de la siguiente manera:

- ✓ **Nombre:** Nombre descriptivo de la Historia de Usuario.
- ✓ **Prioridad:** Grado de prioridad que le asigna el cliente a la Historia de Usuario en dependencia de sus necesidades. Los valores que puede tomar son (Alta, Media o Baja).
- ✓ **Complejidad:** Grado de complejidad que le asigna el equipo de desarrollo a la Historia de Usuario luego de analizarla. (Alta, Media o Baja).
- ✓ **Estimación:** Unidades de tiempo estimadas por el equipo de desarrollo para darle cumplimiento a la Historia de Usuario.
- ✓ **Iteración:** Número de la iteración en la cual será implementada la Historia de Usuario.
- ✓ **Descripción:** Descripción simple que brinda el cliente sobre lo que debe hacer la funcionalidad en cuestión.

Adicionalmente a cada historia de usuario se le asigna un número para facilitar su identificación por parte del equipo de desarrollo y se le puede incluir alguna información adicional que pueda ser útil para la comprensión de la misma. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas.[22]

Durante este proceso se identificaron 13 historias de usuarios, cada una de ellas respondiendo a las diferentes funcionalidades solicitadas por el cliente y dando una idea al resto del equipo de desarrollo de cómo debe ser su posterior implementación. A continuación se describen algunas de las historias de usuarios identificadas con el cliente, seguidas de un prototipo de interfaz, para consultar las restantes historias de usuario dirigirse al (**ANEXO #1**) y para ver los prototipos de interfaces ver el (**ANEXO #2**).

La principal funcionalidad que debe brindar el sistema es la de poder gestionar el modelo de madurez para SOA propuesto por el CDAE de manera flexible e intuitiva. Como fue anteriormente planteado, la estructura del modelo de madurez será representada mediante la estructura de dato árbol. El administrador podrá insertar, modificar y eliminar algún elemento del modelo de madurez, para el desarrollo de estas acciones debe escoger el elemento en el árbol.

Historia de Usuario	
Número: 1	Nombre: Gestionar Modelo
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 4	Iteración Asignada: 1
Descripción: El administrador podrá ir confeccionando la estructura del modelo de madurez.	
Observaciones:	

Tabla 3 Historia de usuario Gestionar Modelo

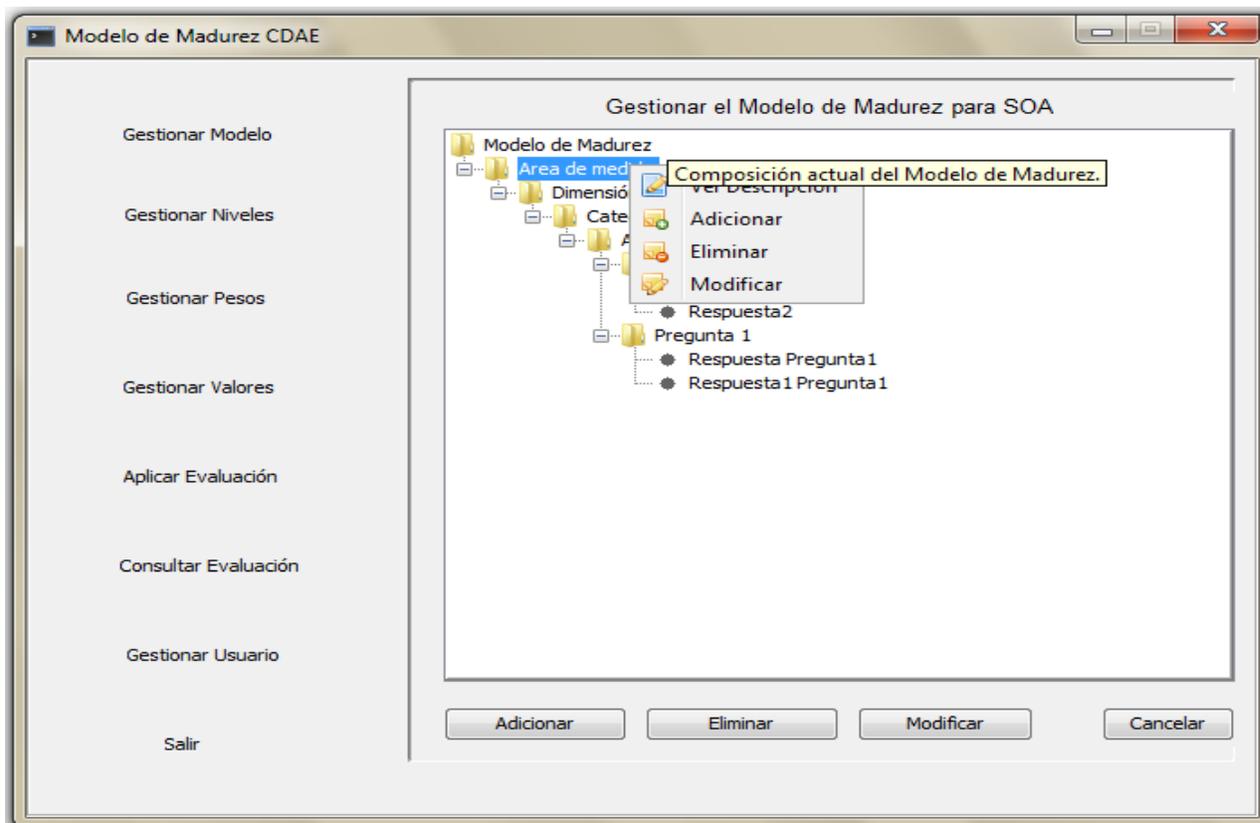


Figura 7 Prototipo de interfaz para la historia de usuario Gestionar Modelo

Una vez concluido el proceso de construcción del modelo de madurez para SOA, el sistema está listo para ser utilizado por el administrador para aplicar una evaluación a alguna determinada organización. Para el desarrollo de dicha acción es imprescindible que se especifique el nombre de la organización a la cual se le hará la evaluación y la fecha en que se comienza dicha evaluación. Además, el representante de dicha organización deberá responder las preguntas formuladas en el modelo de madurez.

Historia de Usuario	
Número: 9	Nombre: Aplicar evaluación
Usuario: Administrador	
Prioridad en Negocio: Medio	Riesgo en Desarrollo: Medio
Puntos Estimados: 4	Iteración Asignada: 2
Descripción: El administrador del sistema se encarga de aplicar una evaluación mediante el modelo de madurez para SOA a alguna organización.	
Observaciones:	

Tabla 4 Historia de usuario Aplicar evaluación

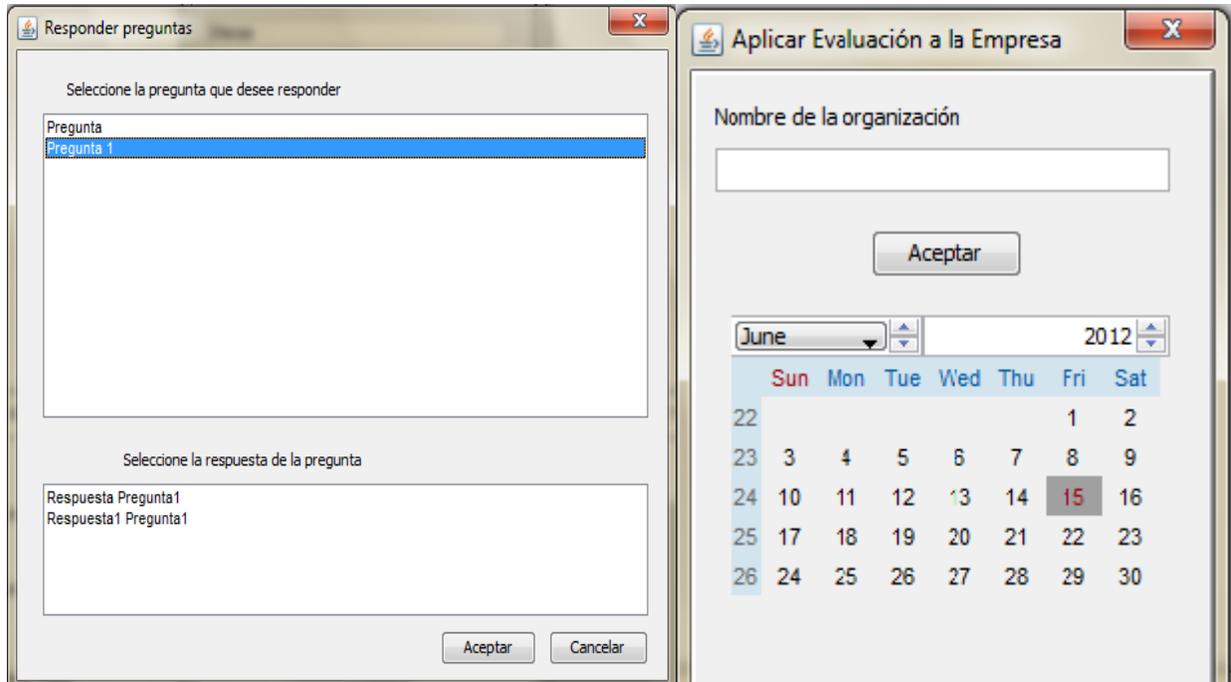


Figura 8 Prototipo de interfaz al aplicar una evaluación

Una vez concluida alguna evaluación a una organización, se podrá acceder a ver los resultados de la misma. Especificando para ello el nombre de la organización, la fecha de la evaluación y el nivel al que se desea ver los resultados obtenidos. El sistema le muestra los resultados de la evaluación escogida por el usuario en una tabla para una mejor comprensión de estos resultados.

Para determinar el nivel alcanzado por algún elemento del modelo de madurez para SOA propuesto, se hace necesario conocer el valor que obtuvo al concluir la evaluación para ello el sistema emplea el siguiente procedimiento.

Cálculo del valor de un Atributo Cualitativo

Para obtener el valor de un atributo primeramente se capturan las preguntas que le pertenecen y se va multiplicando el peso de la pregunta por el valor de la respuesta que le dio el evaluador, luego se suman todos los valores que arrojó la multiplicación y se obtiene finalmente el valor del atributo.

$$valor = \sum (PesoPregunta * ValorRespuesta)$$

Cálculo del valor de una Categoría

Para obtener el valor de una categoría primeramente se capturan los atributos cualitativos que pertenecen a la categoría y se va multiplicando el peso del atributo por su valor, luego se suman todos los valores que arrojó la multiplicación y se obtiene finalmente el valor de la categoría.

$$valor = \sum (PesoAtributo * ValorAtributo)$$

Cálculo del valor de una Dimensión

Para obtener el valor de una dimensión primeramente se capturan las categorías que pertenecen a la dimensión y se va multiplicando el peso de la categoría por su valor, luego se suman todos los valores que arrojó la multiplicación y se obtiene finalmente el valor de la dimensión.

$$valor = \sum (PesoCategoría * ValorCategoría)$$

Cálculo del valor de un Área de Medida

Para obtener el valor de un área de medida primeramente se capturan las dimensiones que pertenecen al área de medida y se va multiplicando el peso de la dimensión por su valor, luego se suman todos los valores que arrojó la multiplicación y se obtiene finalmente el valor del área de medida.

$$valor = \sum (PesoDimensión * ValorDimensión)$$

Al tener estos valores se pueden comparar con los que están definidos para determinar un nivel y tener conocimiento de la madurez en que se encuentra un área de medida, una dimensión, una categoría o un atributo cualitativo.

Historia de Usuario	
Número: 10	Nombre: Consultar evaluación
Usuario: Administrador	
Prioridad en Negocio: Medio	Riesgo en Desarrollo: Medio
Puntos Estimados: 2	Iteración Asignada: 2
Descripción: El administrador del sistema podrá consultar el resultado de las evaluaciones realizadas por el sistema.	
Observaciones:	

Tabla 5 Historia de usuario Consultar evaluación

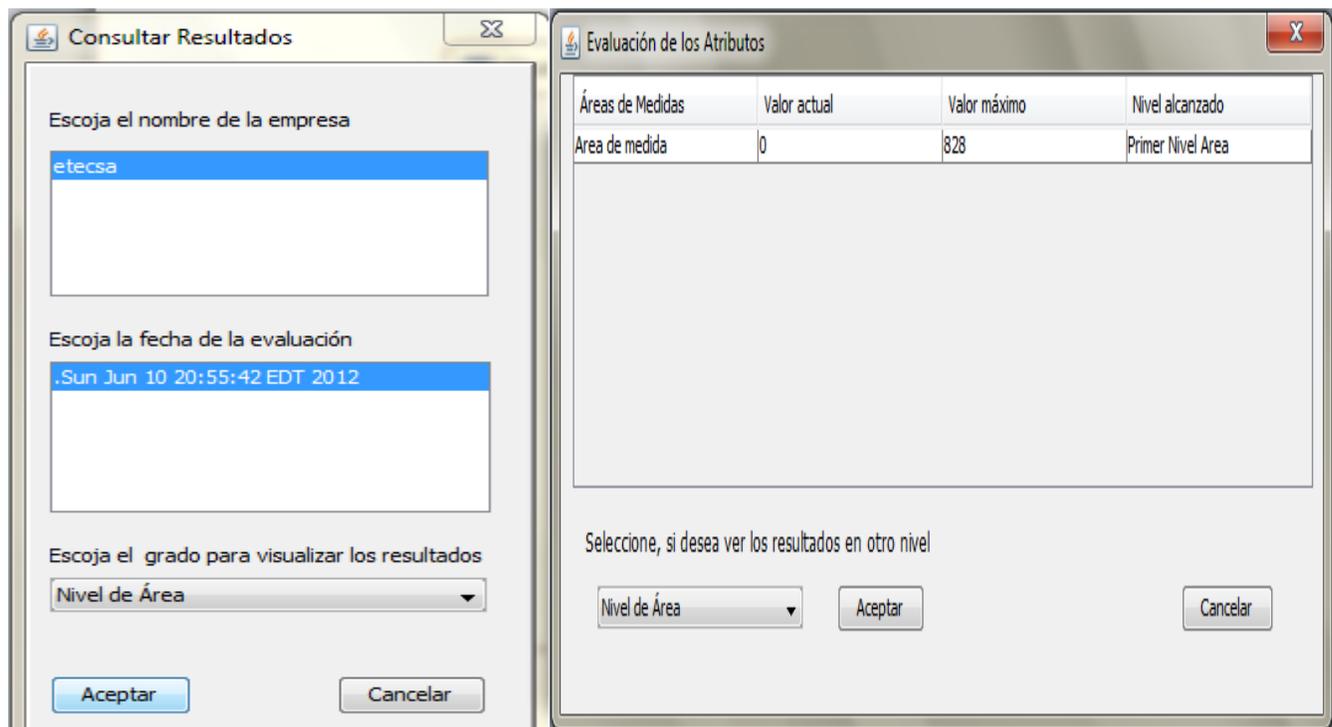


Figura 9 Prototipo de interfaz para la historia de usuario Consultar evaluación

Arquitectura del sistema (arquitectura dividida en tres capas)

Para la realización de la aplicación se hizo necesario crear una arquitectura dividida en tres capas, ya que sus características son fundamentales para el desarrollo y el orden que va a presentar el sistema. De manera que esta pueda contar con tres capas que nos permita un desarrollo organizativo de la aplicación. En la figura se muestra una vista de las capas de la arquitectura con que contará el sistema. La primera capa controla la presentación, en ella se controlan la interfaz de usuario y la navegación. La segunda capa es la encargada de proporcionar a la capa de presentación sus datos, así como de obtener los datos de esta para guardarlos en la capa inferior, en este caso, en la capa de almacenamiento de datos, y luego en dependencia, responder a los eventos presentados en la capa de presentación.

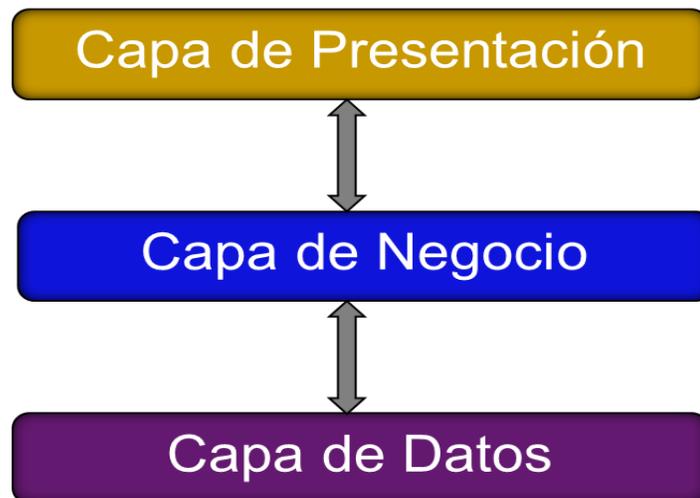


Figura 10 Arquitectura dividida en tres capas

Capa de Presentación: Es la que ve el usuario, también se la denomina capa de usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario, realiza un filtrado previo para comprobar que no hay errores de formato. También es conocida como interfaz gráfica y debe tener la característica de ser amigable para el usuario. Esta capa se comunica únicamente con la capa de negocio [29].

Capa Lógica de aplicación: Se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de lógica del negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y

presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él [29].

Capa de Datos: Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio [29].

Como fue anteriormente descrito en el capítulo anterior, para la conexión con la base de datos se utiliza JDBC. Este para comunicarse con la base de datos utiliza un driver intermedio, que no es más que una clase que implementa la interfaz Driver. Cuando se crea una instancia de una de estas clases Driver, esta se registra con el DriverManager (gestor de drivers) que es la encargada de decidir que driver se ha de utilizar para acceder a la base de datos. A partir de la clase DriverManager, a través del método estático getConnection(String url, String usuario, String password), se obtiene el objeto Connection que representa la conexión a la base de datos a través del Driver que el DriverManager ha seleccionado. Para hacer consultas a la base de datos se utiliza el método executeQuery(String consulta) del objeto Statement, que se obtiene a partir de Connection con createStatement(). Al ejecutar la consulta se obtiene un objeto ResultSet (conjunto resultado) con los campos de las tuplas que cumplen las condiciones de la consulta. Se mueve por cada uno de los resultados del ResultSet usando el método next, que devolverá false cuando no existan más resultados. Para acceder a los valores de los campos de la tupla actual se usan métodos de la forma getTipoDato(String nombreColumna).



Figura 11 Diseño estructural de la base de datos

Planificación

Durante la fase de planificación se realiza una estimación del esfuerzo que costará implementar cada historia de usuario. Este se expresa utilizado como medida el punto. Un punto se considera como una semana ideal de trabajo donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción [27].

Es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y, asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. La planificación no debe ser estricta puesto que hay muchas variables en juego, debe ser flexible para poder adaptarse a los cambios que

puedan surgir. Una buena estrategia es hacer planificaciones detalladas para unas pocas semanas y planificaciones mucho más abiertas para unos pocos meses.

Estimación del tiempo por las historias de usuarios

Para el desarrollo del sistema propuesto en esta investigación científica se realizó una estimación del esfuerzo para cada una de las historias de usuarios identificadas, llegándose a los resultados que se muestran en la siguiente tabla.

Historia de Usuario	Puntos de estimación
Gestionar Modelo	4
Autenticar	2
Gestionar Área de Medida	2
Gestionar Dimensión	2
Gestionar Categoría	2
Gestionar Atributo Cualitativo	2
Gestionar Pregunta	2
Realizar Evaluación	4
Consultar Evaluación	4
Gestionar Usuario	2
Gestionar Nivel	2
Gestionar Valor	1
Gestionar Peso	1

Tabla 6 Estimación de tiempo por historia de usuario

El tiempo total estimado para el desarrollo del sistema es de 30 puntos de estimación, que equivalen a 7 meses y 2 semana ideales de trabajo interrumpible.

Plan de Iteraciones

Una se vez identificadas las historia de usuario del sistema y estimado el esfuerzo dedicado a la realización de cada una de estas se procede a la planificación de la etapa de implementación del proyecto

[27]. En base a lo antes mencionado se decide realizar esta etapa en tres iteraciones, las cuales se detallan a continuación.

Iteración 1

Esta iteración tiene como objetivo la implementación de las historias de usuario de mayor prioridad. Durante el transcurso de la misma se creará la base de la arquitectura del sistema con una funcionalidad mínima haciendo mayor énfasis en la planificación y gestión de las diferentes áreas del modelo de madurez para SOA. Al final se contará con una primera versión de prueba, la cual será mostrada al cliente con el objetivo de obtener una retroalimentación para el grupo de trabajo.

Iteración 2

El objetivo de ésta es la implementación de las historias de usuario de prioridad media. Al finalizar se contará con una versión de prueba con las funcionalidades concernientes a la visualización del proceso de realizar una evaluación con todos los criterios que lleva éste, además de las funcionalidades implementadas en la iteración anterior. Esta será mostrada al cliente con el objetivo de realizar cambios necesarios en base a la opinión del mismo.

Iteración 3

Durante el transcurso de ésta, se implementaron las historias de usuario de baja prioridad. Al finalizar la misma se constará con la versión 1.0 del producto final, adicionando todo lo concerniente al manejo de estadísticas y visualización de los resultados obtenidos tanto los posteriores como el actual, además de las funcionalidades anteriores. Como resultado de ésta, el sistema será puesto en funcionamiento durante un período de tiempo para evaluar su desempeño.

Plan de duración de las iteraciones

Como parte del ciclo de vida de un proyecto utilizando XP se crea el plan de duración de cada una de las iteraciones, en este caso se hace para el único equipo de desarrollo con que se cuenta. Este plan se encarga de mostrar las historias de usuario que serán abordadas en cada una de las iteraciones, así como la duración estimada de estas últimas y el orden en que se implementaran las historias de usuario.

Iteración	Orden de las historias de usuario a implementar	Duración de la iteración
Iteración 1	1. Gestionar Modelo	16 semanas
	2. Gestionar Área de Medida	
	3. Gestionar Dimensión	
	4. Gestionar Categoría	
	5. Gestionar Atributo Cualitativo	
	6. Gestionar Pregunta	
	7. Gestionar Valor	
	8. Gestionar Peso	
Iteración 2	1. Realizar Evaluación	8 semanas
	2. Consultar Evaluación	
Iteración 3	1. Gestionar Usuario	6 semanas
	2. Autenticar	
	3. Gestionar Nivel	

Tabla 7 Plan de duración de las iteraciones

Plan de Entregas

A continuación se presenta el plan de entregas ideado para la fase de implementación. Como resultado del mismo se harán versiones de la aplicación de escritorio al finalizar cada iteración en la fecha aproximada que se indica en la siguiente tabla.

Módulo	Fin 1ra Iteración 3ra semana de Febrero	Fin 2da Iteración 3ra semana de Abril	Fin 3ra Iteración 1ra semana de Junio
Sistema Propuesto	0.1	0.2	1.0

Tabla 8 Plan de entrega

Diseño del Sistema

El diseño adecuado para el software es aquel que supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación del equipo de desarrollo y tiene el menor número posible de clases y métodos. Para el diseño de las aplicaciones, la metodología XP no requiere la representación del sistema mediante diagramas de clases utilizando notación UML. En su lugar se usan otras técnicas como las tarjetas CRC (Clase, Responsabilidad, Colaborador) como una extensión informal a UML. La técnica de las tarjetas CRC se puede usar para guiar el sistema a través de análisis guiados por la responsabilidad. Las clases se examinan, se filtran y se refinan en base a sus responsabilidades con respecto al sistema, y las clases con las que necesitan colaborar para completar sus responsabilidades.

Tarjeta CRC

Las tarjetas CRC son una herramienta de reflexión en el diseño de software orientado a objetos. Fueron propuestas por Ward Cunningham y Kent Beck. Se utilizan normalmente cuando primero se determinan las clases que se necesitan y cómo van a interactuar y después se implementa la solución.[30]

A continuación se muestra algunas de las tarjetas CRC que fueron confeccionadas:

Tarjeta CRC	
Clase: Principal	
Súper clase:	
Subclase(s):	
Responsabilidades	Colaboraciones
✓ Gestionar Usuario	✓ Usuario
✓ Gestionar Modelo	✓ Área Medida
✓ Gestionar Área de Medida	✓ Dimensión
✓ Gestionar Dimensión	✓ Categoría
✓ Gestionar Categoría	✓ Atributo
✓ Gestionar Atributo cualitativo	✓ Pregunta
✓ Gestionar Pregunta	✓ Respuesta
✓ Gestionar Nivel	✓ Consulta
✓ Gestionar Valor	✓ Organización
✓ Gestionar Peso	✓ Evaluación

<ul style="list-style-type: none"> ✓ Realizar Evaluación ✓ Consultar Resultados ✓ Comparar Evaluaciones ✓ Mostrar estado del modelo ✓ Mostrar listado de administradores ✓ Mostrar información de un nodo 	
---	--

Tabla 9 Tarjeta CRC de la clase Principal

Tarjeta CRC	
Clase: Consulta Súper clase: Subclase(s):	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> ✓ Insertar Usuario ✓ Insertar Área de Medida ✓ Insertar Dimensión ✓ Insertar Categoría ✓ Insertar Atributo cualitativo ✓ Insertar Pregunta ✓ Insertar Respuestas ✓ Insertar Organización ✓ Insertar Evaluación ✓ Insertar Nivel ✓ Insertar Valor ✓ Insertar Peso ✓ Modificar Usuario ✓ Modificar Área de Medida ✓ Modificar Dimensión ✓ Modificar Categoría ✓ Modificar Atributo cualitativo 	<ul style="list-style-type: none"> ✓ Conexión ✓ Statement ✓ ResultSet

<ul style="list-style-type: none"> ✓ Modificar Pregunta ✓ Modificar Nivel ✓ Modificar Valor ✓ Modificar Peso ✓ Eliminar Usuario ✓ Eliminar Área de Medida ✓ Eliminar Dimensión ✓ Eliminar Categoría ✓ Eliminar Atributo cualitativo ✓ Eliminar Pregunta ✓ Eliminar Respuestas ✓ Eliminar Organización ✓ Eliminar Evaluación ✓ Eliminar Nivel ✓ Eliminar Valor ✓ Eliminar Peso ✓ Seleccionar descripción de un nodo ✓ Seleccionar nombre de un nodo ✓ Seleccionar listado de administradores 	
--	--

Tabla 10 Tarjeta CRC de la clase Consulta

Tarjeta CRC	
Clase: Conexión	
Súper clase:	
Subclase(s):	
Responsabilidades	Colaboraciones
✓ GetConnection	✓ DriverManager

Tabla 11 Tarjeta CRC de la clase Conexión

Patrones de diseño

Un patrón de diseño es una solución repetible a un problema recurrente en el diseño de software. Esta solución no es un diseño terminado que puede traducirse directamente a código, sino más bien una descripción sobre cómo resolver el problema, la cual puede ser utilizada en diversas situaciones. Los patrones de diseño reflejan todo el rediseño y re modificación que los desarrolladores han ido haciendo a medida que intentaban conseguir mayor reutilización y flexibilidad en su software.

Los patrones documentan y explican problemas de diseño, y luego discuten una buena solución a dicho problema. Con el tiempo, los patrones comienzan a incorporarse al conocimiento y experiencia colectiva de la industria del software, lo que demuestra que el origen de los mismos radica en la práctica misma más que en la teoría[31].

Patrones GRASP (Patrones de Software para la asignación General de Responsabilidades)

Estos patrones constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objetos. Los patrones GRASP se dividen en 6 ellos son: experto, creador, controlador, fachada, alta cohesión y bajo acoplamiento[32].

Patrones GRASP utilizados:

- **Creador:** El patrón creador ayuda a identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases. Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creador. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia, es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización.

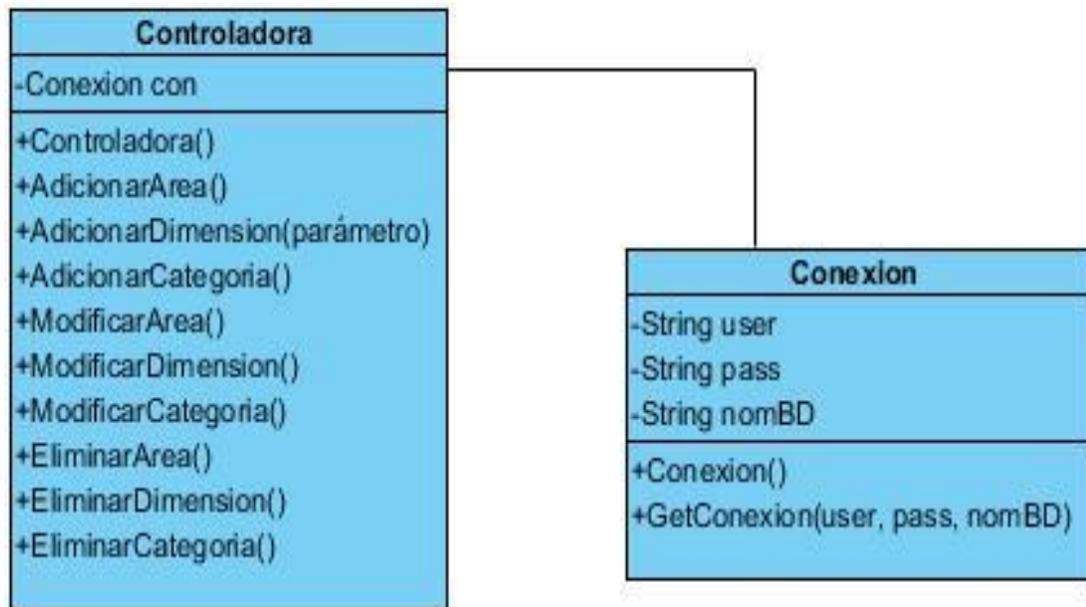


Figura 12 Diagrama de clase en donde se evidencia la utilización del patrón Creador⁶

- **Controlador:** Es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto es para aumentar la reutilización de código y a la vez tener un mayor control. Recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

⁶ Las clases expuesta en la imagen no cuentan con todos los métodos y atributos definidos, es solo una pequeña representación de las clases para validar la utilización del patrón Creador.

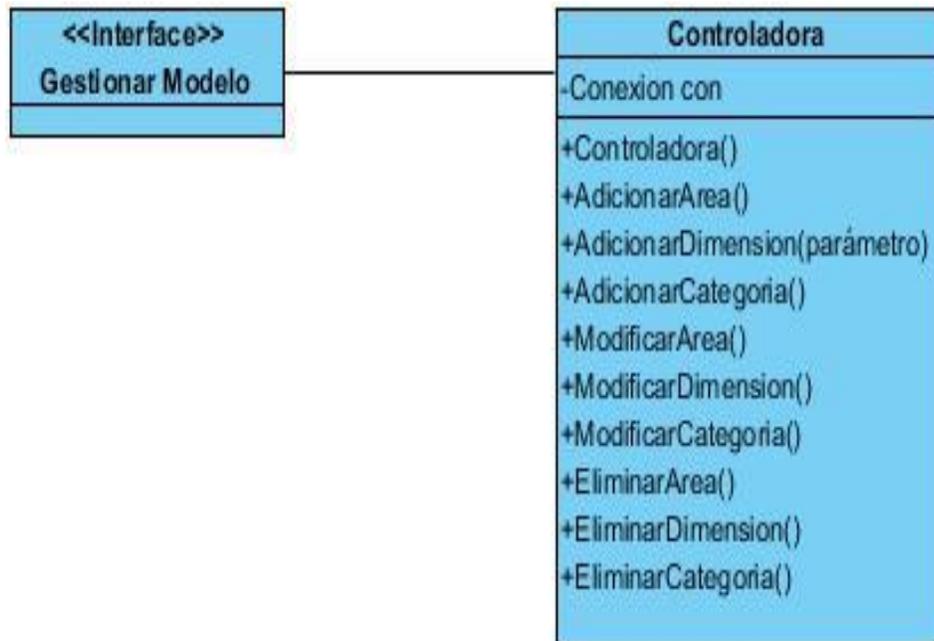


Figura 13 Diagrama de clase en donde se evidencia la utilización del patrón Controlador⁷

- **Bajo acoplamiento:** Es el patrón que mantiene las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.
- **Experto:** La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase se considera experta si contiene toda la información necesaria para realizar la labor que tiene encomendada.

⁷ La clase Controladora expuesta en la imagen no cuenta con todos los métodos y atributos definidos, es solo una pequeña representación de la clase para validar la utilización del patrón Controlador.

Conclusiones parciales del capítulo

Durante la elaboración de este capítulo se hace referencia a las fases de Exploración y fase de Planificación propias de la metodología de desarrollo utilizada para la implementación de la aplicación de escritorio. En este capítulo se determinaron las funcionalidades que debe tener el sistema mostrados en historias de usuarios. También se hace una descripción de todos los artefactos generados mediante las historias de usuarios, el plan de iteraciones y el plan de entrega, además se definió la arquitectura a emplear para el desarrollo del software. Se asume una implementación por etapas la cual fue detallada, asegurando su correcta elaboración. Además, se especificó el diseño de clases del sistema mediante la utilización de las tarjetas CRC, así como se definieron los patrones de diseño a utilizar.

Capítulo#3 Implementación y prueba

Introducción

En el presente capítulo se detallan las tres iteraciones llevadas a cabo durante la etapa de codificación del sistema, ya que XP plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo en cada iteración un producto funcional el cual va a ser evaluado por el cliente, para luego su continuo desarrollo, cumpliendo con las expectativas recomendadas por el mismo. Además, se definen las pruebas a realizarle al sistema para comprobar su funcionamiento.

Iteraciones en el desarrollo del sistema

En esta fase se genera todo el código fuente necesario para satisfacer las historias de usuario definidas para la solución y se describen todas las tareas realizadas en cada iteración. Al inicio de cada Historia de Usuario, se lleva a cabo una revisión del plan de iteraciones y se modifica de ser necesario. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable. Estas tareas, pueden escribirse utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente.

Una vez realizada la planificación, se determinaron tres iteraciones de desarrollo sobre el sistema, obteniendo como resultado un producto con todas las restricciones y características deseadas por el cliente para posteriormente ser utilizado. A continuación se hace alusión a cada una de las iteraciones.

Iteración 1

No HU	Historias de Usuarios	Tiempo de implementación	
		Estimación	Real
1	Gestionar Modelo	4	4
2	Gestionar Área de Medida	2	2
3	Gestionar Dimensión	2	2
4	Gestionar Categoría	2	2
5	Gestionar Atributo Cualitativo	2	2
6	Gestionar Pregunta	2	2
7	Gestionar Valor	1	1
8	Gestionar Peso	1	1

Tabla 12 Historias de usuarios abordadas en la primera iteración

Tareas de las historias de usuarios pertenecientes a la primera iteración

Gestionar Modelo

Tarea #1

Tarea por Historia de Usuario	
Número de tarea: 1	Número de historia de usuario: 1
Nombre de la tarea: Adicionar elemento a la estructura del modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 20 de octubre del 2011	Fecha fin: 27 de octubre del 2011
Descripción: El administrador adicionará un nuevo elemento a la estructura del modelo, especificando en el árbol el padre del nuevo elemento a adicionar.	

Tabla 13 Primera tarea asociada a la historia de usuario Gestionar Modelo

Tarea #2

Tarea por Historia de Usuario	
Número de tarea: 2	Número de historia de usuario: 1
Nombre de la tarea: Modificar elemento en la estructura del modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 27 de octubre del 2011	Fecha fin: 3 de noviembre del 2011
Descripción: El administrador podrá modificar algún elemento asociado al modelo, para realizar dicha acción primeramente deberá de escoger el elemento en el árbol.	

Tabla 14 Segunda tarea asociada a la historia de usuario Gestionar Modelo

Tarea #3

Tarea por Historia de Usuario	
Número de tarea: 3	Número de historia de usuario: 1
Nombre de la tarea: Eliminar elemento en la estructura del modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 3 de noviembre del 2011	Fecha fin: 17 de noviembre del 2011
Descripción: El administrador podrá eliminar algún elemento asociado al modelo, para realizar dicha acción primeramente deberá de escoger el elemento en el árbol, una vez realizada esta acción los nodos hijos del nodo seleccionado también serán eliminados.	

Tabla 15 Tercera tarea asociada a la historia de usuario Gestionar Modelo

Gestionar Área de Medida

Tarea #1

Tarea por Historia de Usuario	
Número de tarea: 1	Número de historia de usuario: 2
Nombre de la tarea: Adicionar área de medida al modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 17 de noviembre del 2011	Fecha fin: 24 de noviembre del 2011
Descripción: Se verifica la información referente al área de medida a adicionar, una vez comprobado los datos de almacena la nueva área en la BD.	

Tabla 16 Primera tarea asociada a la historia de usuario Gestionar Área de Medida

Tarea #2

Tarea por Historia de Usuario	
Número de tarea: 2	Número de historia de usuario: 2
Nombre de la tarea: Eliminar área de medida al modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 24 de noviembre del 2011	Fecha fin: 27 de noviembre del 2011
Descripción: El administrador selecciona el área de medida que desee eliminar del modelo de madurez y posteriormente se elimina de la BD.	

Tabla 17 Segunda tarea asociada a la historia de usuario Gestionar Área de Medida

Tarea #3

Tarea por Historia de Usuario	
Número de tarea: 3	Número de historia de usuario: 2
Nombre de la tarea: Modificar área de medida al modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 27 de noviembre del 2011	Fecha fin: 30 de noviembre del 2011
Descripción: El administrador selecciona el área de medida que desee modificar y el sistema le muestra la información referente a su selección para que se lleve a cabo la modificación de los datos.	

Tabla 18 Tercera tarea asociada a la historia de usuario Gestionar Área de Medida

Gestionar Dimensión

Tarea #1

Tarea por Historia de Usuario	
Número de tarea: 1	Número de historia de usuario: 3
Nombre de la tarea: Adicionar dimensión al modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 1 de diciembre del 2011	Fecha fin: 8 de diciembre del 2011
Descripción: El usuario selecciona el área de medida a la cual le añadirá la nueva dimensión, se verifica la información referente a la dimensión, una vez comprobado los datos de almacena la nueva dimensión en la BD.	

Tabla 19 Primera tarea asociada a la historia de usuario Gestionar Dimensión

Tarea #2

Tarea por Historia de Usuario	
Número de tarea: 2	Número de historia de usuario: 3
Nombre de la tarea: Eliminar dimensión del modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 8 de diciembre del 2011	Fecha fin: 11 de diciembre del 2011
Descripción: El administrador selecciona la dimensión que desea eliminar del modelo de madurez y posteriormente se elimina de la BD.	

Tabla 20 Segunda tarea asociada a la historia de usuario Gestionar Dimensión

Tarea #3

Tarea por Historia de Usuario	
Número de tarea: 3	Número de historia de usuario: 3
Nombre de la tarea: Modificar dimensión al modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 11 de diciembre del 2011	Fecha fin: 14 de diciembre del 2011
Descripción: El administrador selecciona la dimensión que desee modificar y el sistema le muestra la información referente a su selección para que se lleve a cabo la modificación de los datos.	

Tabla 21 Tercera tarea asociada a la historia de usuario Gestionar Dimensión

Gestionar Categoría

Tarea #1

Tarea por Historia de Usuario	
Número de tarea: 1	Número de historia de usuario: 4
Nombre de la tarea: Adicionar categoría al modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 14 de diciembre del 2011	Fecha fin: 21 de diciembre del 2011
Descripción: El usuario selecciona la dimensión a la cual le añadirá la nueva categoría, se verifica la información referente a la categoría, una vez comprobado los datos de almacena la nueva dimensión en la BD.	

Tabla 22 Primera tarea asociada a la historia de usuario Gestionar Categoría

Tarea #2.

Tarea por Historia de Usuario	
Número de tarea: 2	Número de historia de usuario: 4
Nombre de la tarea: Eliminar categoría del modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 5 de enero del 2012	Fecha fin: 8 de enero del 2012
Descripción: El administrador selecciona la categoría que desea eliminar del modelo de madurez y posteriormente se elimina de la BD.	

Tabla 23 Segunda tarea asociada a la historia de usuario Gestionar Categoría

Tarea #3

Tarea por Historia de Usuario	
Número de tarea: 3	Número de historia de usuario: 4
Nombre de la tarea: Modificar categoría al modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 8 de enero del 2012	Fecha fin: 11 de enero del 2012
Descripción: El administrador selecciona la categoría que desee modificar y el sistema le muestra la información referente a su selección para que se lleve a cabo la modificación de los datos.	

Tabla 24 Tercera tarea asociada a la historia de usuario Gestionar Categoría

Gestionar Atributo Cualitativo

Tarea #1

Tarea por Historia de Usuario	
Número de tarea: 1	Número de historia de usuario: 5
Nombre de la tarea: Adicionar atributo cualitativo al modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 11 de enero del 2012	Fecha fin: 18 de enero del 2012
Descripción: El usuario selecciona la categoría a la cual le añadirá el nuevo atributo cualitativo, se verifica la información referente al atributo cualitativo, una vez comprobado los datos de almacena la nueva dimensión en la BD.	

Tabla 25 Tabla 24 Primera tarea asociada a la historia de usuario Gestionar Atributo Cualitativo

Tarea #2

Tarea por Historia de Usuario	
Número de tarea: 2	Número de historia de usuario: 5
Nombre de la tarea: Eliminar atributo cualitativo del modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 18 de enero del 2012	Fecha fin: 21 de enero del 2012
Descripción: El administrador selecciona el atributo cualitativo que desea eliminar del modelo de madurez y posteriormente se elimina de la BD.	

Tabla 26 Segunda tarea asociada a la historia de usuario Gestionar Atributo Cualitativo

Tarea #3

Tarea por Historia de Usuario	
Número de tarea: 3	Número de historia de usuario: 5
Nombre de la tarea: Modificar atributo cualitativo al modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 21 de enero del 2012	Fecha fin: 24 de enero del 2012
Descripción: El administrador selecciona el atributo cualitativo que desee modificar y el sistema le muestra la información referente a su selección para que se lleve a cabo la modificación de los datos.	

Tabla 27 Tercera tarea asociada a la historia de usuario Gestionar Atributo Cualitativo

Gestionar Pregunta

Tarea #1

Tarea por Historia de Usuario	
Número de tarea: 1	Número de historia de usuario: 6
Nombre de la tarea: Adicionar pregunta al modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 24 de enero del 2012	Fecha fin: 31 de enero del 2012
Descripción: El usuario selecciona el atributo cualitativo al cual le añadirá la nueva pregunta, se verifica la información referente a la pregunta, una vez comprobado los datos de almacena la nueva dimensión en la BD.	

Tabla 28 Primera tarea asociada a la historia de usuario Gestionar Pregunta

Tarea #2

Tarea por Historia de Usuario	
Número de tarea: 2	Número de historia de usuario: 6
Nombre de la tarea: Eliminar pregunta del modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 1 de febrero del 2012	Fecha fin: 4 de febrero del 2012
Descripción: El administrador selecciona el pregunta que desea eliminar del modelo de madurez y posteriormente se elimina de la BD.	

Tabla 29 Segunda tarea asociada a la historia de usuario Gestionar Pregunta

Tarea #3

Tarea por Historia de Usuario	
Número de tarea: 3	Número de historia de usuario: 6
Nombre de la tarea: Modificar pregunta del modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 4 de febrero del 2012	Fecha fin: 7 de febrero del 2012
Descripción: El administrador selecciona la pregunta que desee modificar y el sistema le muestra la información referente a su selección para que se lleve a cabo la modificación de los datos.	

Tabla 30 Tercera tarea asociada a la historia de usuario Gestionar Pregunta

Gestionar Valor

Tarea #1

Tarea por Historia de Usuario	
Número de tarea: 1	Número de historia de usuario: 7
Nombre de la tarea: Adicionar un nuevo valor	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 7 de febrero del 2012	Fecha fin: 10 de febrero del 2012
Descripción: El usuario introduce el valor, se verifica la información y se almacena el nuevo valor.	

Tabla 31 Primera tarea asociada a la historia de usuario Gestionar Valor

Tarea #2

Tarea por Historia de Usuario	
Número de tarea: 2	Número de historia de usuario: 7
Nombre de la tarea: Eliminar un valor	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 10 de febrero del 2012	Fecha fin: 13 de febrero del 2012
Descripción: El administrador selecciona el valor que desea eliminar y posteriormente se elimina de la BD.	

Tabla 32 Segunda tarea asociada a la historia de usuario Gestionar Valor

Gestionar Peso

Tarea #1

Tarea por Historia de Usuario	
Número de tarea: 1	Número de historia de usuario: 8
Nombre de la tarea: Adicionar un nuevo peso	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 13 de febrero del 2012	Fecha fin: 16 de febrero del 2012
Descripción: El usuario introduce el peso, se verifica la información y se almacena el nuevo peso.	

Tabla 33 Primera tarea asociada a la historia de usuario Gestionar Peso

Tarea #2

Tarea por Historia de Usuario	
Número de tarea: 2	Número de historia de usuario: 8
Nombre de la tarea: Eliminar un peso	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 16 de febrero del 2012	Fecha fin: 19 de febrero del 2012
Descripción: El administrador selecciona el peso que desea eliminar y posteriormente se elimina de la BD.	

Tabla 34 Segunda tarea asociada a la historia de usuario Gestionar Peso

Iteración 2

No HU	Historias de Usuarios	Tiempo de implementación	
		Estimación	Real
9	Realizar Evaluación	4	4
10	Consultar Evaluación	4	4

Tabla 35 Historias de Usuarios abordadas en la segunda iteración

Tareas de las Historias Usuarios pertenecientes a la segunda iteración

Realizar Evaluación

Tarea #1

Tarea por Historia de Usuario	
Número de tarea: 1	Número de historia de usuario: 9
Nombre de la tarea: Autenticar la empresa	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 19 de febrero del 2012	Fecha fin: 26 de febrero del 2012
Descripción: El evaluador escribe el nombre de la empresa a la cual le aplicará la evaluación, además especifica la fecha de la misma y se almacena en la BD.	

Tabla 36 Primera tarea asociada a la historia de usuario Realizar Evaluación

Tarea #2

Tarea por Historia de Usuario	
Número de tarea: 2	Número de historia de usuario: 9
Nombre de la tarea: Mostrar preguntas a responder	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 1 de marzo del 2012	Fecha fin: 15 de marzo del 2012
Descripción: El sistema muestra las preguntas a responder separadas de acuerdo al atributo cualitativo al que pertenezcan.	

Tabla 37 Segunda tarea asociada a la historia de usuario Realizar Evaluación

Tarea #3

Tarea por Historia de Usuario	
Número de tarea: 3	Número de historia de usuario: 9
Nombre de la tarea: Responder preguntas	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 16 de marzo del 2012	Fecha fin: 22 de marzo del 2012
Descripción: El evaluador responde las preguntas mostradas por el sistema, el cual almacenará las respuestas dadas por el evaluador en la BD.	

Tabla 38 Tercera tarea asociada a la historia de usuario Realizar Evaluación

Consultar Evaluación

Tarea #1

Tarea por Historia de Usuario	
Número de tarea: 1	Número de historia de usuario: 10
Nombre de la tarea: Autenticar la empresa	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 23 de marzo del 2012	Fecha fin: 30 de marzo del 2012
Descripción: El evaluador consulta la evaluación arrojada por el sistema al haber terminado el algoritmo para determinar el nivel en que se encuentra la empresa teniendo en cuenta las respuestas dadas a las preguntas.	

Tabla 39 Primera tarea asociada a la historia de usuario Consultar Evaluación

Iteración 3

No HU	Historias de Usuarios	Tiempo de implementación	
		Estimación	Real
11	Gestionar Usuarios	2	2
12	Autenticar Usuario	2	2
13	Gestionar Nivel	2	2

Tabla 40 Historias de Usuarios abordadas en la tercera iteración

Tareas de las Historias Usuarios pertenecientes a la tercera iteración

Gestionar Usuarios

Tarea #1

Tarea por Historia de Usuario	
Número de tarea: 1	Número de historia de usuario: 11
Nombre de la tarea: Adicionar un nuevo usuario al sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 1 de abril del 2012	Fecha fin: 8 de abril del 2012
Descripción: El administrador introduce los datos del nuevo usuario del sistema y esta información es almacenada en la BD.	

Tabla 41 Primera tarea asociada a la historia de usuario Gestionar Usuario

Tarea #2

Tarea por Historia de Usuario	
Número de tarea: 2	Número de historia de usuario: 11
Nombre de la tarea: Eliminar usuario del sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 9 de abril del 2012	Fecha fin: 12 de abril del 2012
Descripción: El administrador selecciona el usuario que desea eliminar del sistema y posteriormente se elimina de la BD.	

Tabla 42 Segunda tarea asociada a la historia de usuario Gestionar Usuario

Tarea #3

Tarea por Historia de Usuario	
Número de tarea: 3	Número de historia de usuario: 11
Nombre de la tarea: Modificar usuario del sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 13 de abril del 2012	Fecha fin: 17 de abril del 2012
Descripción: El administrador selecciona el usuario que desea modificar y el sistema le muestra su información almacenada para que se lleve a cabo la modificación de los datos.	

Tabla 43 Tercera tarea asociada a la historia de usuario Gestionar Usuario

Autenticar Usuario

Tarea #1

Tarea por Historia de Usuario	
Número de tarea: 1	Número de historia de usuario: 12
Nombre de la tarea: Introducir usuario y contraseña	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 10 de mayo del 2012	Fecha fin: 17 de mayo del 2012
Descripción: La persona introduce su usuario y contraseña en el sistema.	

Tabla 44 Primera tarea asociada a la historia de usuario Autenticar Usuario

Tarea #2

Tarea por Historia de Usuario	
Número de tarea: 2	Número de historia de usuario: 12
Nombre de la tarea: Verificar usuario y contraseña	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 17 de mayo del 2012	Fecha fin: 24 de mayo del 2012
Descripción: Se verifica que el usuario y la contraseña sean válidos, una vez comprobado estos se otorgan los permisos adecuados acuerdo de al rol que juegue en la BD.	

Tabla 45 Segunda tarea asociada a la historia de usuario Autenticar Usuario

Gestionar Nivel

Tarea #1

Tarea por Historia de Usuario	
Número de tarea: 1	Número de historia de usuario: 13
Nombre de la tarea: Adicionar un nuevo nivel al modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 24 de mayo del 2012	Fecha fin: 31 de mayo del 2012
Descripción: El administrador introduce los datos del nuevo nivel del modelo y esta información es almacenada en la BD.	

Tabla 46 Primera tarea asociada a la historia de usuario Gestionar Nivel

Tarea #2

Tarea por Historia de Usuario	
Número de tarea: 2	Número de historia de usuario: 13
Nombre de la tarea: Eliminar nivel del modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 1 de junio del 2012	Fecha fin: 8 de junio del 2012
Descripción: El administrador selecciona el nivel que desea eliminar del modelo y posteriormente se elimina de la BD.	

Tabla 47 Segunda tarea asociada a la historia de usuario Gestionar Nivel

Pruebas

En la fase de producción se requieren de pruebas adicionales y revisiones de rendimiento antes de que el sistema al haber concluido las iteraciones definidas sea trasladado al entorno del cliente.

Uno de los pilares fundamentales de XP es el proceso de pruebas, el cual anima a los desarrolladores a probar constantemente tanto como sea posible. Mediante esta filosofía se reduce el número de errores no detectados así como el tiempo entre la introducción de este en el sistema y su detección. Todo esto contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones. La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma

automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente.

Pruebas unitarias

En el transcurso de implementación de cada historia de usuario perteneciente a una determinada iteración, se realizaron al código generado una serie de pruebas para determinar su correcto funcionamiento.

Requisitos para la correcta realización de una prueba unitaria:

- ✓ **Automatizable:** no debería requerirse una intervención manual.
- ✓ **Completas:** deben cubrir la mayor cantidad de código.
- ✓ **Repetibles o Reutilizables:** no se deben crear pruebas que sólo puedan ser ejecutadas una sola vez.
- ✓ **Independientes:** la ejecución de una prueba no debe afectar a la ejecución de otra.
- ✓ **Profesionales:** las pruebas deben ser consideradas igual que el código, con la misma profesionalidad, documentación.

La realización de las pruebas unitarias brindó unas series de ventajas al proceso de implementación del sistema, entre estas ventajas se encuentran:

- **Los errores están más acotados y son más fáciles de localizar:** dado que tenemos pruebas unitarias que pueden desenmascararlos.
- **Documenta el código:** Las propias pruebas son documentación del código puesto que ahí se puede ver cómo utilizarlo.
- **Fomentan el cambio:** Las pruebas unitarias facilitan que el programador cambie el código para mejorar su estructura. Además, permiten hacer pruebas sobre los cambios y así asegurarse de que los nuevos cambios no han introducido errores.

Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las Historias de Usuario. Durante las iteraciones las historias de usuario seleccionadas serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una HU ha sido implementada correctamente. Una historia de usuario puede tener todas las pruebas de aceptación que

necesite para asegurar su correcto funcionamiento. El objetivo final de estas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable.

Las pruebas de aceptación tienen más peso que las unitarias ya que constituyen un indicador de la satisfacción del cliente con la solución además de marcar el final de una iteración y el comienzo de la siguiente. Se recomienda que el cliente sea quien diseñe estas pruebas o que al menos participe de manera activa en el proceso.

Como resultado de las pruebas de aceptación se obtendrán artefactos descritos en tablas, estas contarán con los siguientes campos:

- ✓ Código: servirá como identificador de la prueba realizada, a su vez será sugerente al nombre de la prueba a la que hace referencia.
- ✓ UH: tendrá el nombre de la historia de usuario a la que hace referencia la prueba a realizar.
- ✓ Nombre: nombre que se le da a la prueba a realizar.
- ✓ Descripción: se describe la funcionalidad que se desea probar.
- ✓ Condiciones de Ejecución: mostrará las condiciones que deben cumplirse para poder llevar a cabo el caso de prueba, estas condiciones deben ser satisfechas antes de la ejecución del caso de prueba para que se puedan obtener los resultados esperados.
- ✓ Entradas / Pasos de Ejecución: se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las entradas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.
- ✓ Resultado esperado: se hará una breve descripción del resultado que se espera obtener con la prueba realizada.
- ✓ Evaluación de la prueba: acorde al resultado de la prueba realizada se emitirá una evaluación sobre la misma. Esta evaluación tendrá uno de los tres resultados que a continuación se describen:
 - 1) Bien: cuando el resultado de la prueba es exactamente el esperado por el usuario.
 - 2) Parcialmente bien: cuando el resultado no es completamente el esperado por el cliente o usuario de la aplicación y muestra resultados erróneos o fuera de contexto.
 - 3) Mal: cuando el resultado de la prueba realizada genera un error de codificación en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, trayendo

como consecuencia que la funcionalidad requerida por el cliente no tenga resultado, lo que invalida también la UH.

A continuación se muestran las pruebas de aceptación realizadas para las historias de usuario que fueron mostradas en el capítulo anterior, si se desea consultar las restantes pruebas de aceptación dirigirse al (ANEXO #3).

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Gestionar modelo.	
Descripción: Prueba para la funcionalidad de ir conformando la estructura del modelo de madurez.	
Condiciones de Ejecución: Se debe mostrar el nodo raíz del modelo de madurez, en la implementación del sistema se le llamó a este nodo “Modelo de Madurez”, por lo que inicialmente debería mostrarse solamente este nodo, y a partir del mismo se procede a ir conformando la estructura del árbol.	
Entradas/Pasos de Ejecución: Se intenta adicionar, eliminar o modificar algún elemento sin antes determinar su nodo padre. Se intenta adicionar, eliminar o modificar algún elemento determinando su nodo padre.	
Resultado Esperado: El sistema rechaza la acción en donde no se especifica el nodo padre y acepta la acción en donde sí se especifica el nodo padre.	
Evaluación de la Prueba: Bien.	

Tabla 48 Prueba de aceptación sobre la historia de usuario Gestionar Modelo

Caso de Prueba de Aceptación	
Código: HU11_P1	Historia de Usuario: 11
Nombre: Realizar evaluación.	
Descripción: Prueba para la funcionalidad de autenticar usuario.	
Condiciones de Ejecución: Debe de estar insertado en la BD al menos un usuario. Se utilizará un usuario existente con datos válidos y un usuario con datos incorrectos.	
Entradas/Pasos de Ejecución: Se intenta gestionar un usuario con datos válidos y otro usuario con datos inválidos o vacíos.	
Resultado Esperado: El sistema permite gestionar el usuario con los datos válidos y muestra un mensaje de alerta al intentar gestionar el usuario con datos inválidos.	
Evaluación de la Prueba: Bien.	

Tabla 49 Prueba de aceptación sobre la historia de usuario Realizar Evaluación

Caso de Prueba de Aceptación	
Código: HU9_P1	Historia de Usuario: 10
Nombre: Consultar evaluación.	
Descripción: Prueba para la funcionalidad de consultar evaluación.	
Condiciones de Ejecución: Debe de estar registrada al menos una evaluación. Se utilizará una empresa que no posea ninguna evaluación realizada y otra que si posea al menos una evaluación.	
Entradas/Pasos de Ejecución: Se intenta acceder a la interfaz que muestre los resultados con el nombre de la empresa con ninguna evaluación realizada, y posteriormente con el nombre de la empresa que si posee una evaluación.	
Resultado Esperado: El sistema muestra los resultados de la empresa con evaluación y no procede con la empresa que no cuenta con ninguna evaluación, mostrando un mensaje de alerta.	
Evaluación de la Prueba: Bien.	

Tabla 50 Prueba de aceptación sobre la historia de usuario Consultar Evaluación

Conclusiones parciales del capítulo

En el presente se definieron las tareas a desarrollar pertenecientes a cada historia de usuario en cada una de las tres iteraciones anteriormente planificadas. Estas tareas fueron desarrolladas de acorde al tiempo anteriormente planificado para las mismas, posibilitando de esta manera que se cumpliera el plan de entrega del producto, descrito en el capítulo anterior. Por último se realizaron pruebas al sistema para comprobar su funcionalidad. Llevándose a cabo las pruebas de aceptación, la cual fue seleccionada ya que la misma es llevada a cabo por el cliente y es el mejor evaluador que puede tener el sistema, ya que el mismo, puede verificar que el sistema cumpla con sus expectativas.

Conclusiones generales del trabajo realizado

Como resultado del presente trabajo, se realizó un estudio de las herramientas y tecnologías que más se adecuaban a las necesidades en el desarrollo de la solución propuesta. Se definieron los requisitos que se deberían tener en cuenta para permitir un correcto desarrollo en la confección y aplicación del modelo de madurez para SOA definido por el CDAE. Para un correcto trabajo en el cumplimiento de los requisitos definidos, se estableció la arquitectura que debería tener la aplicación a desarrollar, así como los patrones de diseño que permitieran obtener ventajas significativas en su utilización. El proceso de implementación del sistema, para una mejor planificación, fue realizado a través de tres iteraciones, en cada iteración se implementaron unas series de requisitos de acuerdo a su relevancia en el negocio, estableciéndose unas series de tareas a realizar en la implementación de cada uno de los requisitos.

Al concluir todas las acciones anteriormente mencionadas, se obtuvo un sistema informático que satisface las necesidades del CDAE, y el cual se le realizó unas series de pruebas para comprobar su correcto funcionamiento.

Referencias bibliográficas

1. Sessions, R. "A Comparison of the Top Four Enterprise Architecture Methodologies," 20 de febrero 2008 [cited; Available from: www.objectwatch.com].
2. National_Institute_of_Health. *Enterprise Architecture*. 2008 [cited; Available from: <http://enterprisearchitecture.nih.gov/About/What/>].
3. Schekkerman, J. *What you have to know about Services Orientation*. 2006 [cited; Available from: <http://www.enterprise-architecture.info>].
4. Ibrahim, M.a.L. *Service-Oriented Architecture and Enterprise Architecture, Part 1: A framework for understanding how SOA and Enterprise Architecture work together*. 2007 [cited; Available from: <http://www.ibm.com/developerWorks/>].
5. Cámara, J., *Recomendaciones para la adopción de SOA*.
6. CRS_It_Consulting. [cited; Available from: http://www.crs-itconsulting.com/index.php?option=com_content&view=article&id=1217].
7. Sebor, J. *7 Steps to SOA Success*
2008 [cited; Available from: <http://www.destinationcrm.com/>].
8. McKendrick, J. *How do we really know when SOA 'fails'?* . 2007 [cited; Available from: <http://www.zdnet.com/>].
9. McKendrick, J. *Explaining SOA failure to the boss: here are some real good excuses*. 2008 [cited; Available from: <http://www.zdnet.com/>].
10. Meehan, M. *SOA adoption marked by broad failure and wild success*. 2008 [cited; Available from: <http://searchsoa.techtarget.com/>].
11. Roch_, E. *What Constitutes a SOA Failure?* . 2009 [cited; Available from: <http://www.toolbox.com/>].
12. Kavis, M. *10 Reasons why Enterprise Initiatives fail*. 2008 [cited; Available from: <http://www.toolbox.com/>].
13. Welke_, R.a.H., _Rudy, *Service Oriented Architecture Maturity*. IEEE 2010.
14. *Lenguaje de programación JAVA*. [cited; Available from: <http://www.ecured.cu/index.php/Java>].
15. programación, L.d. *Lenguajes de programación*. 2009 [cited 10 de diciembre 2011]; Available from: <http://www.lenguajes-de-programacion.com/programacion-java.shtml>.
16. Gordo, A.C. *Analizador Gráfico de Red (sniffer) entorno GNU*. 2008 [cited 12/1/2012]; Available from: <http://es.scribd.com/doc/77126003/26/La-libreria-Swing>.
17. *Sistemas Gestores de Base de Datos*. 2012 [cited 23 de enero 2012]; Available from: http://www.ecured.cu/index.php/Sistema_Gestor_de_Base_de_Datos.
18. 2010 [cited 2012; Available from: <http://packages.debian.org/es/lenny/postgresql-8.3>].
19. Rup, M.X.V.M. 2009 [cited 2012; Available from: http://metodologiaxpvsmetodologiarup.blogspot.com/2008/04/metodologia-xp_9714.html].
20. Patricio Letelier, M.C.P. (2009) *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. **Volume**,
21. Carlos Sánchez González. 2009 [cited 2012; Available from: <http://oness.sourceforge.net/proyecto/html/ch05.html>].
22. Castillo, O., Figueroa, Daniel y Sevilla, Hector. *Programacion Extrema. Programacion Extrema*. [cited 15 de febrero 2012]; Available from: <http://programacionextrema.tripod.com/index.htm>.
23. Périssé, M.C., *Una Metodología Simplificada*. 2001.
24. EcuRed, P.d. *Visual Paradigm*. [cited; Available from: http://www.ecured.cu/index.php/Visual_Paradigm].
25. Group, O.M. *UML® Resource Page*. [cited 2012; Available from: <http://www.uml.org/>].

26. Orizondo, A.C., *Abordando el reto de crear Sistemas de Información empleando Arquitecturas Empresariales y Arquitecturas Orientadas a Servicios desde un enfoque evolutivo más completo, integral y flexible*. 2012.
27. BECK, F. M. Y. K. *Planeando en Programación Extrema*, 2000.
28. VALDÉS, D.P. *Los diferentes lenguajes de programación para la Web*. 2007 [cited 2012; Available from: <http://www.maestrosdelWeb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-Web/>].
29. *Arquitectura tres capas*. 2010 [cited; Available from: <http://kernelerror.net/programacion/php/arquitectura-3-capas/>].
30. *Tarjetas CRC*. 2010 [cited 25 de enero 2012]; Available from: <http://lsi.ugr.es/~mvega/docis/crc.pdf>.
31. Campo*, G.D., *Patrones de Diseño, Refactorización y Antipatrones. Ventajas y Desventajas de su Utilización en el Software Orientado a Objetos*. 2009.
32. Sophia. *Patrones GRASP*. 2008 [cited; Available from: http://sophia.javeriana.edu.co/~lcdiaz/ADOO2006-3/grasp_cpaternostro-lvargas-jviafara.pdf].

Anexos

Anexo #1 Historias de Usuario

Historia de Usuario	
Número: 2	Nombre: Gestionar Área de Medida
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: Solo el administrador es el encargado de las acciones pertinentes sobre un área de medida, como son la adicción, la modificación y la eliminación.	
Observaciones:	

Anexo 1 Historia de usuario Gestionar Área de Medida

Historia de Usuario	
Número: 3	Nombre: Gestionar Dimensión
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: Solo el administrador es el encargado de las acciones pertinentes sobre una dimensión, como son la adicción, la modificación y la eliminación.	
Observaciones:	

Anexo 2 Historia de usuario Gestionar Dimensión

Historia de Usuario	
Número: 4	Nombre: Gestionar Categoría
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: Solo el administrador es el encargado de las acciones pertinentes sobre una categoría, como son la adicción, la modificación y la eliminación.	
Observaciones:	

Anexo 3 Historia de usuario Gestionar Categoría

Historia de Usuario	
Número: 5	Nombre: Gestionar Atributo Cualitativo
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: Solo el administrador es el encargado de las acciones pertinentes sobre un atributo cualitativo, como son la adicción, la modificación y la eliminación.	
Observaciones:	

Anexo 4 Historia de usuario Gestionar Atributo Cualitativo

Historia de Usuario	
Número: 6	Nombre: Gestionar Pregunta
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: Solo el administrador es el encargado de las acciones pertinentes sobre una pregunta, como son la adicción, la modificación y la eliminación.	
Observaciones:	

Anexo 5 Historia de usuario Gestionar Pregunta

Historia de Usuario	
Número: 7	Nombre: Gestionar Valor
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: Los administradores pueden realizar la acción de adicionar, eliminar y modificar un valor que represente el cuantitativamente a una pregunta.	
Observaciones:	

Anexo 6 Historia de usuario Gestionar Valor

Historia de Usuario	
Número: 8	Nombre: Gestionar Peso
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 1	Iteración Asignada: 1
Descripción: Los administradores pueden realizar la acción de adicionar, eliminar y modificar un peso que cuantifique a algún elemento del modelo de madurez para SOA.	
Observaciones:	

Anexo 7 Historia de usuario Gestionar Peso

Historia de Usuario	
Número: 11	Nombre: Gestionar Usuario
Usuario: Administrador	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Baja
Puntos Estimados: 2	Iteración Asignada: 3
Descripción: El administrador se encarga de añadir, eliminar o modificar un usuario del sistema.	
Observaciones:	

Anexo 8 Historia de usuario Gestionar Usuario

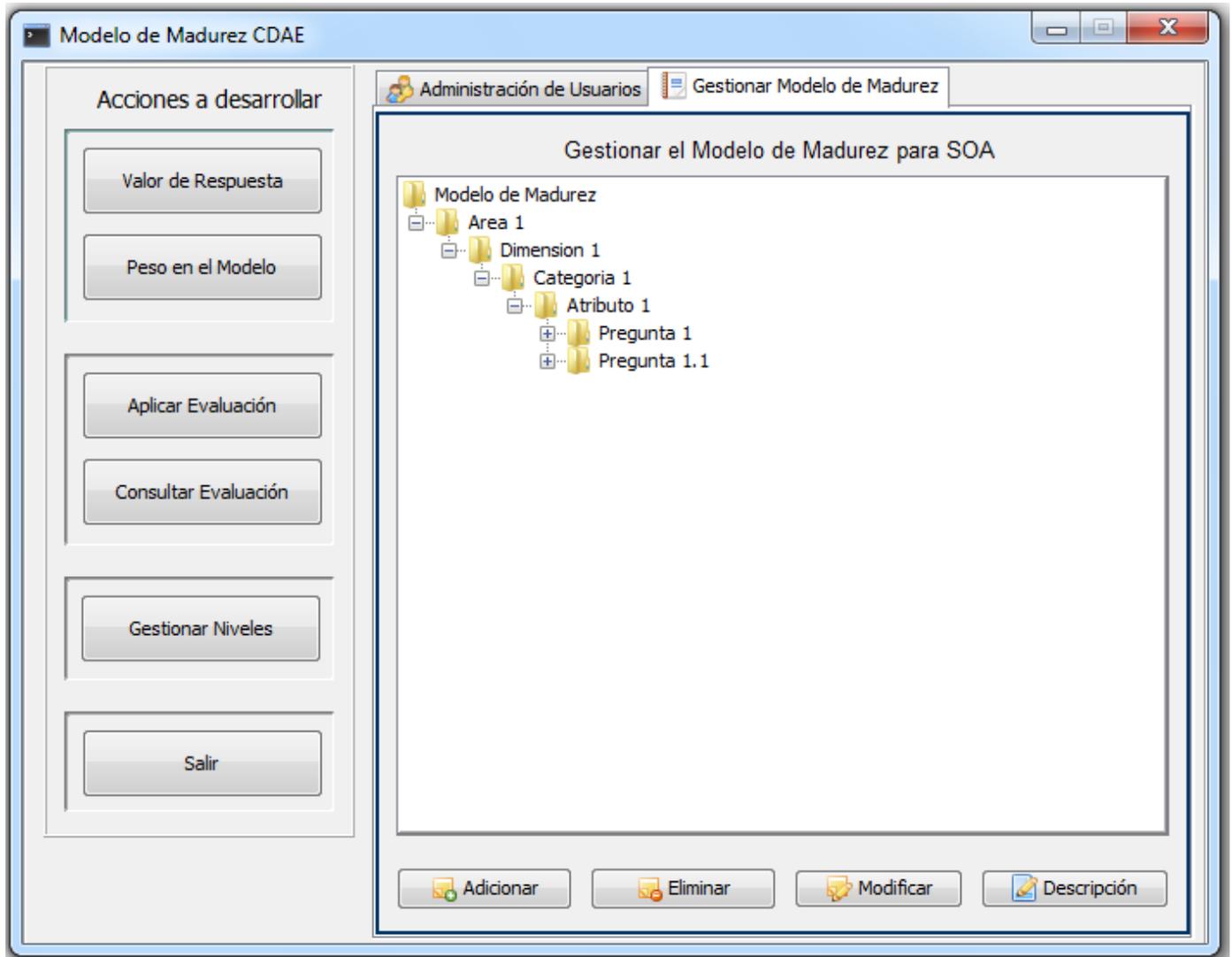
Historia de Usuario	
Número: 12	Nombre: Autenticar
Usuario: Administrador	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Baja
Puntos Estimados: 2	Iteración Asignada: 3
Descripción: Los administradores deben autenticarse para poder tener acceso al sistema, para el proceso de autenticación deben de especificar su usuario y contraseña.	
Observaciones:	

Anexo 9 Historia de usuario Autenticar

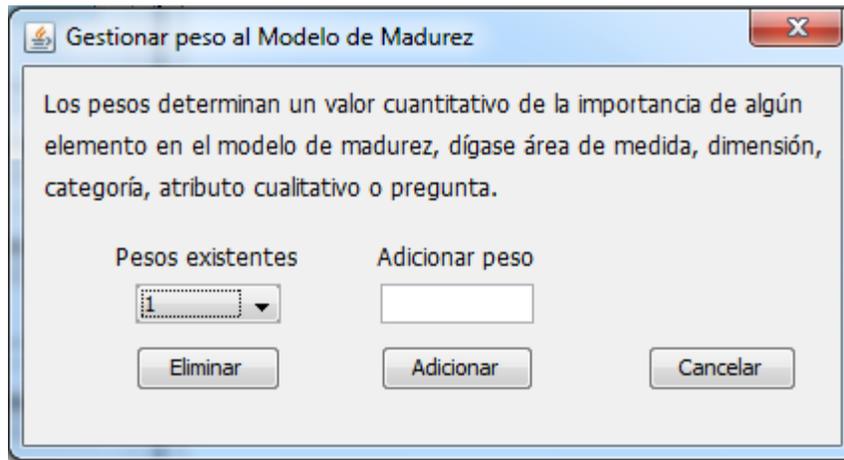
Historia de Usuario	
Número: 13	Nombre: Gestionar Nivel
Usuario: Administrador	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Baja
Puntos Estimados: 2	Iteración Asignada: 3
Descripción: Los administradores pueden realizar la acción de adicionar, eliminar y modificar un nivel de madurez para SOA en el modelo propuesto.	
Observaciones:	

Anexo 10 Historia de usuario Gestionar Nivel

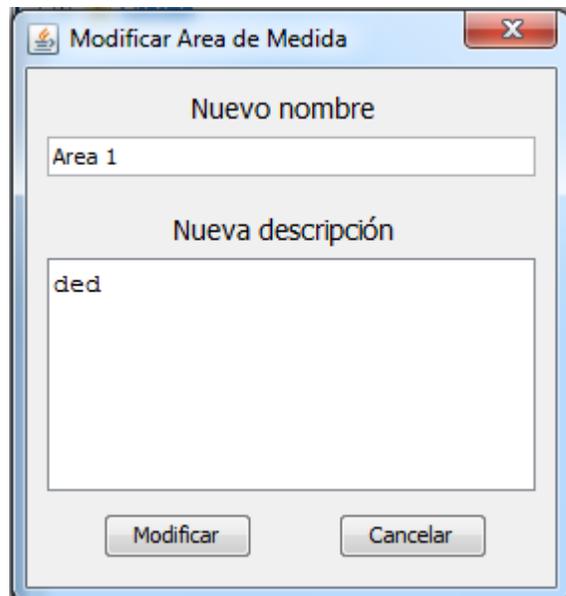
Anexo #2 Prototipos de interfaz



Anexo 11 Interfaz principal



Anexo 12 Interfaz gestionar peso



Anexo 13 Interfaz modificar área de medida

Modificar Dimensión

Nuevo nombre
Dimension 1

Nueva descripción
dswd

Nueva área
Area 1

Nuevo peso
1

Modificar Cancelar

Anexo 14 Interfaz modificar dimensión