

IMPLEMENTACIÓN DE UN SOFTWARE ORIENTADO A LA WEB QUE GESTIONE LA APLICACIÓN DE LA TÉCNICA DE CALIDAD SEIS SIGMA AL PROCESO DE DESARROLLO DE SOFTWARE, SOBRE LA PLATAFORMA JAVA EE 5.0 EMPLEANDO UN FRAMEWORK INTEGRADOR JBOSS SEAM 2.2.0

Cecilia Hinojosa Raza¹, Edison Lascano², Luis Yacchirema Espín³

1 Carrera de Ingeniería de Sistemas e Informática, Escuela Politécnica del Ejército, Sangolquí, Ecuador, chinojosa@espe.edu.ec

2 Carrera de Ingeniería de Sistemas e Informática, Escuela Politécnica del Ejército, Sangolquí, Ecuador, edison_lascano@yahoo.com

3 Carrera de Ingeniería de Sistemas e Informática, Escuela Politécnica del Ejército, Sangolquí, Ecuador, lmye_1984@yahoo.com

RESUMEN

En el presente proyecto se desarrolló un software orientado a la web que sirva como herramienta de apoyo en el control del proceso de desarrollo de software, y complemente la “Guía Básica para la Aplicación de la Técnica de Calidad Seis Sigma al Proceso de Desarrollo de Software”. La Guía fue desarrollada por estudiantes de la Escuela Politécnica del Ejército como proyecto de graduación, tomando en cuenta el modelo DMAIC (Definir, Medir, Analizar, Mejorar, Controlar) de Seis Sigma una metodología de referencia para llevar la calidad hasta niveles próximos a la perfección. Al disponer de una metodología que permita la aplicación de buenas prácticas en los procesos, y una herramienta que permita llevar un control de su aplicabilidad, se conseguirá incrementar notablemente una mejora continua en los procesos de desarrollo de software, y con el transcurso del tiempo se podrá disponer de información muy importante que sirva para analizar posibles falencias en proyectos anteriores, y así tomar medidas preventivas y/o correctivas que fortalezcan dichos procesos. Este proyecto también está enfocado en dar a conocer algunas alternativas de tecnologías que pueden ser utilizadas en desarrollos web, es así, que se trabajó sobre la plataforma Java EE 5.0, y se empleó un framework web llamado JBoss Seam. Además, para cubrir todo el ciclo de vida de desarrollo del software, se aplicó una metodología ágil denominada ICONIX.

Palabras Clave:

Guía Básica: Documento en el cual se definen seis fases que buscan mejorar la eficiencia en el control de calidad del proceso de desarrollo de software, a través de la aplicación de la técnica de calidad Seis Sigma.

Seis Sigma: Una metodología de mejora de procesos.

Iconix: Metodología ágil que cubre todo el ciclo de desarrollo de software.

JBoss Seam: Framework web que facilita la integración de las especificaciones principales de la Plataforma Java EE 5.0 (EJB 3.0 y JSF).

ABSTRACT

In this project we developed a web-oriented software that serves as a support tool in controlling the software development process, and complement the "Basic Guide to application of the Six Sigma Quality Technique to the software development process". The guide was developed by students from the Escuela Politécnica del Ejercito as a graduation project, taking into account the DMAIC (Define, Measure, Analyze, Improve, and Control) model as a reference methodology to bring quality to levels close to perfection. Having a methodology that allows the application of best practices in processes, and a tool to keep track of their applicability, notably can be achieved by increasing the continuous improvement in software development processes, and over time we can have very important information that serves to analyze possible failures in previous projects, and thus taking preventive and corrective measures to strengthen these processes. This project is also focused on alternatives to present some Technologies that can be used in web development, and that is why we work with the Java EE 5.0 platform, and use a web framework called JBoss Seam. In addition, to cover the entire lifecycle of software development, an agile methodology called ICONIX is applied.

Key Words:

Basic Guide: Document that identifies six stages that aim to improve efficiency in quality control of the software development process, through the quality technique application, Six Sigma.

Six Sigma: Process of methodology improvement.

Iconix: Agile methodology covering the entire software development cycle.

JBoss Seam: Web framework that facilitates the integration of the main specifications of the Java EE 5.0 platform (EJB 3.0 y JSF).

1. INTRODUCCIÓN

El proceso de desarrollo de software es muy complejo y lleva consigo alto riesgo, existen diferentes fases que demandan organización que serán compartidas por un grupo de personas con una clara distribución de responsabilidades entre los diferentes roles que se dan en el proceso de producción. La inadecuada utilización de recursos, adopción de malas prácticas en los procesos, presupuesto de tiempo y dinero insuficientes, etc., son factores que incrementan el fracaso de un proyecto.

En vista de esta problemática es importante optimizar el proceso de desarrollo de software, porque los sistemas informáticos están sujetos a diferentes innovaciones en la industria de la tecnología. Esta realidad impone a la empresas desarrolladoras de software a la adopción de varios modelos, técnicas y herramientas que permitan adaptarse fácilmente a los cambios tecnológicos, y así obtener aplicaciones en menor tiempo, más vistosas y de menor costo; pero sobre todo que sean fáciles de mantener, extender y modificar, tomando en cuenta que los requerimientos de los usuarios son cada vez más exigentes. Bajo estas consideraciones, se desarrolló una herramienta que sirva como apoyo en el control del proceso de desarrollo de software, y además complemente una "Guía para la Aplicación de la Técnica de Calidad Seis Sigma al Proceso de Desarrollo de Software" [1], teniendo en cuenta que el objetivo de esta guía es mejorar el nivel de eficiencia en el control de calidad del proceso de desarrollo de software, aplicando la técnica de calidad SEIS SIGMA.

La mayoría de herramientas y tecnologías empleadas en el desarrollo de este proyecto son open source, y fueron utilizadas con el fin de fomentar su aplicabilidad en proyectos de desarrollo de software basados en web.

El resto del artículo ha sido organizado como sigue: La sección 2 describe la metodología aplicada para cubrir el ciclo de vida del proceso de desarrollo del software, además, de las tecnologías y herramientas empleadas. En la sección 3 se resume el trabajo de desarrollo del software dividido en cuatro fases que son Análisis de Requerimientos, Análisis y Diseño Preliminar, Diseño Detallado, e Implementación. La sección 4 resume los resultados obtenidos en una encuesta, ejecutada con el fin de evaluar el software expuesto en presente artículo. La sección 5 expone los trabajos relacionados en el estudio de SEIS SIGMA. Y por último la sección 6 presenta las conclusiones y trabajo futuro que se derivan del presente proyecto.

2. METODOLOGÍA, TECNOLOGÍAS Y HERRAMIENTAS APLICADAS

2.1 Metodología

Con el fin de promover buenas prácticas en proyectos de desarrollo de software, se aplicó una metodología ágil denominada ICONIX que unifica un conjunto de métodos de orientación a objetos. Esta metodología fue elaborada a partir de una síntesis del Proceso Unificado de los "Three Amigos" (Booch, Rumbaugh y Jacobson), y ofrece el soporte de UML, dirigida por casos de uso como RUP. ICONIX se divide en dos flujos de trabajo que son Estático y Dinámico (ver Fig. 1), y se fundamenta en cuatro fases para cubrir todo el ciclo de vida del proceso de desarrollo de software que son Análisis de Requerimientos, Análisis y Diseño Preliminar, Diseño Detallado, e Implementación [2].

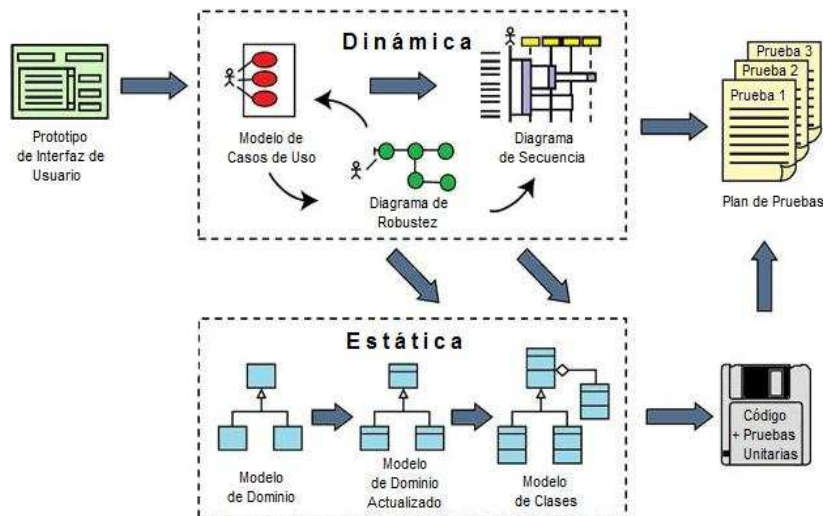


Fig. 1: Flujos de Trabajo de Iconix

2.2 Tecnologías

La mayoría de tecnologías utilizadas en el desarrollo del software, fueron especificaciones de la Plataforma Java EE 5.0, aunque también se empleó un framework web de implementación denominado JBoss Seam en su versión 2.2.0.

A continuación se describe cada una de estas tecnologías.

2.2.1 Plataforma Java EE 5.0

La Plataforma Java es un entorno orientado a ejecutar aplicaciones desarrolladas en lenguaje de programación java. En este contexto, Java EE es un entorno de programación, que define un conjunto de especificaciones de APIs Java, para el desarrollo de aplicaciones distribuidas multicapa (3-capas, n-capas) basadas en web, empleando el lenguaje de programación java.

La especificación de Java define los siguientes componentes Java EE [3]:

- Las aplicaciones clientes y Applets, son componentes que se ejecutan en el cliente.
- Java Servlets, JavaServer Faces (JSF), y JavaServer Pages (JSP), son componentes web que se ejecutan en el lado del servidor.
- Enterprise JavaBeans (EJB), son componentes (beans empresariales) de negocio y persistencia que se ejecutan en el lado del servidor.

Generalmente las aplicaciones Java EE están hechas a base de componentes que se encuentran corriendo en diferentes máquinas como se observa en la Fig. 2.

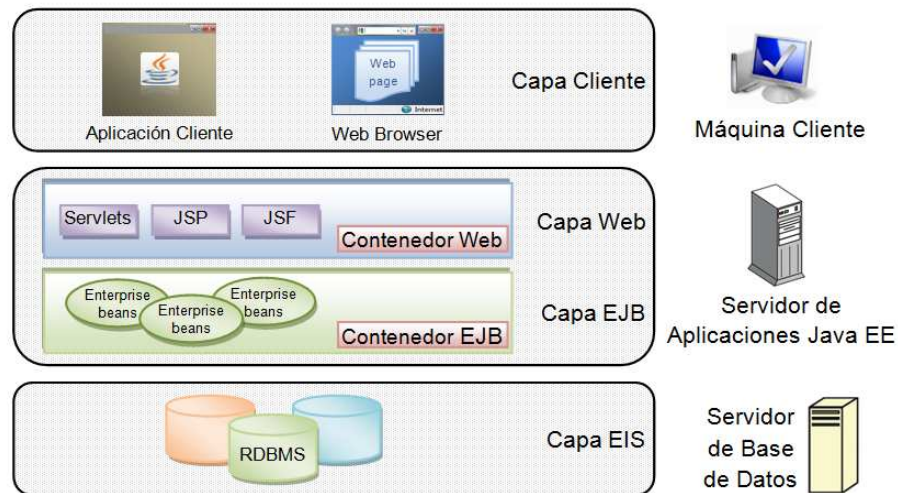


Fig. 2: Arquitectura Java EE

2.2.2 JDBC (Java Database Connectivity)

JDBC es un API estándar de Java para ejecutar sentencias SQL. Está formado por un conjunto de clases e interfaces escritas en el lenguaje de programación java para facilitar el acceso a una gran variedad de base de datos relacionales (RDBMS). Diseñada para ser independiente de la plataforma, y provee una interface de programación única, que independiza a las aplicaciones del motor de base de datos usado.

Para acceder a una base de datos (desde programas java) se requiere de un driver (controlador) JDBC, en el cual se encuentran definidas un conjunto de interfaces que el proveedor de base de datos implementa. Existen cuatro tipos de controladores que son: Puente JDBC-ODBC más driver ODBC, Driver Java parcialmente Nativo, Driver Java nativo JDBC-Net, y Driver Java puro y protocolo-nativo. Este último fue empleado en el desarrollo del software expuesto en el presente artículo.

2.2.3 JPA (Java Persistence API)

JPA es un API de persistencia desarrollada para la plataforma Java EE e incluida en el estándar EJB 3 (ver Fig. 3). Pero también puede funcionar independientemente del resto de componentes EJB 3, hasta puede ser usado por una aplicación Java SE (aplicación desktop).

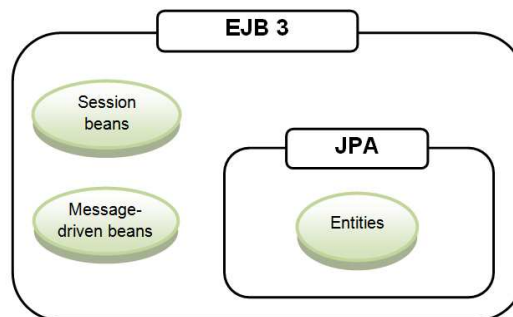


Fig. 3: Organización del API EJB 3.0

El estándar JPA define [4]:

- El mapeo objeto relacional (ORM), mediante metadata que mapea entidades (objetos java) a tablas relacionales. Esto hace que no sea necesario el uso de ficheros descriptores XML.
- La interface EntityManager, que define una API estándar para realizar las operaciones de persistencia (CRUD) de las entidades.
- El Java Persistence Query Language (JPQL), para consultas y lecturas de datos de aplicación persistidos (algo así como un SQL orientado a objetos).

2.2.4 JSF (JavaServer Faces)

La tecnología JSF es un marco de trabajo (framework para capa web) que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. Este framework implementa el patrón MVC (Modelo-Vista-Controlador), el cual separa las aplicaciones en tres capas diferenciadas (ver Fig. 4): datos (Modelo), interfaz de usuario (Vista) y lógica de control (Controlador). Esto permite que el mantenimiento de las aplicaciones JSF sea sencillo.

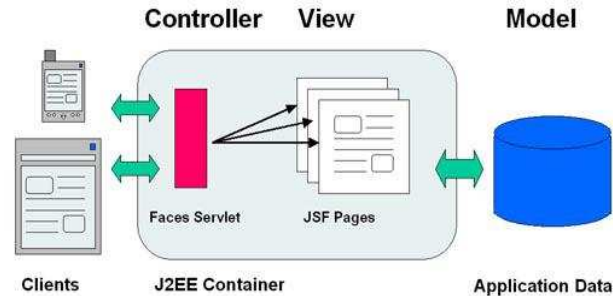


Fig. 4: Modelo MVC

2.2.5 JBoss Seam Framework

Seam es un framework open source que reúne los estándares de Java EE para que puedan trabajar como una solución integrada. Seam integra tecnologías como JavaScript asíncrono y XML (AJAX), JavaServer Faces (JSF), Java Persistence API (JPA)/Hibernate, Enterprise Java Beans (EJB 3.0) y Business Process Management (BPM). Seam ha sido diseñado desde cero para eliminar la complejidad, tanto en la arquitectura y los niveles de la API. Permite a los desarrolladores ensamblar aplicaciones web complejas con simples clases Java anotadas, un rico conjunto de componentes de interfaz de usuario, y muy poco XML [5].

En la siguiente figura se ilustra la tecnologías que Seam es capaz de reunir en una aplicación típica [6].

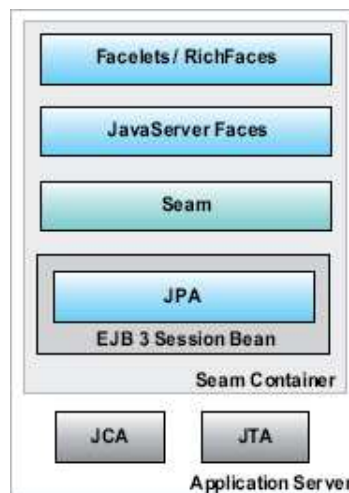


Fig. 5: Algunas tecnologías que integra Seam

2.3 Herramientas

2.3.1 Enterprise Architect 7.5

EA (Enterprise Architect) es una herramienta CASE, utilizada para la construcción de sistemas de software, para el modelado de procesos de negocios, y para objetivos de modelado más generalizados. EA está basada en la especificación UML 2.1, ofreciendo soporte para diferentes tipos de diagramas de UML como: diagramas estructurales (clase, objeto, compuesto, paquete, componente, y despliegue.), diagramas de comportamiento (casos de uso, comunicación, secuencia, descripción de la interacción, actividad, estado, y tiempo), y diagramas extendidos como: análisis (actividad simple), personalizado (para requisitos, cambios, y UI). Además, permite la

generación e ingeniería inversa de código fuente, soporte para casos de prueba (JUnit y NUnit), control de versiones (empleando CVS o SCC), generación de documentación en formatos HTML y RTF, entre otras prestaciones. Básicamente, esta herramienta fue utilizada porque facilita la aplicación de la metodología ICONIX.

2.3.2 Netbeans 6.7.1

Es un Entorno de Desarrollo Integrado (IDE) que soporta el desarrollo de todos los tipos de aplicaciones Java. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Todas las funciones del IDE son provistas por módulos donde cada uno de estos provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones.

2.3.3 Subversion 1.6.6

Esta herramienta fue empleada específicamente para controlar las distintas versiones del código fuente del software. Subversion es un sistema de control de versiones, utilizado para el mantenimiento de código fuente, documentación técnica y páginas web. Es decir, Subversion maneja ficheros y directorios a través del tiempo. Subversion puede ser utilizado en cualquier lenguaje de programación y para cualquier tipo de proyecto, implementa un sistema de ficheros versionado "virtual" que sigue los cambios sobre árboles de directorios completos a través del tiempo. Tener en cuenta que se instaló el soporte de Subversion en Netbeans.

Para facilitar la creación de repositorios, usuarios y otras configuraciones se utilizó el paquete VisualSVN Server 2.0.8, que incluye Subversion 1.6.6 y el servidor web Apache 2.2.13 ya configurados y listos para funcionar.

2.3.4 MySQL Server 5.1

Es un Sistema de Administración de Bases de Datos Relacionales, multihilo y multiusuario. MySQL Server trabaja en entornos cliente/servidor o incrustados, funciona en diferentes plataformas, proporciona sistemas de almacenamiento transaccional y no transaccional, tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas, entre otras funciones.

2.3.5 JBoss Application Server 5.1.0

JBoss es un servidor de aplicaciones Java EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo que lo soporte. JBoss implementa todo el paquete de servicios de Java EE (EJB, JMS, JTS/JTA, Servlets/JSP, JNDI).

3. DISEÑO E IMPLEMENTACIÓN

De acuerdo a la metodología ICONIX, el trabajo de desarrollo del software se dividió en cuatro fases que son:

- Fase de Análisis de Requerimientos.
- Fase de Análisis y Diseño Preliminar.
- Fase de Diseño Detallado.
- Fase de Implementación.

3.2 Fase de Análisis de Requerimientos

Se realizó un listado de requerimientos funcionales del sistema que es un requisito indispensable para iniciar esta fase.

Las principales actividades realizadas dentro de esta fase fueron las siguientes:

- Creación de un modelo de dominio (ver Fig. 6). Los objetos de dominio (cosas y conceptos del mundo real) fueron identificados en el listado de requerimientos, y la mayoría de relaciones utilizadas para asociar estos objetos fueron "agregación" y "generalización", ya que ICONIX resalta que casi un 95% de las asociaciones se representan modelándolas como agregación o generalización.

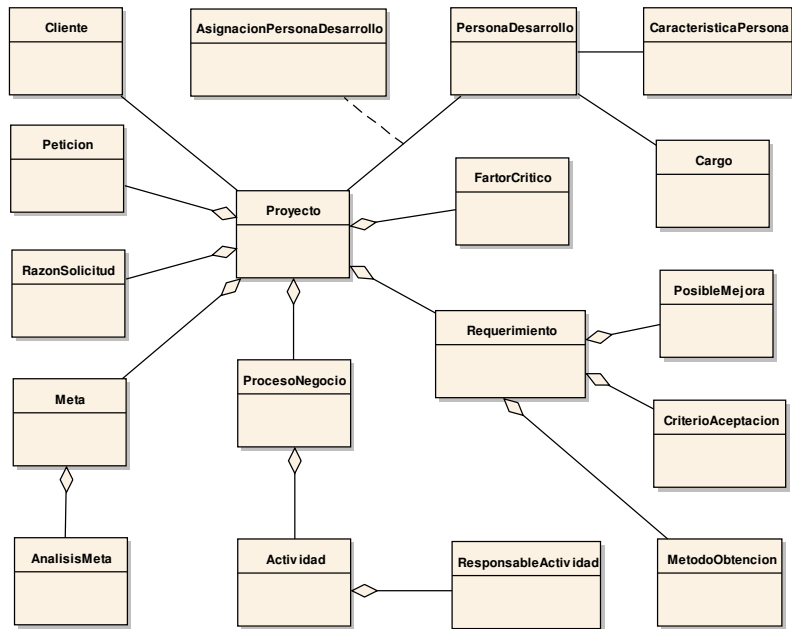


Fig. 6: Modelo de Dominio inicial

- Realización de prototipos de interfaz gráfica de usuario (GUI) con el fin de facilitar la identificación y redacción de los casos de uso. Al realizar los prototipos de GUI se incluyó todos los botones y menús que el usuario puede tocar para generar eventos dentro de un caso de uso, esto facilitó la descripción de los casos de uso. El nombre de estas interfaces fueron utilizadas en las descripciones de los casos de uso como recomienda la metodología ICONIX.
- Elaboración del diagrama de casos de uso que muestra un conjunto de casos de uso, actores y sus relaciones.

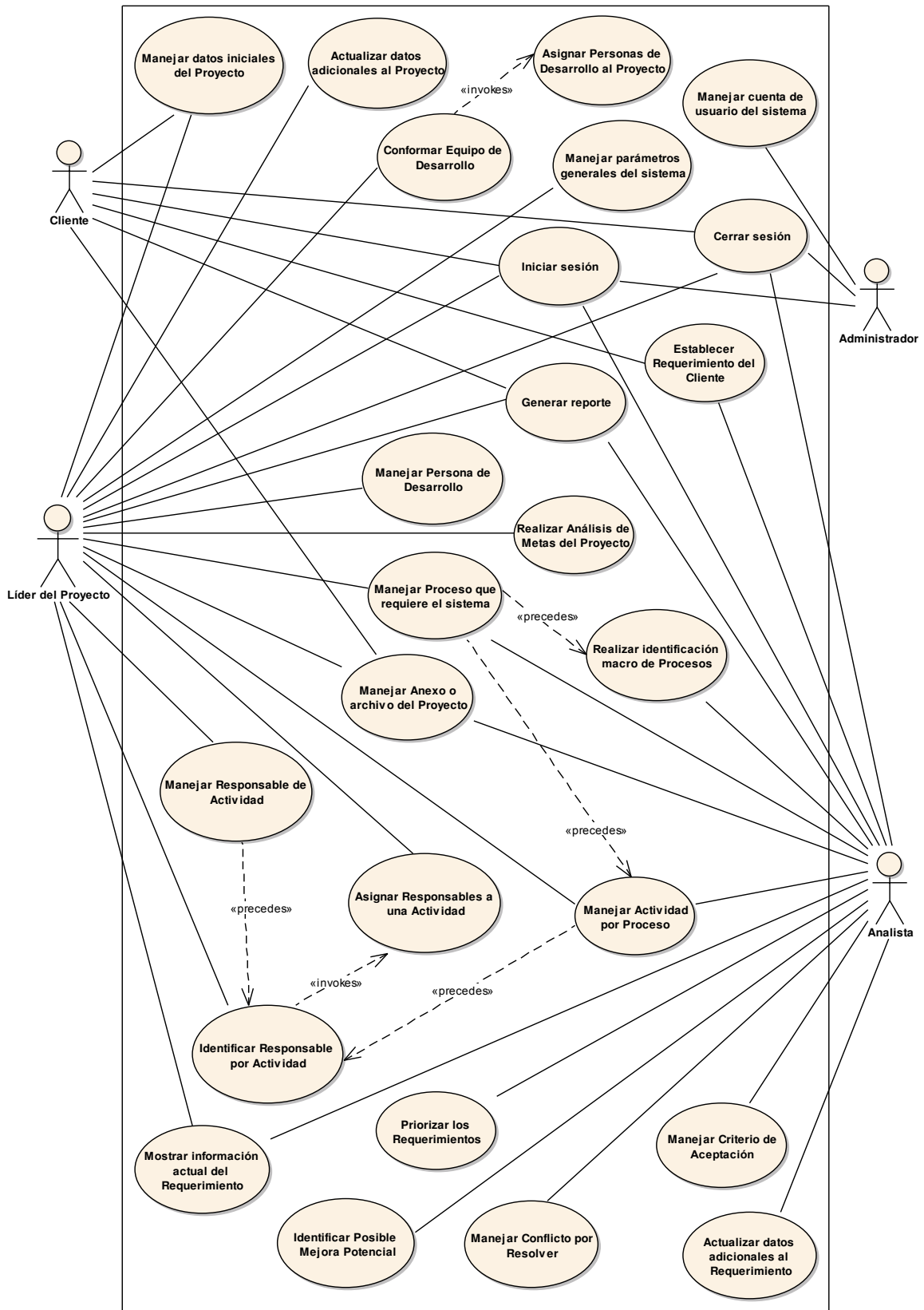


Fig. 7: Modelo de Casos de Uso

- Creación de un diagrama de paquetes con el fin de organizar los casos de uso.

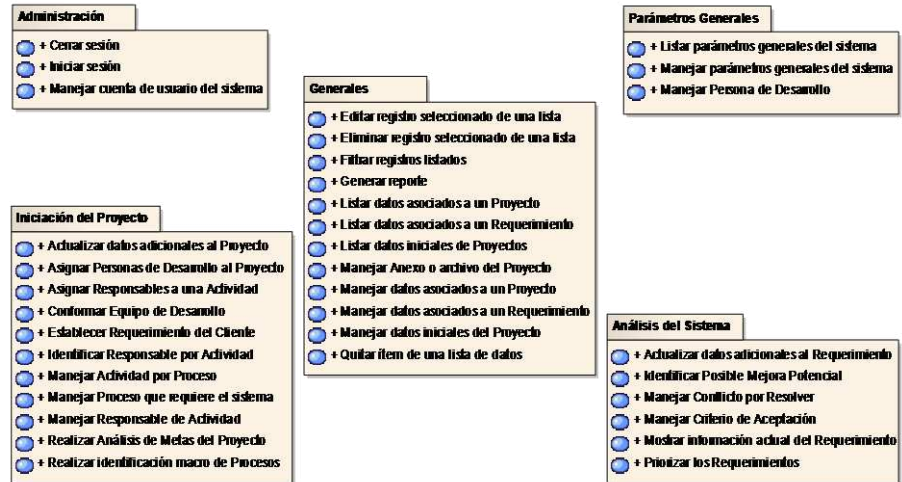


Fig. 8: Diagrama de Paquetes

- Esta fase finalizó definiendo la interacción del usuario con el sistema, es decir, se realizó el primer proyecto de casos de uso (o especificación de casos de uso a nivel de borrador) que luego fue utilizado y refinado en la fase de “Análisis y Diseño Preliminar”.

3.3 Fase de Análisis y Diseño Preliminar

En esta fase se asoció los casos de uso a los objetos, utilizando una técnica denominada “Análisis de Robustez”. Esta técnica fue aplicada siguiendo las diferentes consideraciones de la metodología ICONIX.

Dentro del Análisis de Robustez se realizó un “diagrama de robustez” por cada caso de uso, este diagrama permitió analizar el texto narrativo de los casos de uso, facilitando la identificación de un conjunto de objetos (objetos interfaz, objetos entidad, objetos controladores) participantes en los casos de uso.

Además, mientras se graficaba los diagramas de robustez, se actualizó el modelo de dominio de forma incremental (agregar las nuevas clases de dominio y atributos identificados), y se reescribió los casos de uso.

En la Tabla I, se muestra el formato de la especificación de un caso de uso.

Tabla I: Especificación del caso de uso “Iniciar sesión”

Especificación de Caso de Uso	
Código	01
Caso de Uso	Iniciar sesión
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar conectado al intranet mediante el browser. El usuario debe estar viendo la Página Principal del sistema en el browser.
Flujo de Eventos	<p>Básico</p> <p>El usuario da clic en el enlace “Iniciar sesión” de la Página Principal, y el sistema muestra la Página de Inicio de sesión. El usuario ingresa su nombre de usuario y contraseña y da clic en la opción “Ingresar”.</p> <p>El sistema verifica si existen los datos ingresados en cuenta de usuario. Si estos datos existen, el sistema autentica al usuario en sesión y se direcciona a la Página Principal mostrando un mensaje de bienvenida, y el nombre del usuario autenticado. Además el sistema cambia el enlace “Iniciar sesión” por el enlace “Cerrar sesión”.</p>
	<p>Alternativos</p> <ul style="list-style-type: none"> Datos inválidos: El sistema se direcciona a la misma Página de Inicio de sesión mostrando un mensaje que dice “Nombre de usuario y/o contraseña son incorrectos, inicio de sesión falló”. Usuario cancela inicio de sesión: El usuario da clic en la opción “Cancelar”, y el sistema muestra la Página Principal.
Pos condiciones	<ul style="list-style-type: none"> Se observa el nombre de usuario en sesión en la Página Principal.

En la siguiente figura se ilustra el respectivo diagrama de robustez para el caso de uso descrito en la Tabla I:

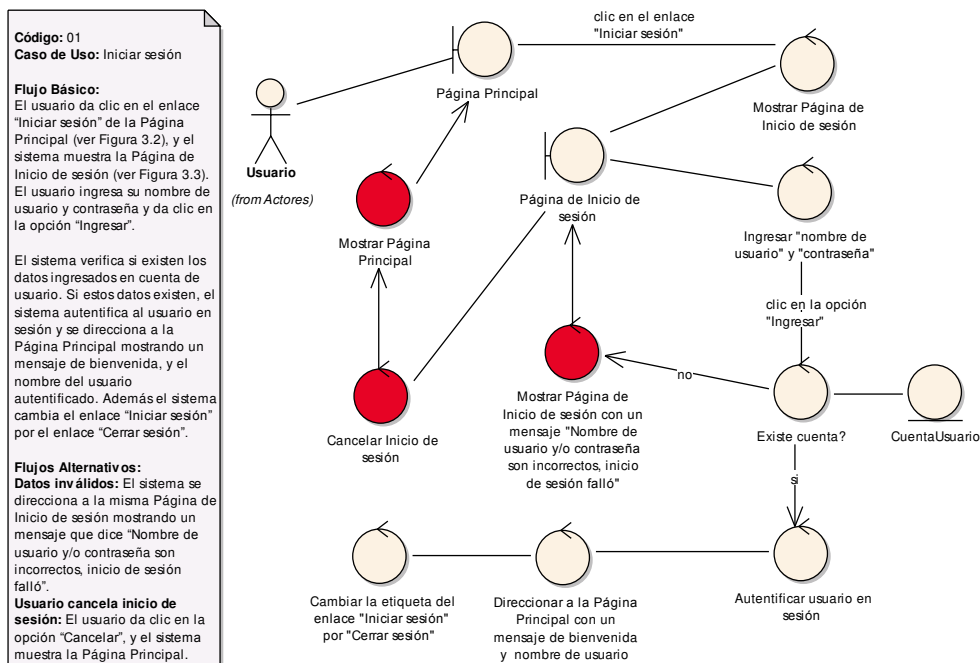


Fig. 9: Diagrama de Robustez para el caso de uso "Iniciar sesión"

3.4 Fase de Diseño Detallado

El diseño detallado es conocido como la asignación de comportamiento, es decir, la asignación de las funciones del software. Para conducir el diseño detallado se realizó un diagrama de secuencia por cada caso de uso. Antes de realizar los diagramas de secuencia se diseñó la arquitectura técnica del software (ver Fig. 10).

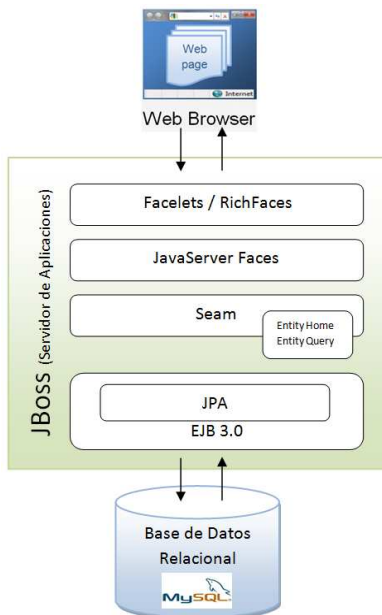


Fig. 10: Arquitectura Java EE con Seam

A continuación se expone algunos aspectos relevantes de la arquitectura:

- Al trabajar con JavaServer Faces (JSF) es recomendable usar Facelets como controlador de vista en lugar de JavaServer Pages (JSP). Facelets es un framework open source para plantillas (templates) centrado en JSF, motivo por el cual se integran de manera muy fácil. Facelets no depende del contenedor web, el proceso de compilación es más rápido que JSP, y soporta EL (Expression Language).

- Se empleó RichFaces para añadir la capacidad de Ajax (Asynchronous JavaScript And XML - JavaScript asíncrono y XML) dentro de las aplicaciones JSF. RichFaces es un framework open source que provee una librería de componentes visuales para JSF, y permite la integración de funcionalidades Ajax en dichos componentes mediante el soporte de la librería Ajax4JSF.
- En un proyecto Seam las vistas de JSF terminan con el sufijo .xhtml, siendo esta extensión el sufijo por defecto utilizado por Facelets.
- Se utilizó un “framework de clases” (colección especializada de componentes plantillas) suministrada por Seam para construir controladores de página que realizan operaciones CRUD y consultas en instancias de entidades. El conjunto jerárquico de estas clases se ilustra en la siguiente figura.

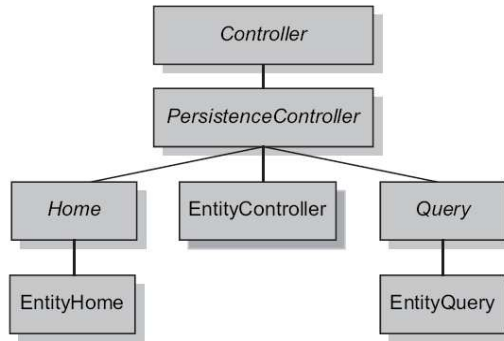


Fig. 11: Jerarquía de clases del framework Seam

Tomando en cuenta estos aspectos de la arquitectura se procedió a graficar un diagrama de secuencia por cada caso de uso. La siguiente figura muestra el diagrama de secuencia del caso de uso “Iniciar sesión” expuesta en la Tabla I.

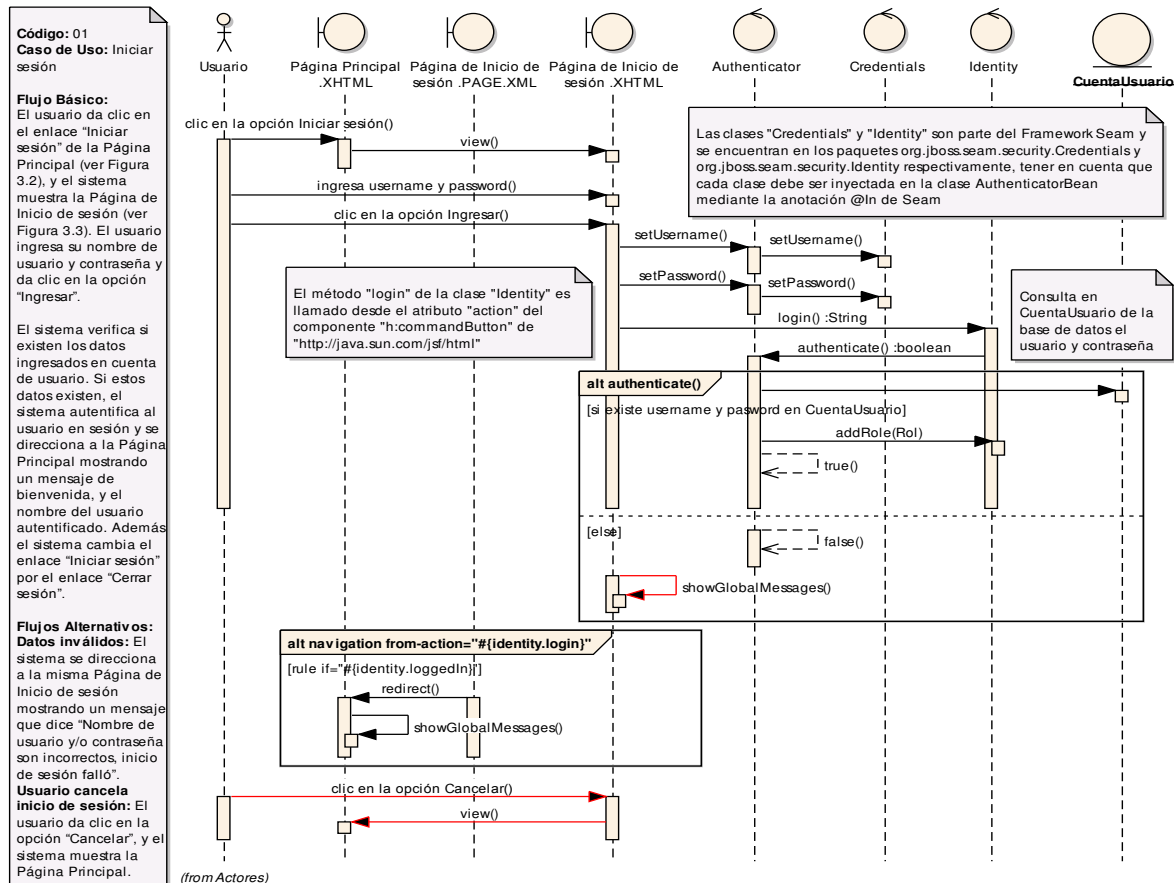


Fig. 12: Diagrama de secuencia para el caso de uso “Iniciar sesión”

Para empezar la siguiente fase se finalizó el modelo estático, adicionando los detalles del diseño en el diagrama de clases.

3.5 Fase de Implementación

En esta fase se realizó la codificación del software en base al análisis y diseño realizado en las fases anteriores. Antes de empezar la codificación se instaló y configuró el ambiente de desarrollo con las herramientas mencionadas en la sección 2.3 del presente artículo. Luego se creó la estructura inicial del proyecto web con ayuda de la herramienta “Seam Generator” que forma parte del framework Seam. Posteriormente, la edición de este proyecto se realizó en Netbeans IDE.

Finalmente, para probar el software y realizar las pruebas funcionales con el usuario, se desplegó este proyecto en el servidor de aplicaciones JBoss.

4. RESULTADOS

Se realizó una encuesta con el fin de evaluar el software expuesto en el presente artículo. Para este fin se subió temporalmente al Internet el aplicativo y la encuesta.

Luego se reunió a un grupo de personas con la finalidad de explicarles en su totalidad el funcionamiento del software, y concluir llenado la encuesta. También se realizó la encuesta a otro grupo de personas de forma virtual (online), enviándoles el link de acceso del aplicativo, a fin de que utilicen el aplicativo y llenen la encuesta.

Los resultados que se muestran a continuación, fueron procesados a través de una hoja de cálculo del programa Google Docs, y corresponden a una sección de la encuesta destinada a obtener una apreciación general del usuario sobre el aplicativo.

Fig. 13: En que facilitó las ayudas proporcionadas por la herramienta. **Fig. 14:** Apreciación de los mensajes que orientan al usuario en el ingreso de datos. **Fig. 15:** Nivel de intuición que adquiere el usuario con las ayudas proporcionadas por la herramienta. **Fig. 16:** Rango de porcentaje que representa el grado de aceptación del usuario para utilizar la herramienta en sus proyectos de desarrollo de software. **Fig. 17:** Rango de errores encontrados al utilizar la herramienta (fueron corregidos). **Fig. 18:** Rango en el cual el usuario recomienda a otras personas el uso de esta herramienta (rango del 1 al 5, siendo 1 el más bajo).

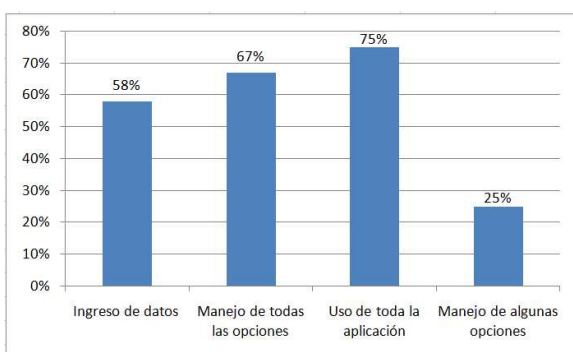


Fig. 13: Ayudas

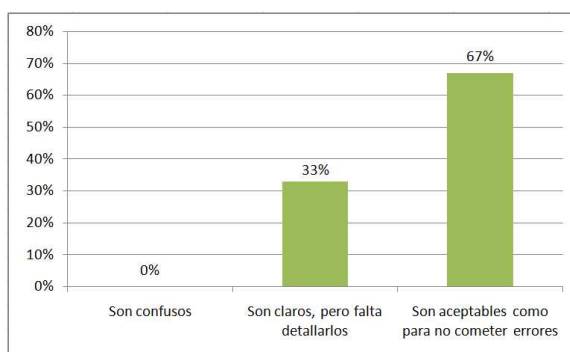


Fig. 14: Mensajes de manejo de datos

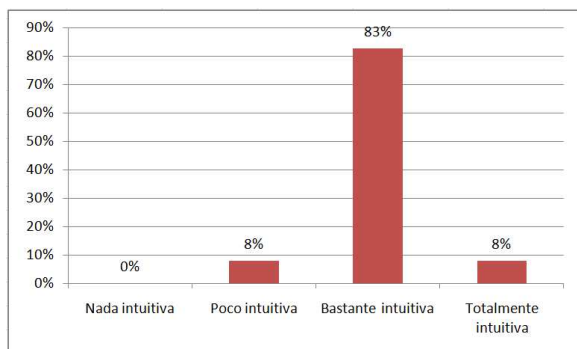


Fig. 15: Uso de la herramienta

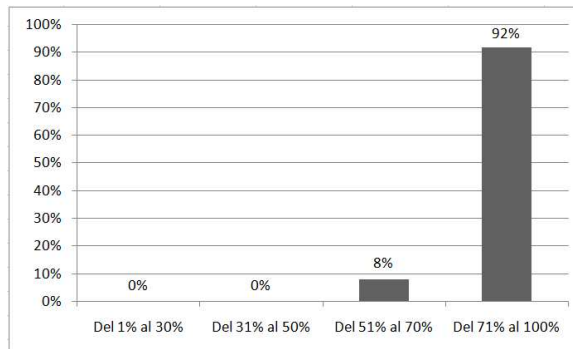


Fig. 16: Aceptación para probar la herramienta

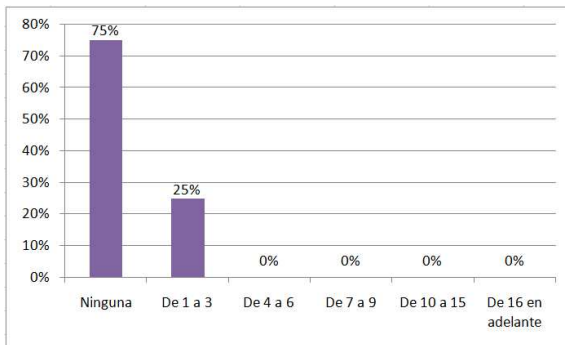


Fig. 17: Errores encontrados (fueron corregidos)

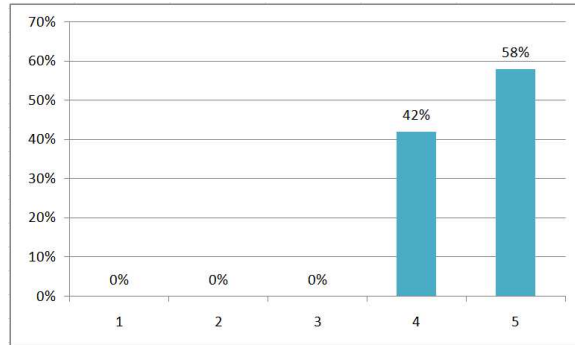


Fig. 18: Rango de recomendación de la herramienta

5. TRABAJOS RELACIONADOS

Previa a la realización de este proyecto, estudiantes de la Escuela Politécnica del Ejército realizaron un trabajo de investigación que se titula “Estudio de la aplicación de técnicas de calidad en el proceso de desarrollo de software” cuyo objetivo principal fue desarrollar una guía que permita mejorar el nivel de eficiencia en el control de calidad del proceso de desarrollo de software, a través de la aplicación de la técnica Seis Sigma, siendo esta guía la base para el desarrollo del proyecto expuesto en este artículo.

6. CONCLUSIONES Y TRABAJO FUTURO

Utilizando la guía desarrollada por los estudiantes de la ESPE se desarrolló una herramienta enfocada en controlar el proceso de desarrollo de software. Quedando automatizadas las dos primeras fases (Análisis de Requisitos y Iniciación del Proyecto) de las seis que propone el modelo de la guía.

En la obtención de los requerimientos funcionales del software, se aplicó algunas directrices propuestas por la metodología ICONIX que facilitaron la ejecución de esta tarea, y es así, que mediante la técnica de “Análisis de Robustez” se logró efectuar un análisis profundo de los casos de uso.

La codificación de los componentes de la capa de negocio del software se aceleró considerablemente al emplear la especificación EJB 3.0 de la plataforma Java EE 5.0, debido a que las transacciones, la persistencia, la concurrencia, u otros servicios de nivel de sistema, fueron responsabilidad del servidor de aplicaciones; esto permitió centrarse exclusivamente en la codificación de la lógica de negocio del sistema.

La herramienta CASE Enterprise Architect facilitó la aplicación de la metodología ICONIX, ya que esta herramienta dispone de un ejemplo muy completo que describe todas las fases de ICONIX.

El análisis de una encuesta ejecutada con el fin evaluar la herramienta expuesta en el presente artículo, reflejó que es intuitiva en un 83% gracias a las ayudas proporcionadas para el uso de sus diferentes opciones; los mensajes que orientan al usuario en el ingreso de datos tuvo una aceptación del 67%; es importante mencionar que un 92% de usuarios estarían dispuestos en probar la herramienta en sus proyectos de desarrollo de software, además han recomendado el uso de esta herramienta seleccionando un rango del 1 al 5 siendo el 1 el más bajo (rangos seleccionados: 4 - 42%, 5 - 58%).

Como trabajo futuro se propone la automatización de las fases restantes (Diseño del Sistema, Construcción, Evaluación y Aseguramiento de Calidad, e Implementación) consideradas en la Guía que fue la base para el desarrollo del presente proyecto.

7. REFERENCIAS BIBLIOGRÁFICAS

- [1]. Monge Daniel, A., y Mosquera Lourdes, V. (2009). Estudio de la Aplicación de Técnicas de Calidad al Proceso de Desarrollo de Software (Anexo E, código DCC- AIS - G1)
- [2]. Doug Rosenberg and Matt Stephens. (2007). Use Case Driven Object Modeling with UML. Berkeley: Apress
- [3]. Sun Microsystems, Inc. The Java EE 5 Tutorial. Santa Clara, U.S.A
- [4]. Debu Panda, Reza Rahman, Derek Lane, (2007). EJB 3 in Action. United States of America: Manning Publications Co.
- [5]. Sitio oficial de JBoss Seam, <http://www.seamframework.org/Home>
- [6]. Dan Allen, (2009). Seam in Action. United States of America: Manning Publications Co.