



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS E INFORMÁTICA**

AUTOR: ORTIZ MUÑOZ GABRIEL ALEJANDRO

**TEMA: MANTENIMIENTO Y DESPLIEGUE DEL SISTEMA DE
PLANIFICACIÓN EN EL NIVEL DE GERENCIA SECTORIAL DE LA
UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE**

DIRECTOR: ING. MARIO, RON

CODIRECTOR: ING. ANDRÉS, DE LA TORRE

SANGOLQUÍ, DICIEMBRE 2013

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. ORTIZ MUÑOZ GABRIEL ALEJANDRO como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

Sangolquí, Diciembre del 2013

ING. MARIO RON.

ING. ANDRÉS DE LA TORRE.

DECLARACIÓN DE AUTORÍA DE RESPONSABILIDAD

Yo, Ortiz Muñoz Gabriel Alejandro, certifico que contribuyo directamente a la creación de este manuscrito, a la génesis y análisis de sus datos, por lo cual me encuentro en condiciones de hacer públicamente responsable de él, y acepto que mi nombre figure como su autor. Sangolquí, 11 de Diciembre de 2013.

ORTIZ MUÑOZ GABRIEL ALEJANDRO

AUTORIZACIÓN

Yo, Ortiz Muñoz Gabriel Alejandro con cédula de identidad No. 1717189235, autorizo a la biblioteca Alejandro Segovia de la Universidad de las Fuerzas Armadas (ESPE) se publique la Tesis “MANTENIMIENTO Y DESPLIEGUE DEL SISTEMA DE PLANIFICACIÓN EN EL NIVEL DE GERENCIA SECTORIAL DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE” realizada por mi persona.

Sangolquí, Diciembre de 2013

Ortiz Gabriel

1717189235

DEDICATORIA

Quiero dedicar este proyecto a mi madre, ya que gracias a su apoyo incondicional durante todos estos años, han hecho que me sea posible estar en el lugar que me encuentro hoy en día, sin ello no soy nada

Ortiz Muñoz Gabriel Alejandro

AGRADECIMIENTOS

A toda mi familia, en especial a mi madre quien fue la que con su amor y su apoyo me ayudó en todos los momentos, por estar siempre a mi lado y brindarme siempre su sabiduría y a ese ser especial que llegó a mi vida para llenarla de dulzura, amor y momentos espectaculares. Gracias.

Ortiz Muñoz Gabriel Alejandro

ÍNDICE

Capítulo 1.....	1
INTRODUCCIÓN.....	1
1.1 TEMA.....	1
1.2 ANTECEDENTES.....	1
1.3 PLANTEAMIENTO DEL PROBLEMA	2
1.4 JUSTIFICACIÓN.....	3
1.5 ALCANCE	4
1.6 OBJETIVO	5
1.6.1 OBJETIVO GENERAL.....	5
1.6.2 OBJETIVOS ESPECÍFICOS.....	6
1.6.3 METODOLOGÍA	6
1.7 HERRAMIENTAS	7
Capítulo 2.....	9
MARCO TEÓRICO DE REFERENCIA.....	9
2.1 SISTEMAS DE INFORMACIÓN.....	9
2.1.1 ENTRADA DE INFORMACIÓN	9
2.1.2 ALMACENAMIENTO DE INFORMACIÓN	9
2.1.3 PROCESAMIENTO DE INFORMACIÓN.....	10
2.1.4 SALIDA DE INFORMACIÓN.....	10

2.1.5	SISTEMA DE PROCESAMIENTO DE TRANSACCIONES	11
2.2	BALANCED SCORE CARD	11
2.2.1	¿QUÉ ES EL BALANCE SCORE CARD?	11
2.2.2	MISIÓN Y VISIÓN.....	14
2.2.3	PERSPECTIVAS BALANCE SCORE CARD	15
2.2.4	PLAN ESTRATÉGICO.....	20
2.2.5	GESTIÓN BSC A NIVEL CORPORATIVO.....	20
2.2.6	ADMINISTRACIÓN DEL RENDIMIENTO BSC.....	20
2.2.7	INTEGRACIÓN, BALANCEO Y ESTRATEGIA.....	22
2.3	INDICADORES.....	23
2.3.1	DEFINICIÓN	23
2.3.2	CARACTERÍSTICAS	23
2.4	SCRUM.....	25
2.4.1	DESARROLLO ÁGIL	26
2.4.2	CARACTERÍSTICAS	28
2.4.3	ETAPAS	30
2.4.4	CONCEPTO Y MÉTRICAS.....	32
2.4.5	DOCUMENTOS	34
2.5	INGENIERÍA INVERSA	35
2.5.1	TIPOS.....	36
2.5.2	HERRAMIENTAS.....	37
2.6	JAVA SERVER FACES (JSF)	39

2.6.1	CICLO DE VIDA.....	40
2.7	EJB	42
2.7.1	INTRODUCCIÓN	42
2.7.2	TIPOS.....	44
2.7.3	INVOCAR UN EJB.....	48
2.7.4	EJB Y LA PERSISTENCIA	49
2.8	TRANSACCIONES.....	50
2.8.1	CMT.....	50
2.8.2	BMT.....	50
2.9	MOTORES DE PERSISTENCIA.....	51
2.10	ARQUITECTURA.....	51
2.10.1	CAPA DE PRESENTACIÓN.....	51
2.10.2	CAPA DE NEGOCIO	54
2.10.3	CAPA DE PERSISTENCIA.....	55
2.10.4	BASE DE DATOS	55
2.11	SISTEMAS DE CONTROL DE VERSIONES	56
Capítulo 3.....		58
ANÁLISIS Y DISEÑO		58
3.1	ANÁLISIS DE REQUERIMIENTOS.....	58
3.2	REQUERIMIENTOS FUNCIONALES	59
3.2.1	MODELADO DE USUARIOS.....	59
3.3	PRODUCT BACKLOG.....	61

3.3.1	ANÁLISIS AMBIENTAL	61
3.3.2	DIRECCIONAMIENTO ESTRATÉGICO	69
3.3.3	ESTRUCTURA CORPORATIVA	81
3.3.4	PERFILES.....	82
3.3.5	USUARIOS	83
3.4	DIAGRAMAS DE CASOS DE USO.....	84
3.4.1	ANÁLISIS AMBIENTAL	84
3.4.2	DIRECCIONAMIENTO ESTRATÉGICO	87
3.4.3	PERFILES.....	92
3.4.4	ESTRUCTURA CORPORATIVA	92
3.4.5	USUARIOS	93
3.5	ALCANCE PROYECTO.....	93
3.5.1	ENTREGABLES DEL PROYECTO	94
3.5.2	RESTRICCIONES DEL PROYECTO	94
3.5.3	SUPUESTOS DEL PROYECTO.....	94
3.5.4	PROPÓSITO.....	94
3.6	ESTRUCTURA DE DESGLOSE DE TRABAJO	95
3.7	3. FASES DE PLANEACIÓN.....	96
3.8	INGENIERÍA INVERSA	98
3.8.1	MODELO CONCEPTUAL.....	98
3.8.2	MODELO LÓGICO.....	99
3.8.3	MODELO FÍSICO.....	100

3.9	DISEÑO DEL SISTEMA	101
3.9.1	DIAGRAMA INFRAESTRUCTURA	101
3.9.2	DIAGRAMA DE COMPONENTES.....	102
3.10	DIAGRAMA DE LA BASE DE DATOS.....	105
3.10.1	MODELO LÓGICO.....	105
3.11	DIAGRAMA DE CLASES	108
3.11.1	USUARIOS Y PERFILES	108
3.11.2	DIRECCIONAMIENTO ESTRATÉGICO	109
3.11.3	ANÁLISIS AMBIENTAL	110
Capítulo 4.....		111
DESARROLLO		111
4.1	BASE DE DATOS.....	111
4.1.1	ESTÁNDARES.....	111
4.1.2	NOMENCLATURA DBMS	112
4.1.3	BASE DE DATOS O SCHEMA.....	112
4.2	DISEÑO DE BASES DE DATOS	114
4.2.1	NORMALIZACIÓN	114
4.3	ESTÁNDARES DE PROGRAMACIÓN	115
4.3.1	NOMBRE DE LOS PAQUETES	116
4.3.2	CLASES	116
4.3.3	CONSTANTES.....	116
4.3.4	ESTÁNDARES DE FUNCIONES/MÉTODOS	117

4.4	PATRONES DE DISEÑO	117
4.4.1	PROXY	118
4.4.2	SINGLETON	118
4.4.3	SERVICE LOCATOR	118
4.4.4	DATA ACCESS OBJECT	119
4.5	DESCRIPCIÓN DE MÓDULOS	119
4.5.1	ANÁLISIS AMBIENTAL	119
4.5.2	DIRECCIONAMIENTO ESTRATÉGICO	122
4.5.3	USUARIOS	127
4.5.4	PERFILES	127
4.6	GESTIÓN DE LA CALIDAD	127
4.6.1	ANÁLISIS ARQUITECTURA	127
4.6.2	ANÁLISIS RENDIMIENTO	128
4.6.3	PRUEBAS DE UNIDAD	128
4.6.4	PRUEBA CAJA BLANCA	128
4.6.5	PRUEBA DE CAJA NEGRA	129
4.6.6	PRUEBAS DE INTEGRACIÓN	130
4.6.7	PRUEBAS DE ACEPTACIÓN	131
4.7	MANUAL USUARIO	132
4.8	MANUAL TÉCNICO	132
	Capítulo 5	133
	CONCLUSIONES Y RECOMENDACIONES	133

5.1	CONCLUSIONES	133
5.2	RECOMENDACIONES.....	134
	Bibliografía.....	135
	ANEXO A: MANUAL DE USUARIO	¡Error! Marcador no definido.
	ANEXO B: MANUAL TÉCNICO	¡Error! Marcador no definido.

ÍNDICE DE TABLAS

Tabla 3.1 Historia de usuario: Componentes	61
Tabla 3.2 Historia de usuario: Sub componentes	62
Tabla 3.3 Historia de usuario: Análisis interno	63
Tabla 3.4 Historia de usuario: Análisis externo	64
Tabla 3.5 Historia de usuario: Análisis extendido FODA	65
Tabla 3.6 Historia de usuario: Análisis síntesis FODA.....	66
Tabla 3.7 Historia de usuario: Expectativas	67
Tabla 3.8 Historia de usuario: Problemas	68
Tabla 3.9 Historia de usuario: Matriz FA	69
Tabla 3.10 Historia de usuario: Matriz FO.....	70
Tabla 3.11 Historia de usuario: Matriz DO.	71
Tabla 3.12 Historia de usuario: Matriz DA.....	72
Tabla 3.13 Historia de usuario: Soluciones.....	73
Tabla 3.14 Historia de usuario: Principios filosóficos.....	74
Tabla 3.15 Historia de usuario: Valores institucionales.	75
Tabla 3.16 Historia de usuario: Misión.	76
Tabla 3.17 Historia de usuario: Visión.....	77
Tabla 3.18 Historia de usuario: Estrategia general.	78
Tabla 3.19 Historia de usuario: Objetivos.	79
Tabla 3.20 Historia de usuario: Proyectos.	80
Tabla 3.21 Historia de usuario: Estructura corporativa.	81
Tabla 3.22 Historia de usuario: Perfiles.	82
Tabla 3.23 Historia de usuario: Usuarios	83
Tabla 3.24 Lista de entregables	94

ÍNDICE DE FIGURAS

Figura 2.1 Cuadro de mando integral.....	13
Figura 2.2 Plantilla de un indicador	25
Figura 2.3 Desarrollo tradicional vs desarrollo ágil	26
Figura 2.4 Etapas de Scrum.....	27
Figura 2.5 Ciclo de vida JSF	40
Figura 2.6 Arquitectura de un sistema informático basado en EJB	43
Figura 2.7 Ciclo de vida de un SLSB.....	45
Figura 2.8 Ciclo de vida de un SFSB	46
Figura 2.9 Ciclo de vida Singleton.....	47
Figura 2.10 Diagrama de flujo de un MDB	48
Figura 3.1 Caso de uso: Componentes.....	84
Figura 3.2 Caso de uso de subcomponentes.....	84
Figura 3.3 Caso de uso: Análisis interno.....	85
Figura 3.4 Caso de uso: Análisis externos.....	85
Figura 3.5 Caso de uso: Análisis FODA.....	86
Figura 3.6 Caso de uso: Expectativas.....	86
Figura 3.7 Caso de uso: Problemas	87
Figura 3.8 Caso de uso: Matriz FA.....	87
Figura 3.9 Caso de uso: Matriz FO	87
Figura 3.10 Caso de uso: Matriz DO	88
Figura 3.11 Caso de uso: Matriz DA	88
Figura 3.12 Caso de uso: Soluciones.....	88
Figura 3.13 Caso de uso: Principios filosóficos.....	89
Figura 3.14 Caso de uso: Valores institucionales	89

Figura 3.15 Caso de uso: Misión.....	90
Figura 3.16 Caso de uso: Visión	90
Figura 3.17 Caso de uso: Estructura organizacional	91
Figura 3.18 Caso de uso: Políticas generales.....	91
Figura 3.19 Caso de uso: Perfiles	92
Figura 3.20 Caso de uso: Estructura corporativa.....	92
Figura 3.21 Estructura corporativa	93
Figura 3.22 Estructura de Desglose del Trabajo.....	96
Figura 3.23 Planificación del proyecto	96
Figura 3.24 Diagrama de secuencia de la planificación del proyecto.....	97
Figura 3.25 Modelo conceptual antiguo sistema.....	98
Figura 3.26 Modelo lógico antiguo sistema.....	99
Figura 3.27 Diagrama físico antiguo sistema.....	100
Figura 3.28 Arquitectura de Hardware SISPLAGER.....	101
Figura 3.29 Arquitectura de Hardware SISPLAGER.....	102
Figura 3.30 Diagrama de componentes SISPLAGER	104
Figura 3.31 Modelo lógico usuarios y perfiles	105
Figura 3.32 Modelo lógico direccionamiento estratégico.....	106
Figura 3.33 Modelo lógico análisis ambiental.	107
Figura 3.34 Modelo físico usuarios y perfiles	108
Figura 3.35 Modelo físico direccionamiento estratégico.....	109
Figura 3.36 Modelo físico análisis ambiental	110
Figura 4.1 Login.....	¡Error! Marcador no definido.
Figura 4.2 Cambio de contraseña	¡Error! Marcador no definido.
Figura 4.3 Menú Inferior Usuario.....	¡Error! Marcador no definido.

Figura 4.4 Estándar de la cabecera tabla . ¡Error! Marcador no definido.

Figura 4.5 Estándar de búsqueda en las tablas ¡Error! Marcador no definido.

Figura 4.6 Estándar de búsqueda avanzada en las tablas ¡Error! Marcador no definido.

Figura 4.7 Estándar ¡Error! Marcador no definido.

Figura 4.8 Menú del sistema SISPLAGER ¡Error! Marcador no definido.

Figura 4.9 Cabecera tabla unidad organizacional ¡Error! Marcador no definido.

Figura 4.10 Nueva estructura organizacional ¡Error! Marcador no definido.

Figura 4.11 Nuevo usuario ¡Error! Marcador no definido.

Figura 4.12 Nuevo perfil ¡Error! Marcador no definido.

Figura 4.13 Tabla de permisos a las pantallas ¡Error! Marcador no definido.

Figura 4.14 Menú estructura corporativa .. ¡Error! Marcador no definido.

Figura 4.15 Cuadro de dialogo instalación Java ¡Error! Marcador no definido.

Figura 4.16 Java Setup ¡Error! Marcador no definido.

Figura 4.17 Java instalado correctamente ¡Error! Marcador no definido.

Figura 4.18 Cuadro de bienvenida de MySQL ¡Error! Marcador no definido.

Figura 4.19 Términos y condiciones de la licencia de MySQL ¡Error! Marcador no definido.

Figura 4.20 Opciones de instalación de MySQL; **Error! Marcador no definido.**

Figura 4.21 Cuadro de dialogo para instalar MySQL; **Error! Marcador no definido.**

Figura 4.22 Instalación terminando MySQL; **Error! Marcador no definido.**

Figura 4.23 Configuración de MySQL; **Error! Marcador no definido.**

Figura 4.24 Opciones de configuración MySQL server; **Error! Marcador no definido.**

Figura 4.25 Opciones de configuración MySQL Instance.....; **Error! Marcador no definido.**

Figura 4.26 Opciones de configuración MySQL DATABASE; **Error! Marcador no definido.**

Figura 4.27 Opciones de configuración MySQL en el disco; **Error! Marcador no definido.**

Figura 4.28 Opciones de configuración usuarios MySQL.....; **Error! Marcador no definido.**

Figura 4.29 Opciones de configuración del puerto MySQL; **Error! Marcador no definido.**

Figura 4.30 Opciones de configuración Character set MySQL.....; **Error! Marcador no definido.**

Figura 4.31 Opciones de configuración MySQL; **Error! Marcador no definido.**

Figura 4.32 Creación contraseña; **Error! Marcador no definido.**

Figura 4.33 Menú a ejecutar para configurar MySQL; **Error! Marcador no definido.**

Figura 4.34 Opciones de instalación MySQL WORKBENCH.....; **Error! Marcador no definido.**

Figura 4.35 Opciones de instalación MySQL WORKBENCH.....; **Error! Marcador no definido.**

Figura 4.36 Instalar MySQL WORKBENCH; **Error! Marcador no definido.**

Figura 4.37 Proceso de instalación MySQL WORKBENCH.....; **Error! Marcador no definido.**

Figura 4.38 Finalización de la instalación .; **Error! Marcador no definido.**

Figura 4.39 Conexión BD de MySQL; **Error! Marcador no definido.**

Figura 4.40 Ventana para agregar Script BD; **Error! Marcador no definido.**

Figura 4.41 Cuadro de dialogo para seleccionar Script BD; **Error! Marcador no definido.**

Figura 4.42 Cuadro de dialogo para ejecutar Script BD; **Error! Marcador no definido.**

Figura 4.43 Configuración Datasource Jboss; **Error! Marcador no definido.**

Figura 4.44 inicio del Jboss; **Error! Marcador no definido.**

RESUMEN

El presente proyecto de tesis muestra el fundamento teórico de la planificación gerencial, llevado a la práctica en un sistema informático desarrollado en una arquitectura JEE 6, utilizando la metodología Scrum, con MySQL como DBMS y desplegado en un servidor Jboss 6.

En el transcurso del proyecto se aplica los diversos conocimientos adquiridos durante la Carrera de Ingeniería en Sistemas e Informática. La cuál busca desarrollar una solución para automatizar los controles gerenciales que se realizan en la Universidad de las Fuerzas Armadas ESPE. En el Capítulo 1 se detalla la formulación del problema, el Capítulo 2 aspectos teóricos de la metodología, procesos de planificación, JEE6, en el Capítulo 3 se realiza un análisis y planificación del desarrollo del software, posteriormente el Capítulo 4 cubre en sí el desarrollo de la aplicación utilizando la metodología Scrum. Finalizando en el Capítulo 5 con las conclusiones y recomendaciones. Para finalmente una vez implantado el producto tener una herramienta de planificación estratégica.

PALABRAS CLAVE

- 1. SCRUM**
- 2. JEE6**
- 3. BALANCE SCORD CARD**
- 4. PLANIFICACIÓN ESTRATEGICA**
- 5. JBOSS 6**

ABSTRACT

The present thesis project shows the theoretical foundation of management planning, implemented in a computer system developed with JEE 6 architecture, using the Scrum methodology with MySQL as DBMS and deployed within a Jboss 6 server.

During the project the various of the knowledge acquired during the Career of Systems Engineering and Informatics were applied. Who seeks the development of a solution to automate the management controls process that are performed at the The Army University ESPE. The Chapter 1 detailed problem formulation, Chapter 2 explained the theoretical aspects of methodology, planning process and JEE6 architecture, Chapter 3 makes an analysis and planning of the software development process, then Chapter 4 itself covers the application development using the Scrum methodology. ending in Chapter 5 with conclusions and recommendations. To finally, after being implemented have an efficient product for strategic planning.

KEY WORDS

- 1. SCRUM**
- 2. JEE6**
- 3. BALANCE SCORD CARD**
- 4. STRATEGIC PLANNING**
- 5. JBOSS 6**

CAPÍTULO 1

INTRODUCCIÓN

1.1 TEMA

Desarrollo de un Sistema de Planificación en el Nivel de Gerencia Sectorial de la Universidad de las Fuerzas Armadas ESPE, aplicando Scrum y JEE6.

1.2 ANTECEDENTES

Hoy en día se encuentran en una época en la que la información se genera cada segundo de forma instantánea en todas las organizaciones y en cada uno de sus niveles, en el ámbito empresarial, tener a la mano la información necesaria puede significar una ganancia o una pérdida monetaria, a través de las últimas décadas han aparecido y evolucionado los sistemas de planeación de los recursos empresariales para ayudar en este sector mejor conocidos como ERP , son un tipo de software que permite a las empresas controlar la información que se genera en cada departamento y cada nivel de la misma.

El éxito en la implantación de tecnología ERP es responsabilidad de toda la empresa. Es indispensable fomentar el trabajo en equipo debido a que a partir de la liberación del ERP, los datos ingresados por un usuario serán utilizados por otro de un área completamente diferente y quien finalmente "alimenta y da fuerza" a cualquier sistema es el personal que lo usa. Todos

los retos y costos intrínsecos a los ERP fuerzan a las empresas a realizar un cambio de cultura.

En la actualidad existe una aplicación que fue desarrollada por el Ingeniero Pablo Álvarez quien realizó el análisis, diseño e implementación de un sistema basado en componentes para automatizar la herramienta Balanced Score Card en el área administrativa de Teleamazonas.

Este sistema fue probado en el CTT ESPE-CECAI como un producto de comercialización y de consultoría, se añadió al producto inicial un reporte del cuadro de mando integral.

La UDI se encuentra desarrollando un repositorio de información estadística para las unidades operativas de la Escuela, con el objeto de cumplir los requerimientos de los organismos de control de la Educación superior y de planificación del país.

1.3 PLANTEAMIENTO DEL PROBLEMA

Hasta el momento los controles gerenciales se realizan en la Universidad de las Fuerzas Armadas ESPE mediante informes personalizados de las diferentes dependencias de ésta, se ha tratado de suplir un sistema gerencial de planificación mediante repositorios de información que no permiten una fluida comunicación y control de los aspectos referentes a la ejecución de los objetivos institucionales.

También es necesario cumplir con los requerimientos de los organismos que rigen la educación superior y enfrentar la acreditación, después de automatizar esta herramienta se puede tener un mejor control de las cuatro perspectivas que maneja ésta y que son: la forma de ingresar al mercado, la expansión y crecimiento, la manera de llevar las finanzas, y su trato para con sus clientes, mediante objetivos que comprometen a cada empleado y cuantificados en índices no financieros para medir el cumplimiento de los mismos.

Además la construcción de éste sistema computacional en base en la herramienta Balanced Score Card ayudará a la UDI, a obtener la información precisa y en el momento oportuno aprovechando los beneficios de la tecnología mediante el software y que además se podrá visualizar desde cualquier parte del mundo, puesto que será orientado a la Web.

1.4 JUSTIFICACIÓN

La elaboración de un sistema que automatice la herramienta Balanced Score Card garantizar a las organizaciones la eficaz ejecución de sus estrategias comprometiéndolo a todo su personal a un mejoramiento continuo de todos los procesos y especialmente puntos críticos que tengan o no problemas en cuanto a su funcionar diario. Es decir es una herramienta de administración de empresas que muestra continuamente cuándo una compañía y sus empleados alcanzan los resultados definidos por el plan estratégico. También es una herramienta que ayuda a la compañía a expresar los objetivos e iniciativas necesarias para cumplir con la estrategia.

Uno de los motivos principales que se escoge esta herramienta es que pretende identificar las relaciones causa-efecto que provocan los resultados obtenidos. Por tanto, se trata de obtener información relevante sobre los principales factores que pueden llevar a la consecución de los objetivos de las empresas. Tomando en cuenta estas consideraciones se plantean una solución basada en tecnologías de información por lo cual se desarrolla el sistema mediante una plataforma de libre distribución, para obtener un producto que permita facilitar la gestión y optimizar tanto el tiempo como los procesos realizados por la UDI.

1.5 ALCANCE

El proyecto de tesis cubre las tareas de mantenimiento del sistema construido anteriormente, en donde se realizará la revisión del mismo por los potenciales usuarios, para pasar a la fase de despliegue, en la que se determinarán los requerimientos finales de mantenimiento para construir la nueva versión mejorada y adaptada de sistema, hasta el despliegue de la misma en el nivel de gerencia sectorial, es decir: Vicerrectorados y Gerencia Administrativa, contando con la colaboración del personal de la UDI.

Se realizará el análisis, diseño e implementación de las cuatro perspectivas que maneja el Balanced Score Card, y que son:

- La perspectiva de los clientes.
- La perspectiva de los accionistas.

- La perspectiva de la innovación y crecimiento.
- La perspectiva de los procesos internos.

Dentro de las cuatro perspectivas se podrá realizar el ingreso, actualización y eliminación de los índices financieros y no financieros, de los objetivos, de las estrategias, y de los proyectos y sus responsables.

Además se podrá realizar reportes que permitan tener un mayor control sobre el cumplimiento de dichos índices, los cuales van a permitir al administrador tomar decisiones oportunas.

En la implementación del sistema hay que tomar en cuenta que se realizará con datos reales de la institución, y de no ser posible esto se realizará con datos de prueba.

1.6 OBJETIVO

1.6.1 OBJETIVO GENERAL

Realizar el mantenimiento, análisis, diseño y construcción de un nuevo sistema computacional que permita automatizar la herramienta de gestión Balanced Score Card en la UDI, utilizando la metodología Scrum y la Arquitectura JEE6, impulsando a ésta a su crecimiento y a una mayor competitividad.

1.6.2 OBJETIVOS ESPECÍFICOS

- Investigar los principios en los que se basa el Balanced Score Card como herramienta de gestión y administración de empresas.
- Investigar y analizar sistemas computacionales que permitan automatizar la herramienta Balanced Score Card existentes en el mercado.
- Realizar los nuevos requerimientos de los usuarios potenciales para la nueva versión del sistema.
- Dar mantenimiento al antiguo sistema y adaptarlo a los nuevos requerimientos.
- Implementar el sistema computacional para la UDI, con la información que proporcione ésta, o con datos de prueba para comprobar el correcto funcionamiento del software.

1.6.3 METODOLOGÍA

La metodología que se empleará es Scrum ya que es una metodología ágil ya que permite realizar entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Se ha escogido Scrum debido que el cliente se compromete con el proyecto dado que lo ve crecer iteración a iteración. Asimismo le permite en cualquier momento realinear el software con los objetivos de negocio de su empresa, ya que puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración.

1.7 HERRAMIENTAS

Las herramientas que se emplea para la automatización del Balanced Score Card o Cuadro de Mando Integral son:

Como plataforma de desarrollo JEE6, para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. Como IDE de desarrollo se utiliza Eclipse Índigo puesto que soporta Java 6, que son herramientas gráficas que permiten ahorrar tiempo en el desarrollo de aplicaciones, y que además son gratuitas.

Como herramienta de modelamiento se usa Power Designer, puesto que una de las herramientas más potentes y con más experiencia para este tipo de aplicaciones.

Como herramienta para el diseño del modelo lógico y físico de la base de datos se usa Power Designer, por sus características y fácil manejo.

Como motor de base de datos se usa MySQL, que es un motor suficientemente potente para la aplicación que se va a realizar.

Como servidor de aplicaciones se utiliza Jboss 6 que es un servidor de aplicaciones JEE de código abierto implementado en Java.

CAPÍTULO 2

MARCO TEÓRICO DE REFERENCIA

2.1 SISTEMAS DE INFORMACIÓN

Es un conjunto de elementos orientados al tratamiento y administración de datos e información, interrelacionados con el propósito de prestar atención a las demandas de información de una organización, generados para cubrir una necesidad u objetivo lo que permite elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones. Dichos elementos formarán parte de alguna de las siguientes categorías:

2.1.1 ENTRADA DE INFORMACIÓN

Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas.

2.1.2 ALMACENAMIENTO DE INFORMACIÓN

Es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la

información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos.

2.1.3 PROCESAMIENTO DE INFORMACIÓN

Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base

2.1.4 SALIDA DE INFORMACIÓN

Es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo. En este caso, también existe una interface automática de salida.

2.1.5 SISTEMA DE PROCESAMIENTO DE TRANSACCIONES

Gestiona la información referente a las transacciones producidas en una empresa u organización, también se le conoce como Sistema de Información operativa.

a) Sistemas de información gerencial (MIS)

Orientados a solucionar problemas empresariales en general.

b) Sistemas de soporte a decisiones (DSS)

Herramienta para realizar el análisis de las diferentes variables de negocio con la finalidad de apoyar el proceso de toma de decisiones.

c) Sistemas de información ejecutiva (EIS)

Herramienta orientada a usuarios de nivel gerencial, que permite monitorizar el estado de las variables de un área o unidad de la empresa a partir de información interna y externa a la misma. Es en este nivel cuando los sistemas de información manejan información estratégica para las empresas.

2.2 BALANCED SCORE CARD

2.2.1 ¿QUÉ ES EL BALANCE SCORE CARD?

BSC (Balance Score Card) es una sigla que se traduce al español como “Indicadores Balanceados de Desempeño”. Esta metodología deriva de la

gestión estratégica de empresas y presupone una elección de indicadores que no debe ser restringida al área económico-financiera. Así como no es posible comandar un avión controlando apenas la velocidad, los indicadores financieros no son suficientes para garantizar que una empresa se dirija en la dirección correcta. Por estos motivos, será necesario monitorear, junto a los indicadores económicos-financieros, el desempeño de mercado, los procesos internos, la innovación y la tecnología.

El Balance Score Card ofrece una visión integrada y balanceada de la empresa y permite desarrollar la estrategia en forma clara. Esto se logra a través de objetivos estratégicos identificados en 4 perspectivas: financiera, clientes, procesos internos y aprendizaje e innovación como muestra la Figura 2.1. Cada una de las perspectivas se vincula con las demás mediante relaciones de causa y efecto. BSC promueve, además, el alineamiento de los objetivos estratégicos con indicadores de desempeño, metas y planes de acción para hacer posible la generación de estrategias en forma integrada y garantizar que los esfuerzos de la organización se encuentren en línea con las mismas.

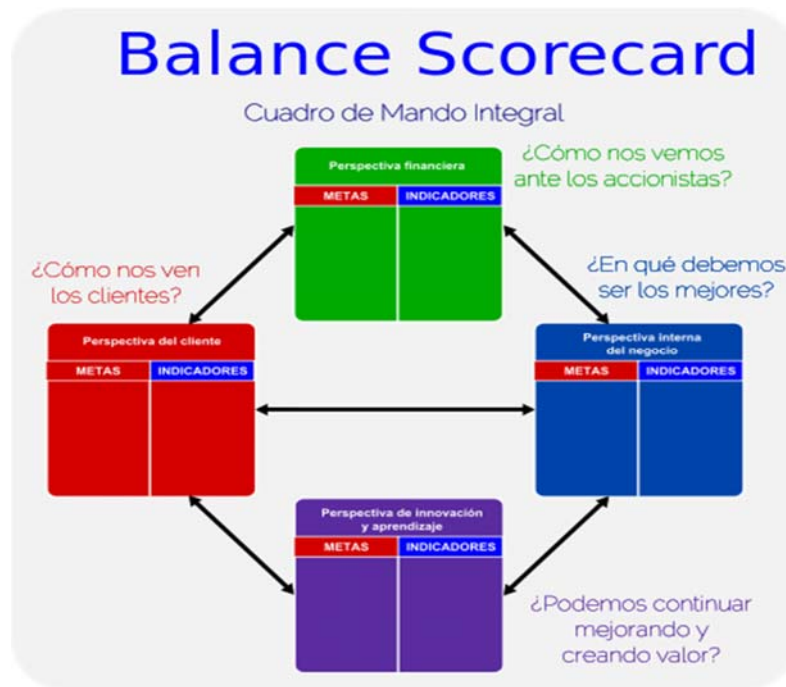


Figura 2.1 Cuadro de mando integral Recuperado

http://es.wikipedia.org/wiki/Cuadro_de_mando_integral

Implementación del BSC en todo el Organismo proporcionará:

- Una metodología común y un marco coordinado para todos esfuerzos interinstitucionales de medición del desempeño.
- Un "lenguaje común" para los administradores de la agencia.
- Una base común para la comprensión de los resultados de medición.
- Una visión integrada de la agencia global.

El BSC también proporciona datos fiables en los cuales se pueden basar para decisiones futuras, a partir de la asignación de los recursos disponibles de la dirección del futuro.

Esto permitirá a la organización gestionar sus actividades y sus recursos con mayor eficacia. Por ejemplo, el BSC puede formar una base común para apoyar un caso de negocio para obtener más recursos.

2.2.2 MISIÓN Y VISIÓN

a) Misión

Es una definición comprensible, clara y concisa del propósito organizacional o identidad. Señala la dirección que las acciones de la organización deben seguir. Revela la imagen que la organización desea proyectar, refleja el concepto que tiene la organización de sí misma, indica los principales productos o servicios que provee, e identifica los principales grupos de interés que la organización busca satisfacer. Refleja, en síntesis, la razón de existir de la organización, y debe reflejar los mandatos del organismo, identificar las principales necesidades y problemas que la organización debería manejar, ser realista, definir la población objetivo, y reconocer las expectativas de los principales grupos de interés.

b) Visión

Es la representación de lo que debe ser en el futuro la organización. Contesta a preguntas relacionadas a cuál es la situación futura deseada de

los usuarios y beneficiarios, cual es la posición futura de la organización en relación a las otras, cuales son las contribuciones distintivas que quieren hacerse en el futuro y/o cuales son los principales proyectos o actividades que se desea desarrollar. La Visión es la imagen futura que una organización desarrolla sobre sí misma y sobre la realidad sobre la cual trabaja. Por lo general la visión incluye tanto los cambios que se desea lograr de la población de objetivo, como la imagen objetivo de la propia institución.

2.2.3 PERSPECTIVAS BALANCE SCORE CARD

a) Perspectiva del cliente

Para esta perspectiva, el "cliente" significa cómo ve a la organización, y qué debe hacer esta para mantenerlo como cliente. Esto incluye a los clientes internos y directos. El grado de satisfacción está asociado directamente con la puntualidad en la entrega de productos o servicios y otros factores que afectan el programa de adquisición. La puntualidad puede incluir una evaluación de lo siguiente:

Porcentaje de clientes satisfechos con la calidad: Esta es la satisfacción del cliente con la calidad de los bienes y servicios entregados. "Calidad", también incluye una evaluación de si los contratistas seleccionados para los premios ofrecen la mejor combinación de calidad y precio. Los datos de esta medida provienen de una encuesta a los clientes.

Porcentaje de clientes satisfechos con la capacidad de respuesta: la cooperación y habilidades de comunicación de la oficina de la adquisición. Las

percepciones, las opciones, y el comportamiento de todos los participantes en el proceso de adquisición afecta al resultado de cualquier adquisición. Este elemento se basa en el grado de capacidad de respuesta del equipo de la adquisición, el éxito de los mecanismos que apoyan la generación de equipos, y el grado de satisfacción con las comunicaciones y la resolución de problemas. Los datos correspondientes a esta medida provienen de una encuesta a los clientes.

b) Perspectiva Financiera

Relación costo. Este elemento representa el costo para cada cargo a gastar un dólar de los fondos de sus clientes. Se reconoce que estos elementos de costo que no puede captar la totalidad del costo de la adquisición, pero se tomó la decisión de no tratar de cuantificar el costos de desarrollo de declaraciones de trabajo, las inspecciones que realizan, haciendo pagos, etc.

Una reducción de costos mediante el uso de tarjetas de compra. Este elemento representa el número de transacciones con tarjetas de compra multiplicado por el estimado los costos evitados por el uso de tarjetas de compra en comparación con la emisión de una orden de compra.

Porcentaje de interés del total de dinero desembolsado. Este elemento representa la cantidad de intereses que se pagan las sanciones como un porcentaje del total de desembolsos por parte del organismo. Este elemento se calcula tomando las penas de interés total pagado por cada oficina y

dividiendo por la cantidad de los desembolsos totales pagados. Los datos para esta medida se pueden extraer de la anualidad del informe de pago del sistema o de los sistemas financieros locales

c) Perspectiva de Procesos Internos de Negocio

Relación de las protestas sufridas por el General Accounting Office (GAO) o el Tribunal de Reclamaciones Federales (COFC). Este elemento mide la relación de protestas confirmada por la GAO o COFC. Para esta medida, una protesta define como una objeción escrita por un proveedor(s) en relación con una acción de adquisición. Esta medida se calcula dividiendo el número de protestas confirmadas por el número total de adjudicaciones de contratos nuevos. GAO datos para este medida se puede extraer de la competencia anual de la GAO en la Contratación Acto informe al Congreso, y los datos COFC se puede extraer de locales archivos de control de protesta.

Porcentaje de las acciones que utilizan el comercio electrónico. Este elemento representa el número total de acciones de adquisición a través del uso del comercio electrónico.

Porcentaje de cumplimiento de los objetivos socioeconómicos. Este elemento da seguimiento a cada logro de la organización de los objetivos socio-económicos establecidos. Este elemento estará integrado por una serie de medidas por separado. Para cada categoría definida, los logros de la organización para cada categoría, según lo informado a la Administración de pequeños Negocios, se divide por la meta establecida para esa categoría. Las

medidas individuales de las categorías no son promedian. Los datos para esta medida se pueden extraer de los archivos de la Oficina local de Utilización de Negocios Pequeños y en Desventaja.

Porcentaje de contrataciones competitivas de un total de contrataciones. Este elemento asume que el ahorro de costo, mayor calidad o un mejor abastecimiento en general logran mediante el uso de la competencia frente a otra competencia. Este elemento da seguimiento al porcentaje de adquisiciones competitivas como porcentaje de las compras totales. Dos elementos de datos se realizarán un seguimiento de esta medida. El primero es el número total de acciones competitivas dividido por el número total de acciones. El segundo elemento es el número total de acciones competitivas más el número de seguimiento de las acciones, dividido por el número total de acciones menos el número de acciones no disponible para la competencia.

d) Aprendizaje y perspectiva de crecimiento

Alcance de la gestión de la información confiable. Esta medida recoge el grado en que los directivos creen que tienen información oportuna, precisa y completa para tomar decisiones de gestión.

Ciento de los empleados cumplen con los estándares obligatorios de cualificación. Esta medida identifica el porcentaje de empleados que cumplen con la adquisición de requisitos de educación obligatoria, la formación y la experiencia como se identifica en las Normas de Calificación de Especialistas del contrato. Se calcula dividiendo el número de empleados de adquisición

que cumplan con la educación, de formación, requisitos y experiencia por el número total de adquisición empleados de la organización. Los datos se derivan de la adquisición local Desarrollo de Carrera de datos del sistema.

Porcentaje de empleados satisfechos con el ambiente de trabajo. Para retener profesionales de alta calidad de adquisición y mejorar el rendimiento de los trabajadores, el ambiente de trabajo debe ser agradable e incluir las medidas necesarias para la realización del trabajo. Esta medida representa el grado de satisfacción de los empleados con elementos tales como herramientas que se proporcionan (por ejemplo, tecnología de la información, material de referencia, etc.), las condiciones de trabajo, y los mecanismos de recompensa. Los datos para esta medida provienen de una encuesta.

Porcentaje de empleados satisfechos con el profesionalismo, la cultura, los valores, y el empoderamiento. La dirección juega un papel vital en el funcionamiento de cada equipo de adquisición por parte de la dirección, motivación y que conduce a su personal.

Adquisición de liderazgo deben fomentar un ambiente profesional que promueva la adquisición eficiente y eficaz de bienes y servicios de contratistas responsables. Esta medida incluye una evaluación de las percepciones de los empleados, cultura, valores, y el empoderamiento. Los datos de esta medida provienen de una encuesta a los empleados.

2.2.4 PLAN ESTRATÉGICO

El plan estratégico, además de reunir los objetivos estratégicos, las metas e iniciativas, brinda a la organización un panorama general, conjugando las prioridades gubernamentales en el período presupuestal. Contiene una guía tanto para la toma de decisiones de largo como de corto plazo.

2.2.5 GESTIÓN BSC A NIVEL CORPORATIVO

Utilizando BSC requiere que la empresa, y cada uno de sus departamentos (incluyendo TI), puedan convertirse organizativamente en este nuevo marco. Esto significa que el proceso de rendimiento de mejora, medición y de gestión deberá estar íntimamente entendido.

2.2.6 ADMINISTRACIÓN DEL RENDIMIENTO BSC

Varios pasos deben llevarse a cabo para establecer medidas de desempeño que tienen sentido y son viables en toda la organización.

Paso 1: Definición de visión de la organización, misión y estrategia

El Balance Score Card (BSC) requiere la creación de una visión, misión y estrategia de la organización. Esto asegura que las medidas de desempeño desarrollados en cada una de las perspectiva de apoyo deben cumplir con los objetivos estratégicos de la organización. También ayudan a los empleados a visualizar y comprender los vínculos entre las medidas de desempeño y la realización exitosa de los objetivos estratégicos.

La clave está en identificar en primer lugar a donde uno quiere que la organización sea en un futuro próximo, a continuación, establecer una visión que parece algo fuera de su alcance.

Paso 2: Desarrollar objetivos de rendimiento, medidas y metas puesto que es esencial para identificar lo que la organización debe hacer bien (es decir, los objetivos de desempeño) para alcanzar la visión. Para cada objetivo que debe ser realizado bien, es necesario identificar las medidas y establecer metas que abarcan un período de tiempo razonable (por ejemplo, de tres a cinco años).

Aunque esto suena simple, muchas variables impacte en el tiempo este ejercicio tendrá. La primera, y más importante, variable es cómo muchas personas se emplean en la organización y el grado en que estarán involucrados en el establecimiento de la visión, misión, las medidas y objetivos.

El BSC traduce la visión de una organización en un conjunto de prestaciones de objetivos distribuidos en cuatro perspectivas: financiera, clientes, internos los procesos de negocio, y el aprendizaje y el crecimiento. Otros objetivos se mantienen para medir los conductores a largo plazo de éxito. Mediante el uso del BSC, una organización controla tanto su rendimiento actual (financiera, la satisfacción del cliente, y el negocio proceso de los resultados) y sus esfuerzos para mejorar los procesos, motivar y educar a los empleados, y mejorar los sistemas de información - su capacidad para aprender y mejorar.

Paso 3: Ajustes pueden ser necesarios

Finalmente, se necesita tiempo para establecer medidas, pero también es importante para reconocer que no puede ser perfecto la primera vez.

2.2.7 INTEGRACIÓN, BALANCEO Y ESTRATEGIA

BSC es un modelo integrado porque utiliza las 4 perspectivas indispensables para ver una empresa o área de la empresa como un todo, luego de dos investigaciones de 1 año de duración: una en los Estados Unidos en 1990 y la otra en Europa en 1996, se ha podido establecer que son estas las 4 perspectivas básicas con las cuales es posible lograr cumplir la visión de una compañía y hacerlo exitosamente.

Es balanceado porque busca el balance entre indicadores financieros y no financieros, el corto plazo y el largo plazo, los indicadores de resultados y los de proceso y un balance entre el entorno y el interior de la firma, ese es el concepto clave y novedoso sobre el cual se basa el nombre "Balance Score Card": Sistema de indicadores balanceados. Lo importante aquí es que los indicadores de gestión de una compañía estén balanceados, es decir existan tanto indicadores financieros como no financieros, de resultado como de proceso y así sucesivamente.

Es una herramienta estratégica porque se trata de tener indicadores que están relacionados entre sí y que cuenten la estrategia de la compañía por medio de un mapa de enlaces causa-efecto (indicadores de resultado e

indicadores impulsores). La mayoría de empresas tienen indicadores aislados, definidos independientemente por cada área de la compañía, los cuales buscan siempre fortalecer el poder de las mismas, fortaleciendo cada vez más las islas o compartimientos funcionales.

Lo que requieren hoy en día las empresas son indicadores relacionados (cruzados) contruidos entre todas las áreas en forma consensuada, buscando siempre negociar los trade-offs no permitiendo que un área sobresalga a costa de otra u otras áreas de la empresa.

2.3 INDICADORES

2.3.1 DEFINICIÓN

Es una variable cuantitativa cuya finalidad es brindar información acerca del grado de cumplimiento de una meta de gestión. Al medir el progreso hacia el logro de estas metas, funciona como una “señal de alerta” que muestra si se está trabajando en la senda correcta de acuerdo con los resultados planificados. En este sentido, sirve para detectar posibles desvíos y corregirlos.

2.3.2 CARACTERÍSTICAS

Los indicadores para cumplir con su rol de control de gestión estratégico y operativo, deberían cumplir con las siguientes características:

- Relevancia: deben aportar información imprescindible para la toma de decisiones, contribuyendo directamente al cumplimiento de los objetivos estratégicos.
- Objetividad: los datos deben ser auditables, es decir, debe existir la posibilidad de ser verificados independientemente de forma externa.
- Ser inequívocos: su análisis no debería permitir interpretaciones ambiguas o contrapuestas.
- Completitud: idealmente un buen conjunto de indicadores debería capturar todas las posibles conductas o acciones relevantes vinculadas al suceso que se intenta captar.
- Sensibilidad: deben ser capaces de reflejar las acciones y esfuerzos del individuo o la unidad que realiza la gestión, es decir, deben basarse sobre variables que la unidad pueda controlar.
- Accesibilidad: su obtención debe tener un costo razonable en relación a los beneficios que reporta la información que brindan.

A continuación la Figura 2.2 muestra una plantilla de un indicador.

a. Información General	
<i>Número /nombre indicador:</i>	<i>Responsable:</i>
<i>Lineamiento estratégico:</i>	<i>Meta:</i>
<i>Objetivo Estratégico:</i>	
<i>Descripción del indicador:</i>	
b. Característica del Indicador	
Clasificación: Impacto, Eficiencia, Eficacia, Calidad	
<i>Frecuencia:</i>	<i>Unidad de medida:</i>
c- Cálculo y especificaciones de datos	
<i>Fórmula:</i>	
<i>Fuente de datos y forma de obtención:</i>	
d. Información sobre la actividad	
<i>Línea de base (último dato relevado):</i>	
<i>Desvíos admitidos:</i>	

Figura 2.2 Plantilla de un indicador

2.4 SCRUM

Es una metodología ágil que aplica de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados

pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum no es un proceso o una técnica para construir productos; es un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas.

2.4.1 DESARROLLO ÁGIL

En la Figura 2.3 se puede ver las principales diferencias entre el campo de Scrum y el modelo clásico de desarrollo.



Figura 2.3 Desarrollo tradicional vs desarrollo ágil Recuperado http://www.scrummanager.net/files/sm_proyecto.pdf

No lo desarrollan equipos diferentes especialistas en distintas áreas, al contrario hay un sólo equipo, formado por personas muy competentes, con perfiles y conocimientos que cubren las disciplinas necesarias para construir el producto.

No hay fases. Éstas pasan a ser tareas que se ejecutan cuando se necesitan. No se hace primero el diseño del concepto o los requisitos, más tarde el análisis, luego el desarrollo, etc.



Figura 2.4 Etapas de Scrum Recuperado
<http://www.epidataconsulting.com/tikiwiki/tiki-index.php?page=Scrum>

Lo que en el software serían las fases de requisitos del sistema, requisitos del software, análisis, diseño, construcción, pruebas e integración, y se ejecutarían de forma secuencial, pasan a ser tareas que se llevan a cabo cada vez que hacen falta. Normalmente a lo largo de pequeñas iteraciones durante todo el desarrollo.

No se espera a desarrollar requisitos detallados antes de empezar el análisis o el desarrollo. Muchas veces éstos no se pueden conocer si no se avanza en el desarrollo, y se va viendo y “tocando” el resultado.

2.4.2 CARACTERÍSTICAS

Las características comunes que se identifican en los entornos de desarrollo son.

a) Incertidumbre

En las empresas, desde la dirección se apunta cuál es la visión genérica que se quiere conseguir, o la dirección estratégica que hay que seguir, pero no un plan detallado del producto y su desarrollo.

b) Auto-organización.

Son equipos auto-organizados. No hay roles de gestión que marquen pautas o asignación de tareas. No se trata de equipos auto-dirigidos, sino auto-organizados. La gestión marca la dirección, pero no la organización. Parten de cero. Deben empezar por crear su propia organización y buscar el conocimiento que necesitan. Son similares a una pequeña empresa “Start-up” en la que todos los integrantes trabajan de forma conjunta y auto-organizada, sin unos patrones organizativos impuestos por una estructura empresarial ajena al grupo.

La dirección de la empresa actúa como un inversor de capital riesgo que aporta los recursos necesarios para que trabajen en su proyecto.

c) Fases de desarrollo solapadas.

En el desarrollo ágil las “fases” pasan a ser “actividades”. El concepto de fase implica sucesión secuencial de unas a otras. En un campo de Scrum los trabajos que se llevan a cabo pierden el carácter de fase y son actividades que se realizan en cualquier momento, de forma simultánea, o a demanda según las necesidades en cada iteración. Por ejemplo, lo que para el desarrollo en cascada es una “modificación de requisitos”, en un modelo ágil es información que enriquece o concreta la visión del producto. El mismo término “modificación” tiene implícito el concepto de que se está “cambiando” algo que ya se había definido.

d) Control sutil

El equipo trabaja con autonomía en un entorno de ambigüedad, inestabilidad y tensión. La gestión establece puntos de control suficientes para evitar que el ambiente de ambigüedad, inestabilidad y tensión del “campo de Scrum” derive hacia descontrol. Pero la gestión no ejerce un control rígido que impediría la creatividad y la espontaneidad. El término “control sutil” se refiere a generar el ecosistema adecuado para un “auto-control entre iguales,” consecuencia de la responsabilidad y del gusto por el trabajo que se realiza.

e) Difusión del conocimiento

Tanto a nivel de proyecto como de organización. Los equipos son multidisciplinares; todos los miembros aportan y aprenden tanto del resto del equipo como de las investigaciones, innovaciones de su producto y de la experiencia del desarrollo. Las personas que participan en un proyecto, con el tiempo van cambiando de equipo en la organización, a otros proyectos; de esta forma se van compartiendo y comunicando las experiencias en la organización.

Los equipos y las empresas mantienen libre acceso a la información, herramientas y políticas de gestión del conocimiento

2.4.3 ETAPAS

El patrón de ciclo de vida de un modelo de desarrollo ágil está compuesto de cinco etapas:

a) Concepto

Se crea la visión del producto o servicio que quiere obtener. Se decide y selecciona al equipo de personas que lo llevará a cabo. Partir sin una visión determinada produce esfuerzo baldío. Igual que para una empresa, la visión es un factor crítico para el éxito del proyecto. Se necesita tener la visión de lo que se quiere, y conocer el alcance del proyecto. Esta información la deben compartir todos los integrantes del equipo.

b) Especulación

Una vez que se dispone de la visión de lo que se quiere conseguir, el equipo especula y construye hipótesis sobre la información de la visión. En esta etapa se determinan las limitaciones impuestas por el entorno de negocio (costes y agendas principalmente) y se determina la primera aproximación de lo que se puede producir.

La gestión ágil investiga y desarrolla tomando como partida la visión del producto. Durante el desarrollo confronta la realidad de lo que va obteniendo. Su valor, posibilidades y la situación de negocio del entorno en cada momento.

La etapa de especulación se repite en cada iteración del desarrollo, y teniendo como referencia la visión y el alcance del proyecto consiste en:

- Desarrollo / revisión de los requisitos generales del producto.
- Desarrollo de una lista con las funcionalidades esperadas
- Construcción de un plan de entrega: fechas en las que se necesitan las versiones, hitos e iteraciones del desarrollo. Este plan refleja ya el esfuerzo que consumirá el proyecto durante el tiempo.
- En función de las características del modelo de gestión y del proyecto puede incluir también estrategias o planes para la gestión de riesgos.

c) Exploración

Desarrollo de las funcionalidades que para generar el siguiente incremento de producto, ha determinado el equipo en la etapa anterior.

d) Revisión

El equipo y los usuarios revisan las funcionalidades construidas hasta ese momento. Trabajan y operan con el producto real para determinar su alineación y dirección con el objetivo.

e) Cierre

Al llegar a la fecha de entrega de una versión de producto (fijada en la fase de concepto y revisada en las diferentes fases de especulación), se obtiene el producto esperado.

2.4.4 CONCEPTO Y MÉTRICAS.**a) Tiempo real o tiempo de trabajo.**

Tiempo efectivo para realizar un trabajo. Se suele medir en horas o días.

b) Tiempo teórico o tiempo de tarea

Tiempo que sería necesario para realizar un trabajo en “condiciones ideales”: si no se produjera ninguna interrupción, llamadas telefónicas, descansos, reuniones, etc.

c) Puntos de función o puntos de funcionalidad

Unidad de medida relativa para determinar la cantidad de trabajo necesaria para construir una funcionalidad o historia de usuario del Product Backlog.

d) Estimaciones

Cálculo del esfuerzo que se prevé necesario para desarrollar una funcionalidad. Las estimaciones se pueden calcular en unidades relativas (puntos de función) o en unidades absolutas (tiempo teórico).

e) Velocidad absoluta

Cantidad de producto construido en un sprint. Se expresa en la misma unidad en la que se realizan las estimaciones (puntos de función, horas o días reales o teóricos).

f) Velocidad relativa

Cantidad de producto construido en una unidad de tiempo de trabajo.

Ejemplo: puntos de función / semana de trabajo real; o horas teóricas / día de trabajo real.

2.4.5 DOCUMENTOS

a) Product Backlog

Es un documento de alto nivel para todo el proyecto. Contiene descripciones genéricas de todos los requerimientos, funcionalidades deseables, etc. Priorizadas según su retorno sobre la inversión (ROI). Es el qué va a ser construido. Es abierto y cualquiera puede modificarlo. Contiene estimaciones a grosso modo, tanto del valor para el negocio, como del esfuerzo de desarrollo requerido. Esta estimación ayuda al product owner a ajustar la línea temporal y, de manera limitada, la prioridad de las diferentes tareas. Por ejemplo, si dos características tienen el mismo valor de negocio la que requiera menor tiempo de desarrollo tendrá probablemente más prioridad, debido a que su ROI será más alto.

b) Sprint Backlog.

Es un documento detallado donde se describe el cómo el equipo va a implementar los requisitos durante el siguiente sprint. Las tareas se dividen en horas con ninguna tarea de duración superior a 16 horas. Si una tarea es mayor de 16 horas, deberá ser dividida en otras menores. Las tareas en el sprint Backlog nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca oportuno.

c) Burn Down chart

Es una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. Dibujando

una línea que conecte los puntos de todos los Sprints completados, donde se puede ver el progreso del proyecto. Lo normal es que esta línea sea descendente (en casos en que todo va bien en el sentido de que los requisitos están bien definidos desde el principio y no varían nunca) hasta llegar al eje horizontal, momento en el cual el proyecto se ha terminado (no hay más requisitos pendientes de ser completados en el Backlog). Si durante el proceso se añaden nuevos requisitos la recta tendrá pendiente ascendente en determinados segmentos, y si se modifican algunos requisitos la pendiente variará o incluso valdrá cero en algunos tramos.

2.5 INGENIERÍA INVERSA

Se define como el proceso de construir especificaciones de un mayor nivel de abstracción partiendo del código fuente de un sistema software o cualquier otro producto, no se genera código, sino que al contrario, el código fuente es examinado, analizado y convertido en nuevo producto.

La ingeniería inversa nunca cambia la funcionalidad del software sino que permite obtener productos que indican cómo se ha construido el mismo.

La finalidad de la ingeniería inversa es la de desentrañar los misterios y secretos de los sistemas en uso a partir del código. Para ello, se emplean una serie de herramientas que extraen información de los datos, procedimientos y arquitectura del sistema existente.

2.5.1 TIPOS

a) Restructuración del código

Es la forma más básica de reingeniería, es donde fundamentalmente se buscan las fallas. Algunos sistemas cuentan con una programación sólida y confiable, pero algunos módulos pueden demostrar debilidades y es ahí en donde se puede hacer una restructuración de algunas líneas de código para tratar de solucionar fallas o encontrar posibles mejoras.

b) Restructuración de datos

En un sistema que posee una base de datos débil, existirá una latente posibilidad de aplicar reingeniería. Los datos que alimentan al sistema no pueden ser imprecisos, si eso sucede todo el sistema colapsaría y la reingeniería sería más urgente. Esto sucede mucho cuando se solicitan ampliaciones y modificaciones de todas y cada una de las bases de datos.

c) Ingeniería Directa

Se puede decir que es la reingeniería automática, donde aplicaciones obsoletas se modifican a través de un disco que al ser insertado detecta lo que se puede reingeniar y se ejecuta dentro del programa, arrojando una versión eficiente.

2.5.2 HERRAMIENTAS

a) Los depuradores

Es un programa que se utiliza para controlar otros programas. Permite avanzar paso a paso por el código, rastrear fallos, establecer puntos de control y observar las variables y el estado de la memoria en un momento dado del programa que se esté depurando. Los depuradores son muy valiosos a la hora de determinar el flujo lógico del programa.

Un punto de ruptura es una instrucción al depurador que permite parar la ejecución del programa cuando cierta condición se cumpla. Por ejemplo, cuando un programa accede a cierta variable, o llama a cierta función de la API, el depurador puede parar la ejecución del programa.

b) Las herramientas de inyección de fallos

Las herramientas que pueden proporcionar entradas malformadas con formato inadecuado a procesos del software objetivo para provocar errores son una clase de herramientas de inserción de fallos. Los errores del programa pueden ser analizados para determinar si los errores existen en el software objetivo. Algunos fallos tienen implicaciones en la seguridad, como los fallos que permiten un acceso directo del asaltante al ordenador principal o red. Hay herramientas de inyección de fallos basados en el anfitrión que funcionan como depuradores y pueden alterar las condiciones del programa para observar los resultados y también están los inyectores basados en redes

que manipulan el tráfico de la red para determinar el efecto en el aparato receptor.

c) Los Desensambladores

Se trata de una herramienta que convierte código máquina en lenguaje ensamblador. El lenguaje ensamblador es una forma legible para los humanos del código máquina. Los desensambladores revelan que instrucciones máquinas son usadas en el código. El código máquina normalmente es específico para una arquitectura dada del hardware. De forma que los desensambladores son escritor expresamente para la arquitectura del hardware del software a desensamblar.

d) De compiladores

Es una herramienta que transforma código en ensamblador o código máquina en código fuente en lenguaje de alto nivel. También existen de compiladores que transforman lengua intermedio en código fuente en lenguaje de alto nivel. Estas herramientas son sumamente útiles para determinar la lógica a nivel superior como bucles o declaraciones if-then de los programas que son de compilados. Los de compiladores son parecidos a los desensambladores pero llevan el proceso un importante paso más allá.

e) Herramientas Case

Las herramientas de ingeniería de sistemas asistida por ordenador Computer Aided Systems Engineering (CASE) aplican la tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo de sistemas para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas. En el caso de la ingeniería inversa generalmente este tipo de herramientas suelen englobar una o más de las anteriores junto con otras que mejoran el rendimiento y la eficiencia.

2.6 JAVA SERVER FACES (JSF)

Es un framework para desarrollo de aplicaciones basado en el patrón de MVC. Las clases de componentes de interfaz de usuario que se incluyen con la tecnología JavaServer Faces encapsulan la funcionalidad de los componentes. La tecnología JavaServer Faces proporciona un intérprete personalizado y una biblioteca de etiquetas JSP personalizada para representar a un cliente de HTML, permitiendo a los desarrolladores de aplicaciones Java Enterprise Edition (JEE) para utilizar la tecnología JavaServer Faces en sus aplicaciones.

La facilidad de uso es el objetivo principal, la arquitectura Java Server Faces define claramente una separación entre la lógica de aplicación y la presentación, mientras que por lo que es fácil de conectar la capa de presentación para el código de aplicación. Este diseño permite que cada miembro de un equipo de desarrollo de aplicaciones Web para centrarse en

su pieza del proceso de desarrollo, y también proporciona un modelo de programación sencillo para unir las piezas.

2.6.1 CICLO DE VIDA

En el desarrollo en JSF es de crucial importancia comprender y dominar las 6 fases del ciclo de vida de JSF, como muestra la Figura 2.5:

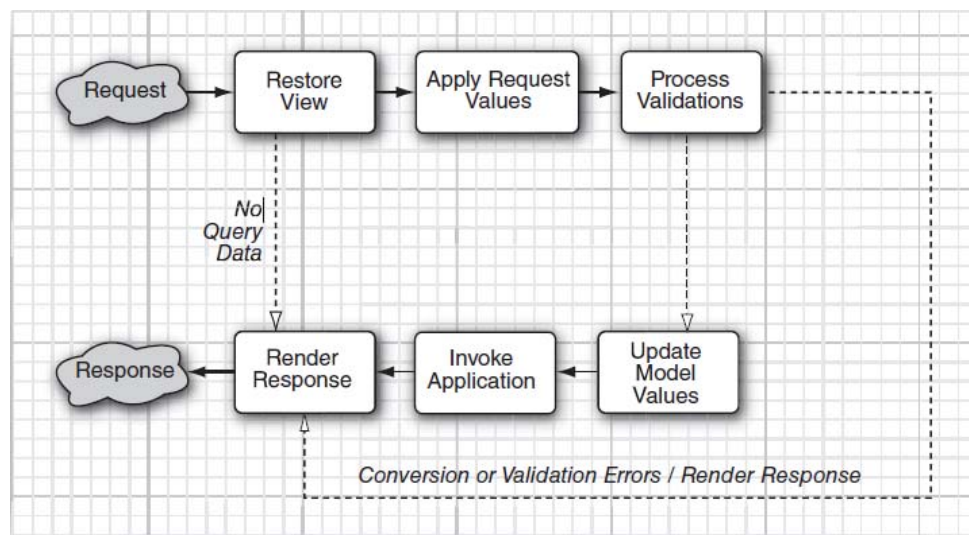


Figura 2.5 Ciclo de vida JSF. Recuperado GEARY, D., & HORSTMANN, C. (2010). core JAVASERVER™ FACES, pag 30, figura 1-15

1. Restore view

Se reconstruye el árbol de componentes.

2. Apply request values

Se leen los valores de la request y se aplican sobre los componentes. En este momento es cuando se llama a los converters. Si hay algún error en la conversión se irá directamente a la fase “render response”. Si un componente tiene “immediate=true” su validación (y el procesamiento de los eventos provocados por esa validación) se hará en esta fase.

3. Process validations

Se validan todos los componentes (todos los que tienen “immediate=false”, ya que los que lo tienen a true ya se validaron en la fase anterior). Si falla alguna de las validaciones se irá directamente a la fase “render response”.

4. Update model values

Los valores de los Backing Beans de JSF se actualizan con los valores que hasta ahora estaban guardados sólo en los componentes.

5. Invoke application

Se invoca a los métodos de los Backing Beans.

6. Render response

Se dibuja la respuesta al usuario. Se pinta el árbol de componentes.

2.7 EJB

2.7.1 INTRODUCCIÓN

El desarrollo de aplicaciones puede ser una tarea compleja, no sólo se debe a que los programas deben hacer su trabajo, tienen que hacerlas bien. Hay una lista de características que el software que un buen software implica:

- Seguridad
- Escalabilidad
- Interoperabilidad
- Robusta / capacidad de recuperación
- Corregir o funciones que se especifican.

Y aunque todos estos son requisitos previos para un producto terminado, ni uno solo es específico para cualquier negocio hasta soluciones personalizadas a los mismos problemas fundamentales que enfrentan todos los demás.

En términos muy sencillos, Enterprise Java Beans (EJB) es una plataforma para la construcción de aplicaciones empresariales portátiles, reutilizables y escalables utilizando lenguaje de programación Java.

EJB permite a los desarrolladores de aplicaciones centrarse en la lógica de negocio sin tener que gastar tiempo en la construcción de servicios tales como transacciones, seguridad, persistencia automática, que usted puede necesitar para la construcción de una aplicación.

Desde el punto de vista del desarrollador, un EJB es una pieza de código Java que se ejecuta en un entorno de ejecución especializado llamado contenedor EJB, que proporciona un número de servicios de componentes. Los servicios de persistencia son proporcionados por framework llamado proveedor de persistencia.

Como se menciona, los componentes EJB viven en un contenedor como muestra la Figura 2.6 donde los EJB, y el contenedor pueden ser vistos como un framework que proporciona servicios a la capa inmediata para el desarrollo de aplicaciones empresariales.

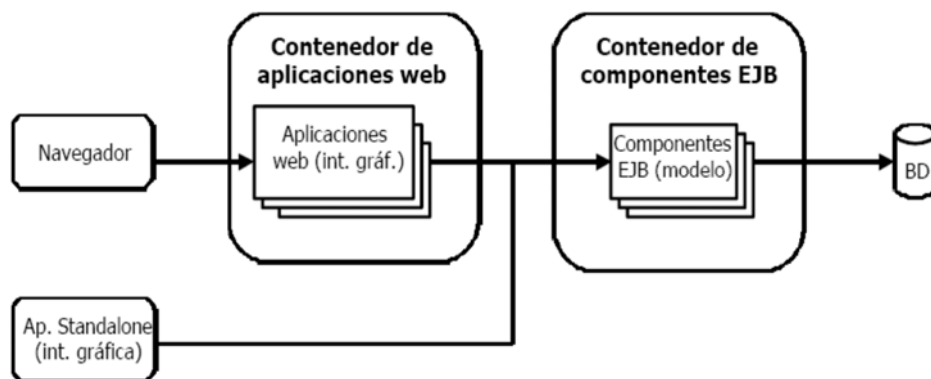


Figura 2.6 Arquitectura de un sistema informático basado en EJB.
Recuperado Burke, B., & Rubinger, A. (2010). Enterprise JavaBeans 3.1.
O'Reilly Media. Pag 44 Figure 5-1

2.7.2 TIPOS

Los EJB residen exclusivamente en el servidor y el cliente debe interactuar con ellos a través de un direccionamiento indirecto.

Un Session Bean es invocado por un cliente para la realización de una operación negocio específico, tales como la comprobación del historial crediticio de un cliente. El nombre de sesión implica que una instancia del bean está disponible para la duración de una "unidad de trabajo" y no sobrevivir a una caída del servidor o el apagado. Hay dos tipos de beans de sesión con estado y sin estado.

a) Stateless (SLSBs)

Los beans de sesión Stateless son útiles para las funciones en las que no se deben mantener el estado de las variables en el servidor, es decir se pierde el valor de las variables.

El contenedor EJB crea un pool en el cual se puede compartir las instancias sin afectar las variables ya que con cada petición se pierde.

CICLO DE VIDA SLSB

El ciclo de vida de un bean SLSBs es muy simple. Tiene sólo dos estados como muestra la Figura 2.7: en el pool se almacenan objetos de sesión que no están en uso. Al momento de la invocación a un SLSBs se creó o devuelve una instancia, luego de ser utilizada la libera o la destruye.

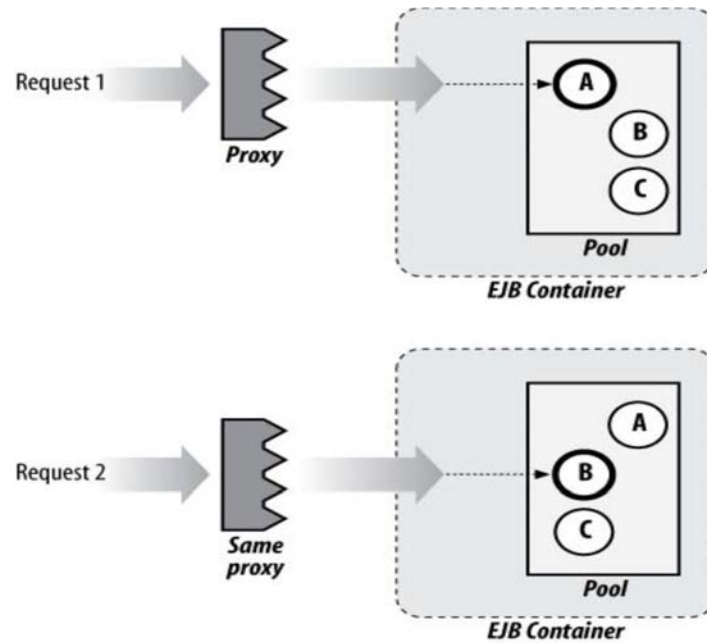


Figura 2.7 Ciclo de vida de un SLSB. Recuperado Burke, B., & Rubinger, A. (2010). Enterprise JavaBeans 3.1. O'Reilly Media. Pag 44 Figure 5-1

b) Statefull (SFSBs)

Los beans de sesión Statefull se diferencian de SLSBs ya que todas las peticiones mantienen los valores de las variables en el servidor.

Invocaciones SFSB mantiene el estado de conversación. Cada objeto proxy SFSB tiene un contexto aislado de la sesión, por lo que las llamadas a una sesión no afectan a otro. Es decir se crea una instancia para cada petición.

CICLO DE VIDA SFSB

Los beans de sesión se dedican a un cliente durante toda su vida como muestra la Figura 2.8, por lo que el intercambio o la puesta en común de los casos no son posible.

Esto significa que cada stateful bean debe ser puesto en estado pasivo antes de que desaloje al fin de preservar la conversación, y debe ser activado para restaurar su estado cuando el EJB objeto se convierte en activo de nuevo.

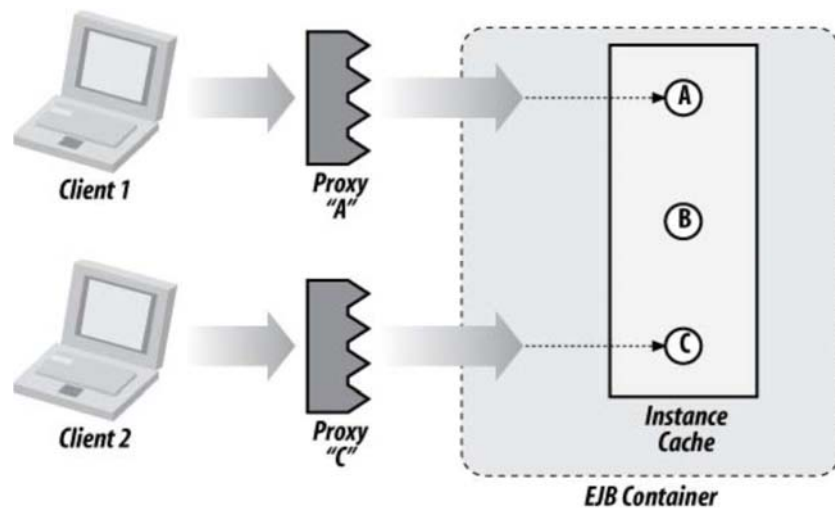


Figura 2.8 Ciclo de vida de un SFSB. Recuperado Burke, B., & Rubinger, A. (2010). Enterprise JavaBeans 3.1. O'Reilly Media. Pag 63 Figure 6-1

c) Singleton

Los beans de sesión Singleton están disponibles desde la versión EJB 3.1 y estos garantizan que solo tendrán una sola instancia en el servidor como

muestra la Figura 2.9, el cual podrá ser compartido por varias peticiones, es decir que en cada servidor que contenga un contenedor EJB se almacenara en cache.

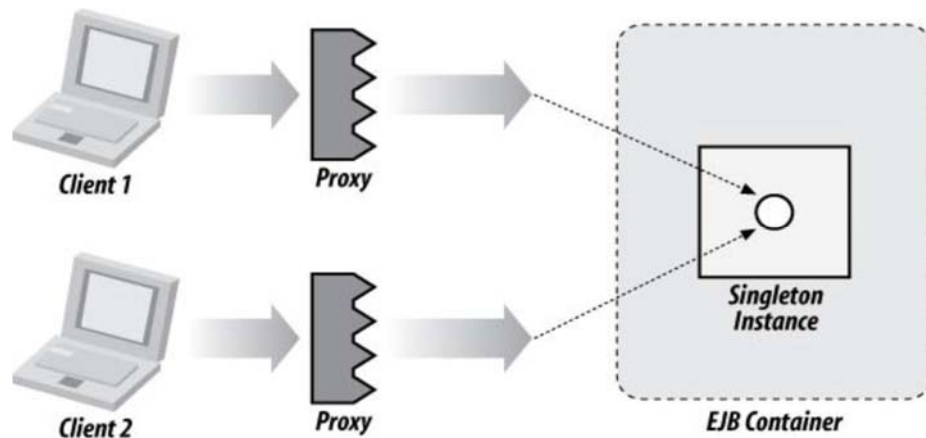


Figura 2.9 Ciclo de vida Singleton. Recuperado Burke, B., & Rubinger, A. (2010). Enterprise JavaBeans 3.1. O'Reilly Media. Pag 81 Figure 7-1

d) Message-Driven Beans (MDB)

Mensajería asíncrona es un paradigma en el que dos o más aplicaciones se comunican a través de un mensaje que describe un evento de negocios. EJB interactúa con los sistemas de mensajería a través de la Java Connector Architecture (JCA) que actúa como una capa de abstracción que permite a cualquier sistema que se adapta como un válido remitente. El bean controlado por mensajes, a su vez, es un oyente que consume mensajes y puede manejarlos directamente o delegar su transformación posterior en otros componentes EJB.

La característica asíncrona de este intercambio significa que un remitente del mensaje no está esperando una respuesta, por lo que no regreso a la persona que llama se proporciona como muestra la Figura 2.10.

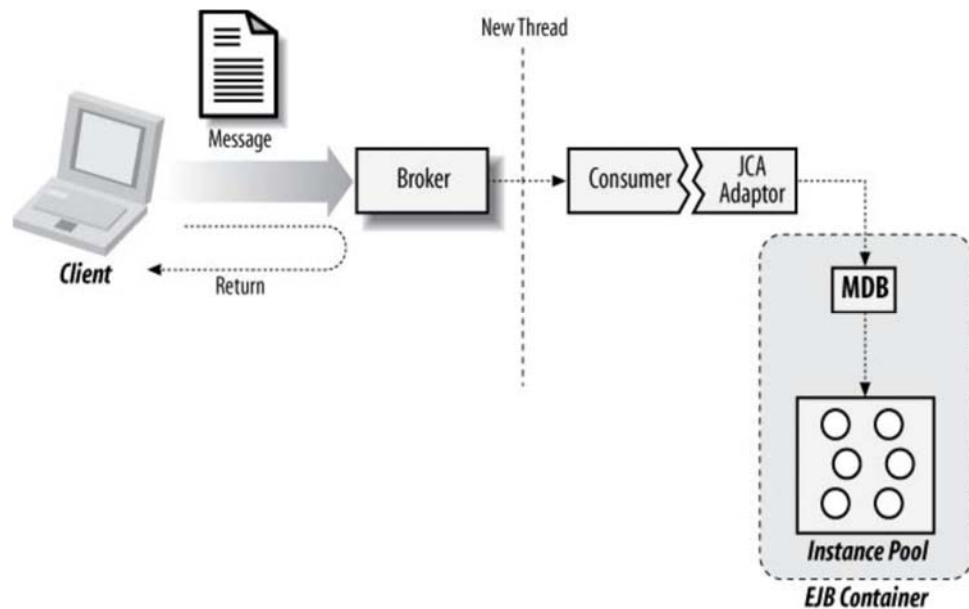


Figura 2.10 Diagrama de flujo de un MDB. Recuperado Burke, B., & Rubinger, A. (2010). Enterprise JavaBeans 3.1. O'Reilly Media. Pag 97 Figure 8-1

2.7.3 INVOCAR UN EJB

a) Invocación Local

Una de las maneras para invocar a un EJB es a través de la llamada local la cual invoca a un EJB que se encuentre en la misma máquina virtual de Java, es decir en el mismo servidor.

Estas invocaciones son mucho más eficientes ya que pueden evitar pasar por una red y pasar referencias a objetos en lugar de serializar los valores.

b) Invocación Remota

La invocación remota se la hace a través de JNDI la cual llama por medio de la clase InitialContext el cual permite invocar un EJB que se encuentra en otro servidor. Esta invocación siempre debe tener objetos serializados para que puedan viajar a través de una red.

2.7.4 EJB Y LA PERSISTENCIA

"El modelo de dominio tiene una clase. La base de datos tiene una tabla. Se ven muy similares. Debería ser sencillo para convertir el uno al otro de forma automática. "Este es un pensamiento probablemente que todos tiene en un momento u otra al escribir un nuevo Data Access Object (DAO) para convertir Java Database Connectivity (JDBC) los conjuntos de resultados en algo orientado a objetos. El modelo de dominio se ve bastante similar al modelo relacional de la base de datos. La técnica de reducción de la brecha entre el modelo de objetos y el modelo relacional que se conoce como mapeo objeto-relacional, a menudo referida como la cartografía llamado ORM.

a) JPA

El framework Java Persistence API se basa en POJO. Aunque el mapeo objeto-relacional es un componente importante de la API, sino que también ofrece soluciones para la integraciones entre la persistencia y las aplicaciones empresariales escalables.

b) Entity

Entidades, en la especificación de Java Persistence, son objetos POJOs serializados que se conservan en la base de datos. Estos representan una tabla mapeando sus columnas como objetos simples y sus relaciones como objetos compuestos en la clase. Estos POJOs pueden ser manipulados desde la lógica de negocio.

2.8 TRANSACCIONES

2.8.1 CMT

En CMT, el contenedor se inicia, se compromete, y revierte una transacción. Más precisamente, el contenedor inicia una transacción JTA antes de que se invoca un método, invoca el método, y en función de lo que sucedió durante la llamada al método, se confirme o se revierte la administrada transacción. Todo lo que se tiene que hacer es decirle al contenedor cómo manejar la transacción ya sea mediante el uso de anotaciones o descriptores de implementación y hacer que se vuelque realizar copias de seguridad de la transacción cuando sea necesario. De forma predeterminada, el contenedor se supone que utiliza CMT sobre todos los métodos de negocio.

2.8.2 BMT

La mayor fortaleza de CMT es también su mayor debilidad. Uso de CMT, que son limitado a que los límites de la transacción se fijan al comienzo y al

final de los métodos de negocio y confían en el contenedor para determinar cuándo una transacción se inicia, se compromete, o se deshace. BMT, por otro lado, permite especificar exactamente estos datos mediante programación, utilizando una semántica similar a la transacción JDBC, modelo con el que es posible que ya esté familiarizado.

2.9 MOTORES DE PERSISTENCIA

Los motores de persistencia son framework que permiten interactuar indirectamente con la base de datos, el cual a través de su dialecto permite realizar consulta a través de POJOS, JPA es implementado por varios motores de persistencia como Hibernate, OpenJPA, Oracle y TopLink.

2.10 ARQUITECTURA

2.10.1 CAPA DE PRESENTACIÓN

a) Java Server Faces (JSR 314)

Es un estándar dentro de la pila JEE (Java Enterprise Edition) que permite el desarrollo de la capa de presentación y control para aplicaciones Web.

JSF permite el desarrollo de aplicaciones Web en base a componentes y eventos. Aproximación típica para el desarrollo de aplicaciones interactivas en entornos de escritorio. De esta manera abandona la aproximación tradicional y más limitada de "petición/respuesta", en la que se trabaja básicamente con documentos y apuesta por el trabajo con pantallas de interacción.

b) Java Servlets

Los servlets son la tecnología de la plataforma Java de elección para la ampliación y mejora de los servidores Web. Los servlets proporcionan un componente basado en método independiente de la plataforma para la creación de aplicaciones basadas en Internet, sin las limitaciones de rendimiento de los programas CGI.

c) JavaServer Pages y Expression Language

JavaServer Pages (JSP) permite a los desarrolladores Web y diseñadores para desarrollar rápidamente y mantener con facilidad, ricas en información, las páginas Web dinámicas que aprovechan los sistemas empresariales existentes. Como parte de la familia de tecnologías Java, JSP tecnología permite el desarrollo rápido de aplicaciones basadas en Web que son independientes de la plataforma. La tecnología JSP separa la interfaz de usuario de generación de contenidos, permitiendo a los diseñadores cambiar el diseño general de la página, sin alterar el contenido dinámico subyacente.

d) JavaServer Pages Standard Tag Library

Encapsula como etiquetas simples la funcionalidad básica común a muchas aplicaciones Web. JSTL es compatible con tareas comunes estructurales como la iteración y condicionales, etiquetas para manipular documentos XML, internacionalización y SQL. También proporciona una estructura para las etiquetas personalizadas con etiquetas JSTL. JSTL 1.2 El lanzamiento de Mantenimiento se alinee con el lenguaje de expresión

unificada (EL) que se entrega como parte de la JavaServer Pages (JSP) 2.1 especificación. Gracias a la EL Unificado, etiquetas JSTL, como las etiquetas JSTL iteración, ahora se puede utilizar con componentes JavaServer Faces de una forma intuitiva. JSTL 1.2 es parte de la plataforma Java EE 5.

e) jQuery

Es una librería JavaScript para conseguir una interfaz rica de usuario en aplicaciones Web RIA (Rich Internet Application). Esto quiere decir que ésta librería JavaScript permite dotar a las páginas Web de efectos visuales para asemejar su uso al de aplicaciones tradicionales de escritorio, mejorando en gran medida la usabilidad de las mismas.

f) Primefaces

Se apoya en el uso extensivo de jQuery para la comunicación Ajax y el uso de componentes visuales, aunque también hace uso de la librería Javascript YUI de Yahoo.

g) RichFaces

Se basa en el innovador soporte para Ajax, que comenzó con RichFaces 3 y está estandarizado en JSF 2. Además de la ampliación de estas capacidades Ajax, RichFaces también mejora otras áreas de JSF 2, incluyendo la optimización del rendimiento de usabilidad, los recursos dinámicos, pelado y desarrollo de componentes. Esto permite a los usuarios aprovechar al máximo todas las mejoras de productividad de JSF 2.

2.10.2 CAPA DE NEGOCIO

a) Enterprise JavaBeans 3.1 (JSR 318)

Enterprise JavaBeans es la arquitectura de componentes del lado del servidor para Java Enterprise Edition. Permite el desarrollo rápido y sencillo de aplicaciones distribuidas, transaccionales, seguras y portátiles basado en la tecnología Java.

b) Contexts and Dependency Injection

CDI es un conjunto de servicios que, utilizados en conjunto, hacen que sea fácil para los desarrolladores utilizar beans de empresa, junto con la tecnología JavaServer Faces en aplicaciones para la plataforma Java EE es uno de los varios Java EE 6 características que ayudan a tejer juntos la capa Web y el nivel transaccional de la plataforma Java EE. Diseñado para usar con los objetos con estado, CDI también tiene muchos usos más amplios, lo que permite a los desarrolladores una gran flexibilidad para integrar diversos tipos.

c) Ireport

Es el más popular motor de creación de informes en código abierto. Está escrito completamente en Java y es capaz de utilizar los datos procedentes de cualquier tipo de fuente de datos y presentar los documentos con precisión de píxeles, lo cuales se pueden ver, imprimir o exportar en una variedad de formatos de documentos incluyendo HTML, PDF, Excel, OpenOffice y Word.

2.10.3 CAPA DE PERSISTENCIA

a) Java Persistence 2.0

El Java Persistence API se basa en las ideas de los principales frameworks de persistencia y APIs como Hibernate, TopLink Oracle y Java Data Objects (JDO), y así como en la primera EJB persistencia gestionada por contenedor. El Grupo de Expertos para la Enterprise JavaBeans 3.0 Specification (JSR 220) tiene una representación de expertos en todas estas áreas, así como de otras personas notables de la comunidad persistencia.

b) Hibernate

Es un motor de persistencia basado en Java Persistence (JPA) el cual da soluciones de almacenamiento NoSQL. Se reutiliza Hibernate objeto de gestión de ciclo de entidades persistentes en base de datos relacional. Para las consultas SQL Java Persistence Query Language como una interfaz para consultar los datos almacenados.

2.10.4 BASE DE DATOS

MySQL Es un sistema de gestión de bases de datos relacional y multiusuario. Es una implementación Cliente-Servidor que consta de un servidor y diferentes clientes (programas/librerías), además que es un software de código abierto esto quiere decir que es accesible para cualquiera, para usarlo o modificarlo, Una de las características más interesantes de MySQL es que permite recurrir a bases de datos multiusuario a través de la

Web y en diferentes lenguajes de programación que se adaptan a diferentes necesidades y requerimientos.

2.11 SISTEMAS DE CONTROL DE VERSIONES

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra dicho producto en un momento dado de su desarrollo o modificación. Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión dando lugar a los llamados sistemas de control de versiones.

a) SUBVERSIÓN

Es un sistema de control de versiones diseñado específicamente para reemplazar al popular CVS. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como SVN por ser el nombre de la herramienta utilizada en la línea de comando.

Una característica importante de Subversión es que, a diferencia de CVS, los demás archivos con versionamiento no tienen cada uno un número de revisión independiente, en cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en un instante determinado del repositorio que se está trabajando.

Subversión puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras. A cierto nivel, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones.

CAPÍTULO 3

ANÁLISIS Y DISEÑO

3.1 ANÁLISIS DE REQUERIMIENTOS

La etapa de Análisis de Requerimientos, es la primera etapa en el desarrollo. Se comienza cuando el Cliente ha detectado una ausencia, falla o falta de oportunidad de la información o simplemente, luego que la organización ha determinado un cambio en sus políticas, reglas o tecnologías a aplicar.

En esta etapa se debe responder una pregunta fundamenta, ¿Qué es lo que quiere el Cliente?, y para ello se debe determinar la situación actual de la empresa, a través de los requeriremos del cliente.

La ingeniería de requerimientos es primordial para el desarrollo de software, ya que su función principal consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema.

La Ingeniería de Requerimientos tiene su importancia en el hecho de servir como mecanismo para:

- Comprender lo que el cliente quiere.
- Analizar las necesidades.

- Especificar la solución sin ambigüedades.
- Validar la factibilidad.
- Validar la especificación.
- Negociar una solución razonable.
- Gestionar los requisitos conforme estos se transforman en una operación del sistema.

Un requerimiento es una necesidad de un usuario para resolver un problema o alcanzar un objetivo. Es un atributo necesario para el sistema.

3.2 REQUERIMIENTOS FUNCIONALES

3.2.1 MODELADO DE USUARIOS

Los perfiles de usuario son elementos importantes en los sistemas adaptativos, puesto que permiten conocer las necesidades o preferencias de los usuarios. En este sentido surge el concepto de Modelo de usuario el cual contiene información de los datos personales, preferencias e intereses, así como datos del entorno del usuario.

El modelado de usuarios consiste en definir usuarios en base a atributos comunes. Los atributos son los que determinan la clasificación de los usuarios tales como acceso, condiciones de acceso, manipulación, de la información.

De acuerdo al grado de conocimiento semántico y sintáctico que el usuario posea, se puede realizar una clasificación general de estos en tres categorías.

Los conceptos de conocimiento sintáctico y semántico tienen una correspondencia directa con las capas que presenta el software. El corazón funcional procesa información semántica mientras que el procesamiento de información sintáctica pertenece a las capas propias de la interfaz del usuario.

3.3 PRODUCT BACKLOG

3.3.1 ANÁLISIS AMBIENTAL

a) Componentes

Tabla 3.1 Historia de usuario: Componentes

Historia de usuario	
Número: 2	Usuario: Mario Ron
Nombre: Componentes	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario ingresa la descripción, tipo de componente (interno o externo) y observaciones del componente. 2. El tipo de componente es una opción requerida. 3. Los departamentos son los únicos que pueden ingresar datos en la pantalla. 	
Observación: Ninguna	

b) Subcomponentes

Tabla 3.2 Historia de usuario: Sub componentes

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Subcomponentes	
Prioridad en el negocio (Alta/Media/Baja): Alta	
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario ingresa la descripción, componente y observaciones del subcomponente. 2. El usuario guarda el subcomponente. 3. El usuario puede ver los reportes de subcomponentes 4. Variación: Falla en selección de componente 5. Si no está seleccionado un componente válido, saldrá un mensaje que tiene que escoger un componente válido. 6. Los departamento son los únicos que pueden ingresar datos en la pantalla 	
Observación: Ninguna	

c) Análisis Interno

Tabla 3.3 Historia de usuario: Análisis interno

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Análisis Interno	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario escoge un componente y subcomponente tipo interno 2. El usuario ingresa la descripción del resultado, y el tipo de resultado (fortaleza, debilidad o acción gerencial). 3. Se pueden generar reportes 4. Variación: Falla en selección de resultado 5. Si no está seleccionado un tipo de resultado válido, saldrá un mensaje que tiene que escoger un tipo de resultado válido. 6. Los departamentos son los únicos que pueden ingresar datos en la pantalla. 	
Observación:	
Precondición: Tiene que estar creado un componente interno, y un subcomponente que dependa de éste.	

d) Análisis Externo

Tabla 3.4 Historia de usuario: Análisis externo

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Análisis Externo	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario escoge un componente y subcomponente. 2. El usuario ingresa la descripción del resultado, y el tipo de resultado (oportunidad, amenaza o acción gerencial). 3. El usuario pueden ver los reportes 4. Variación: Falla en selección de resultado 5. Si no está seleccionado un tipo de resultado válido, saldrá un mensaje que tiene que escoger un tipo de resultado válido. 6. Los departamentos son los únicos que pueden ingresar datos en la pantalla 	
Observación:	
Precondición: Tiene que estar creado un componente externo, y un subcomponente que dependa de éste.	

e) Análisis extendido FODA

Tabla 3.5 Historia de usuario: Análisis extendido FODA

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Análisis Extendido FODA	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario puede priorizar las Oportunidades y Fortalezas con la matriz de Holmes 2. El usuario puede priorizar las Acciones Gerenciales, Amenazas y debilidades si es entrada o salida 3. Una vez priorizado el usuario podrá seleccionar los resultados 4. Los resultados priorizados deben salir en orden de la sumatoria 5. Todos los departamentos, localizaciones, y la corporación pueden priorizar. 	
Observación:	
Precondición: Tiene que estar creado un componente externo, y un subcomponente que dependa de éste.	

f) Análisis Síntesis FODA

Tabla 3.6 Historia de usuario: Análisis síntesis FODA

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Síntesis FODA	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
El usuario pueden visualizar el FODA priorizado	
Observación: Ninguna	

g) Expectativas

Tabla 3.7 Historia de usuario: Expectativas

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Expectativas	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario puede insertar, modifica, o elimina y prioriza las expectativas de acuerdo a la calificación que se le dé por medio del procedimiento de la Matriz de Holmes 2. Deben haber al menos dos expectativas para poder priorizar 	
Observación: Ninguna	

h) Problemas

Tabla 3.8 Historia de usuario: Problemas

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Problemas	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El operador inserta, modifica, o elimina y prioriza las problemas de acuerdo a la calificación que se le dé por medio del procedimiento de la Regla de Decisiones. 2. El Evaluador o el Operador visualizaran los reportes de problemas 3. Deben haber al menos dos problemas para poder priorizar 	
Observación: Ninguna	

3.3.2 DIRECCIONAMIENTO ESTRATÉGICO

a) Matriz FA

Tabla 3.9 Historia de usuario: Matriz FA

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Matriz FA	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario prioriza cada fortaleza con cada una de las amenazas (A, M o B). 2. El usuario ingresa las estrategias que obtuvieron una calificación de A. 3. El usuario guarda las estrategias 4. El usuario pueden ver los reportes 5. Variación: Modificación de estrategias 6. Sólo se puede modificar una estrategia a la vez. 	
Observación:	
Precondición:	
<ol style="list-style-type: none"> 1. Tiene que estar creado un componente externo y un interno, y subcomponentes que dependan de cada uno de estos. 2. Estar creadas Fortalezas y Amenazas, además de acciones gerenciales. 3. Deben estar priorizadas y seleccionadas las fortalezas y Amenazas. 	

b) Matriz FO

Tabla 3.10 Historia de usuario: Matriz FO.

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Matriz FO	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario prioriza cada fortaleza con cada una de las oportunidades (A, M o B). 2. El usuario ingresa las estrategias solo a las que obtuvieron una calificación de A. 3. El usuario guarda las estrategias 4. El usuario pueden ver los reportes 5. Variación: Modificación de estrategias 6. Sólo se puede modificar una estrategia a la vez. 	
Observación:	
Precondición:	
<ol style="list-style-type: none"> 1. Tiene que estar creado un componente externo y un interno, y subcomponentes que dependan de cada uno de estos. 2. Debe estar creadas las Fortalezas y Oportunidades, además de acciones gerenciales. 3. Deben estar priorizadas y seleccionadas las fortalezas y Oportunidades 	

c) Matriz DO

Tabla 3.11 Historia de usuario: Matriz DO.

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Matriz DO	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario prioriza cada debilidad con cada una de las oportunidades (A, M o B). 2. El usuario ingresa estrategias solo a las que obtuvieron una calificación de A. 3. El usuario guarda las estrategias 4. El usuario pueden ver los reportes 5. Variación: Modificación de estrategias 6. Sólo se puede modificar una estrategia a la vez. 	
Observación:	
Pre-condición:	
<ol style="list-style-type: none"> 1. Tiene que estar creado un componente externo y un interno, y subcomponentes que dependan de cada uno de estos. 2. Debe estar creadas las Debilidades y Oportunidades, además de acciones gerenciales. 3. Deben estar priorizadas y seleccionadas las Debilidades y Oportunidades. 	

d) Matriz DA

Tabla 3.12 Historia de usuario: Matriz DA

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Matriz DA	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario prioriza cada debilidad con cada una de las amenazas (A, M o B). 2. El usuario ingresa estrategias solo a las que obtuvieron una calificación de A. 3. El usuario guarda las estrategias 4. El usuario pueden ver los reportes 5. Variación: Modificación de estrategias 6. Sólo se puede modificar una estrategia a la vez. 	
Observación:	
Pre-condición:	
<ol style="list-style-type: none"> 1. Tiene que estar creado un componente externo y un interno, y subcomponentes que dependan de cada uno de estos. 2. Debe estar creadas las Debilidades y Amenazas, además de acciones gerenciales. 3. Deben estar priorizadas y seleccionadas las Debilidades y Amenazas 	

e) Soluciones

Tabla 3.13 Historia de usuario: Soluciones.

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Soluciones	
Prioridad en el negocio (Alta/Media/Baja):	ALTA
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario inserta, modifica, o elimina y prioriza las problemas de acuerdo a la calificación. 2. El usuario pueden visualizar reportes. 	
Observación:	
Pre-condición: Es necesario que existan problemas previos para poder dar soluciones a los mismos.	

f) Principios Filosóficos

Tabla 3.14 Historia de usuario: Principios filosóficos.

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Principios filosóficos	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario inserta, modifica, o elimina los principios filosóficos 2. El usuario pueden visualizar reportes. 	
Observación: Ninguna	

g) Valores Institucionales

Tabla 3.15 Historia de usuario: Valores institucionales.

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Valores Institucionales	
Prioridad en el negocio (Alta/Media/Baja):	ALTA
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario inserta, modifica, o eliminar los valores institucionales. 2. El usuario guarda los valores institucionales. 3. El usuario puede visualizar los reportes. 	
Observación: Ninguna	

h) Misión

Tabla 3.16 Historia de usuario: Misión.

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Misión	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario crea un componente de la misión; el cual podrá ser modificado, eliminado, etc. 2. El usuario podrá visualizar reportes. 3. Variación: Modificación de la Misión 4. La misión podrá ser modificada pero no eliminada. 	
Observación:	
Pre-condición: Se debe hacer una descripción de la misión.	

i) Visión

Tabla 3.17 Historia de usuario: Visión.

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Visión	
Prioridad en el negocio (Alta/Media/Baja): ALTA	
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario crea un componente de la visión; el cual podrá ser modificado, eliminado. etc. 2. El usuario podrá visualizar reportes. 3. Variación: Modificación de la Visión 4. La visión podrá ser modificada pero no eliminada. 	
Observación:	
Pre-condición: Se debe hacer una descripción de la visión.	

j) Estrategia general

Tabla 3.18 Historia de usuario: Estrategia general.

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Estrategia general	
Prioridad en el negocio (Alta/Media/Baja):	ALTA
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El Usuario escoge la persona responsable de ejecutar la estrategia 2. El Usuario ingresa el enunciado y la descripción de la estrategia 3. El Usuario determina las fechas de inicio y fin de la estrategia 4. El Usuario escoge al evaluador de la estrategia que dará un informe de acuerdo a los estándares establecidos 5. El Usuario guarda la estrategia 6. El Usuario ingresa los indicadores de la estrategia 7. El Usuario verá los reporte de las estrategias 8. Las estrategias se crearan como objetivos en los niveles inferiores de la estructura corporativa 	
Observación:	
El responsable de la estrategia, no puede ser a la vez responsable de su ejecución.	

k) Objetivos

Tabla 3.19 Historia de usuario: Objetivos.

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Administración de objetivos	
Prioridad en el negocio (Alta/Media/Baja):	ALTA
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario escoge la fase, perspectiva y unidad organizacional a las cuales pertenecería el objetivo a administrar. 2. El usuario escoge la persona responsable de ejecutar el objetivo 3. El usuario ingresa el enunciado y la descripción del objetivo 4. El usuario determina las fechas de inicio y fin del objetivo 5. El usuario escoge al evaluador del objetivo que dará un informe de acuerdo a los estándares establecidos 6. El usuario guarda el objetivo 7. El usuario ingresa los indicadores del objetivo 8. El usuario pueden ver reporte de objetivos seleccionando el objetivo 	
Observación: Ninguna	

I) Proyectos

Tabla 3.20 Historia de usuario: Proyectos.

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Administración de proyectos	
Prioridad en el negocio (Alta/Media/Baja): ALTA	
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. Existen dos tipo de proyectos abierto y cerrados 2. Una vez aprobado el proyecto no puede ser modificado 3. Los hitos de cada proyecto debe tener fechas de fin que alerten cuando este por finalizar 4. Cada proyecto se creara, gestionará, evaluará y aprobará 5. Solo puede crear y gestionar el director de proyectos 6. Aprueban y evalúan los proyectos la persona de un nivel superior 7. El director de proyecto ingresa la descripción del proyecto 8. El director de proyecto ingresa los indicadores del proyecto 9. El director de proyecto ingresar objetivos específicos del proyecto 10.El director de proyecto ingresar descripción de tarea del proyecto y responsable de la misma 11.El director de proyecto ingresar recurso del proyecto, costo y partida del mismo 12.El director de proyecto guarda el proyecto 	
Observación: Ninguna	

3.3.3 ESTRUCTURA CORPORATIVA

Tabla 3.21 Historia de usuario: Estructura corporativa.

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Modulo de estructura corporativa	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario podrá crear o varias localizaciones a partir de la corporación 2. Una vez creado las localizaciones puede crear varias sedes a partir de una localización 3. Se puede editar, eliminar, ver, y agregar cada estructura corporativa. 	
Observación:	
Precondiciones: Debe estar creado la corporación.	

3.3.4 PERFILES

Tabla 3.22 Historia de usuario: Perfiles.

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Modulo de perfiles	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario podrá administrar los perfiles 2. El usuario dependiendo de la pantalla podrá escoger si puede crear, actualizar, eliminar, generar reportes y aprobar 3. El usuario podrá dar permisos a todas las pantallas del sistema. 	
Observación:	
Precondición: Deberá estar creado al menos una estructura organizacional.	

3.3.5 USUARIOS

Tabla 3.23 Historia de usuario: Usuarios

Historia de usuario	
Número: 1	Usuario: Mario Ron
Nombre: Modulo de usuarios	
Prioridad en el negocio (Alta/Media/Baja):	Alta
Modificación de historia de usuario número	
Descripción:	
<ol style="list-style-type: none"> 1. El usuario ingresara a través de un usuario y contraseña 2. Se presenta el menú con las opciones que tiene disponible el usuario para el perfil que tenga 3. Si el usuario no pudo ser autorizado, no podrá ingresar al sistema. Este problema puede ser causado por error en el ingreso de usuario y clave o porque el usuario no existe como tal dentro del sistema. 4. Cuando el usuario ingresa por primera vez al sistema, el mismo le obliga a que cambie la contraseña. 5. Mostrar una opción “¿Olvido su contraseña?” la cual permite enviar a su mail la contraseña. 6. Se debe poder administrar los usuarios. 7. El usuario podrá cambiar la contraseña. 	
Observación:	
Precondiciones: Debe estar creado al menos un perfil para poder asociar al usuario	

3.4 DIAGRAMAS DE CASOS DE USO

3.4.1 ANÁLISIS AMBIENTAL

a) Componente

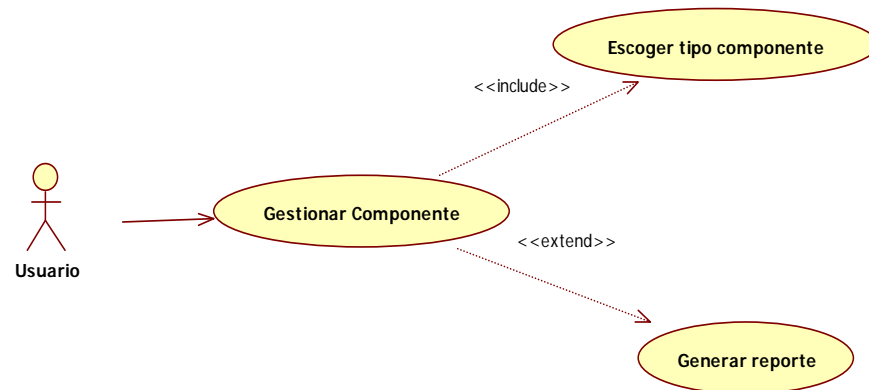


Figura 3.1 Caso de uso: Componentes

b) Subcomponente

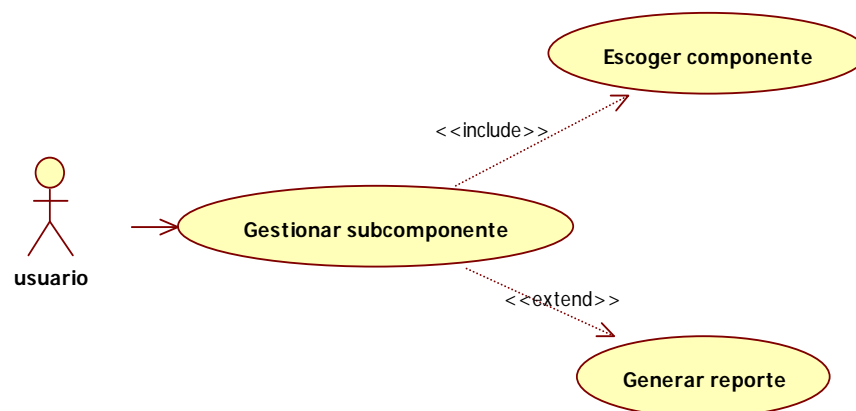


Figura 3.2 Caso de uso de subcomponentes

c) Análisis Interno

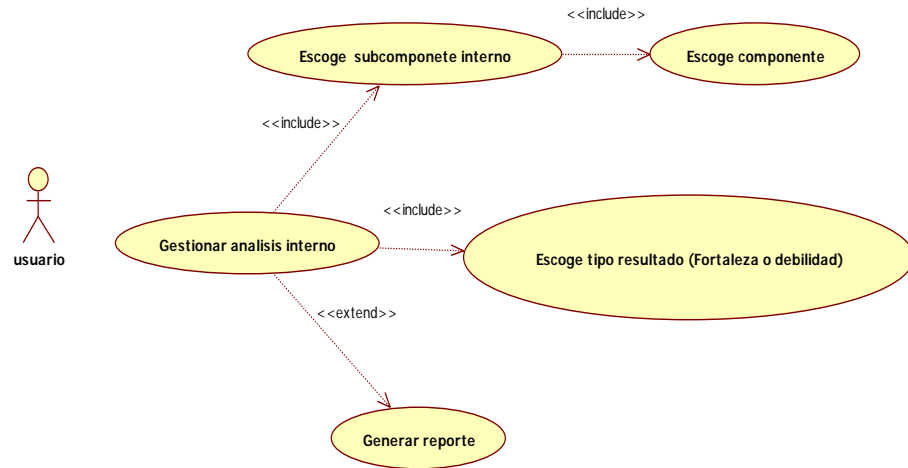


Figura 3.3 Caso de uso: Análisis interno

d) Análisis Externo

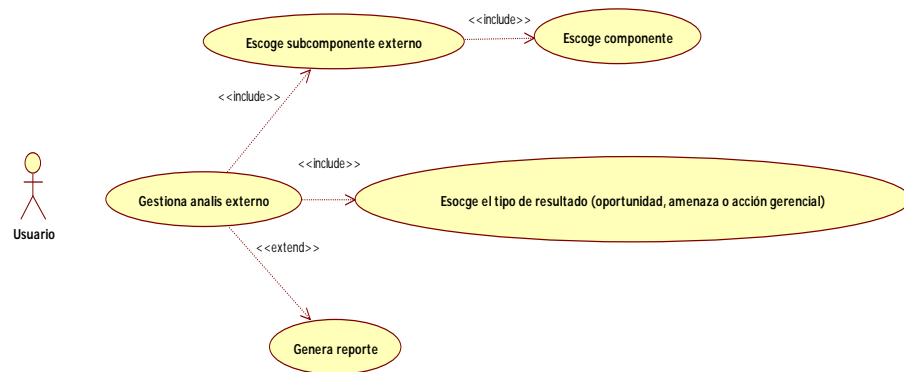


Figura 3.4 Caso de uso: Análisis externos

e) Análisis extendido FODA

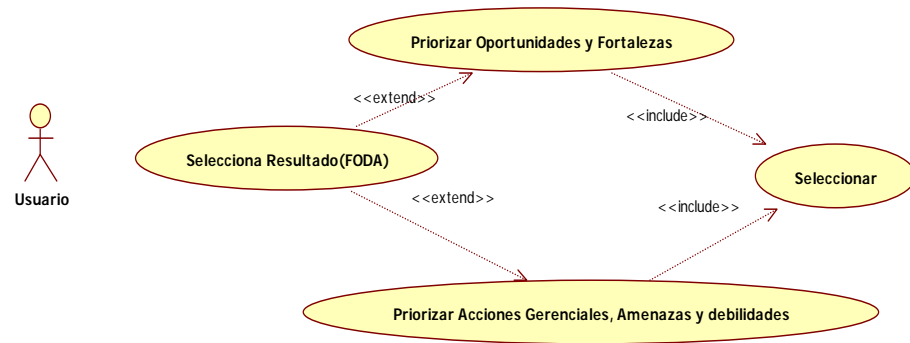


Figura 3.5 Caso de uso: Análisis FODA.

f) Expectativas

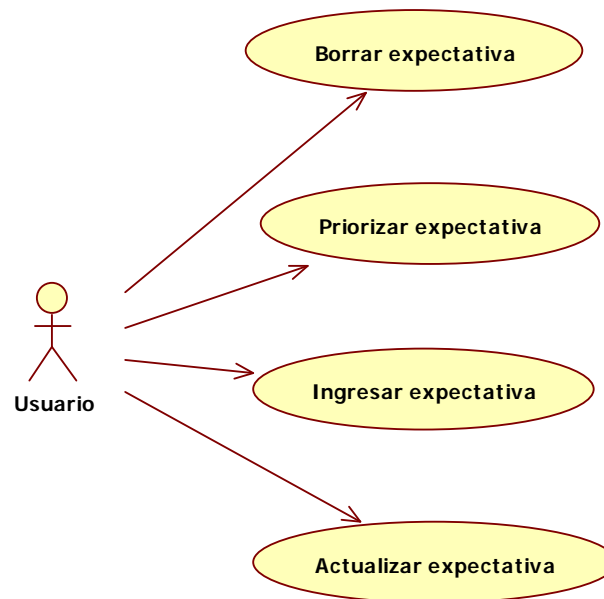


Figura 3.6 Caso de uso: Expectativas

g) Problemas

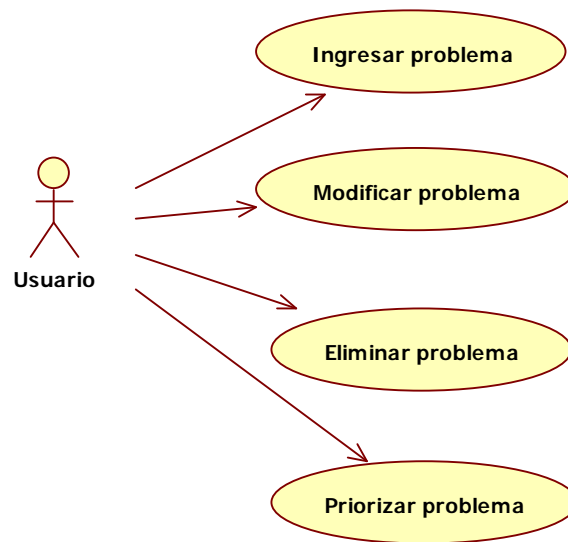


Figura 3.7 Caso de uso: Problemas

3.4.2 DIRECCIONAMIENTO ESTRATÉGICO

a) Matriz FA

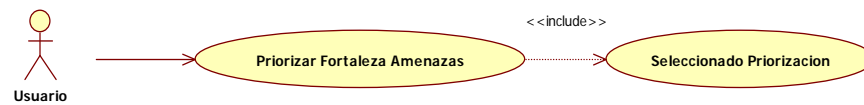


Figura 3.8 Caso de uso: Matriz FA

b) Matriz FO

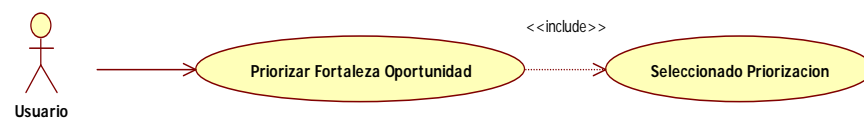


Figura 3.9 Caso de uso: Matriz FO

c) Matriz DO

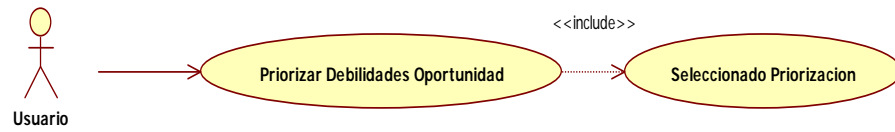


Figura 3.10 Caso de uso: Matriz DO

d) Matriz DA



Figura 3.11 Caso de uso: Matriz DA

e) Soluciones

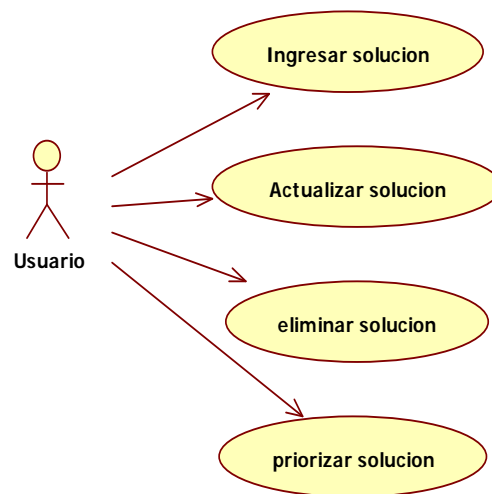


Figura 3.12 Caso de uso: Soluciones

f) Principios Filosóficos

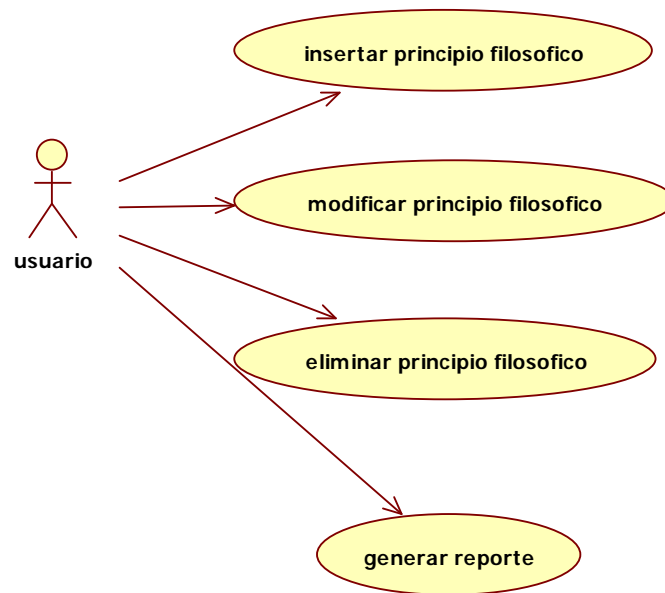


Figura 3.13 Caso de uso: Principios filosóficos

g) Valores Institucionales

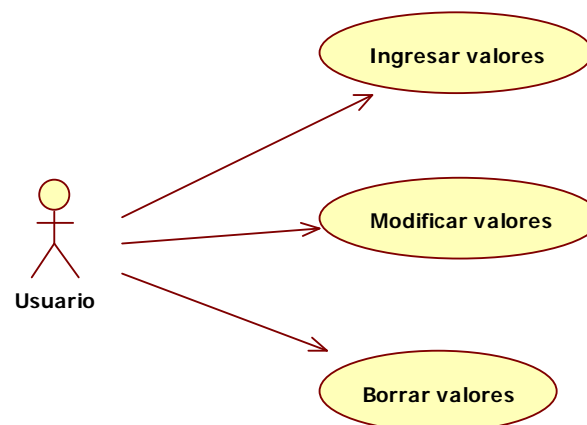


Figura 3.14 Caso de uso: Valores institucionales

h) Misión

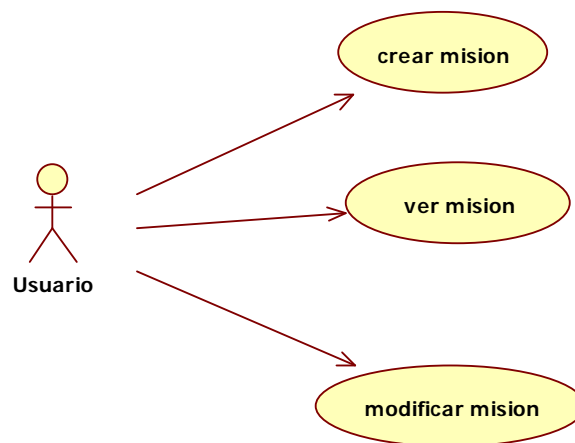


Figura 3.15 Caso de uso: Misión

i) Visión

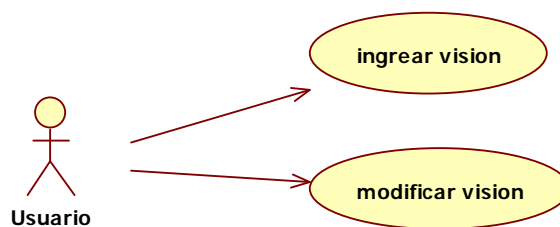


Figura 3.16 Caso de uso: Visión

j) Estrategia organizacional

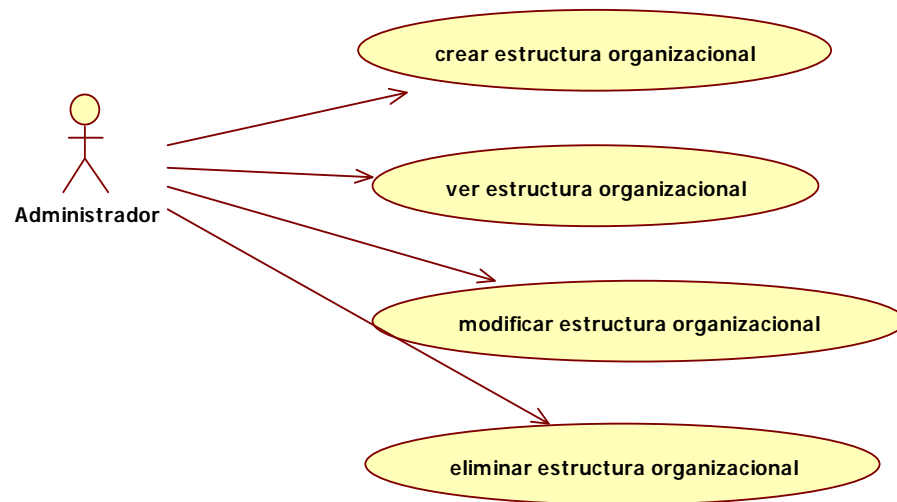


Figura 3.17 Caso de uso: Estructura organizacional

k) Políticas generales

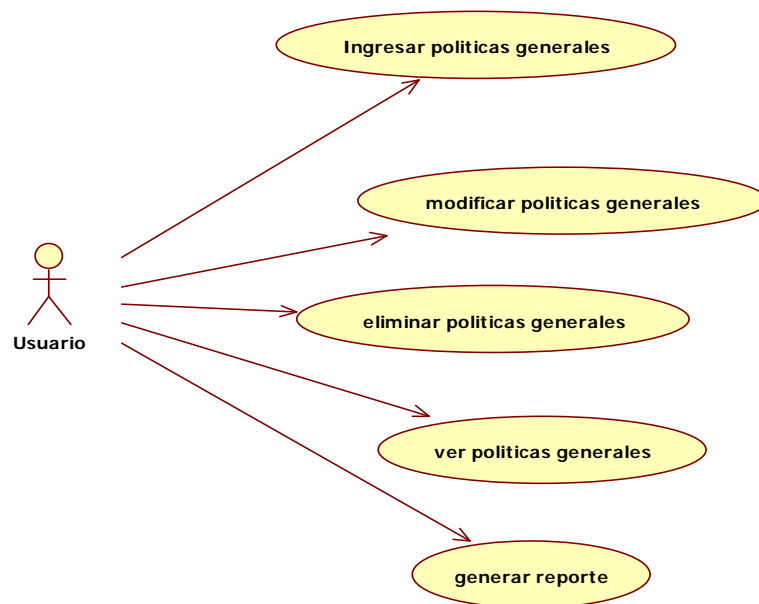


Figura 3.18 Caso de uso: Políticas generales

3.4.3 PERFILES

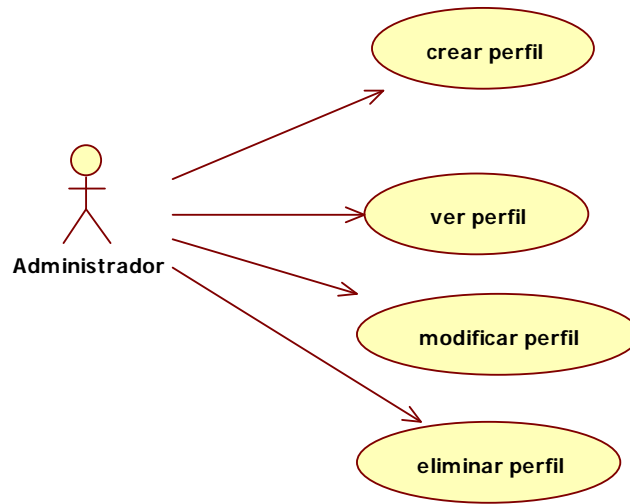


Figura 3.19 Caso de uso: Perfiles

3.4.4 ESTRUCTURA CORPORATIVA

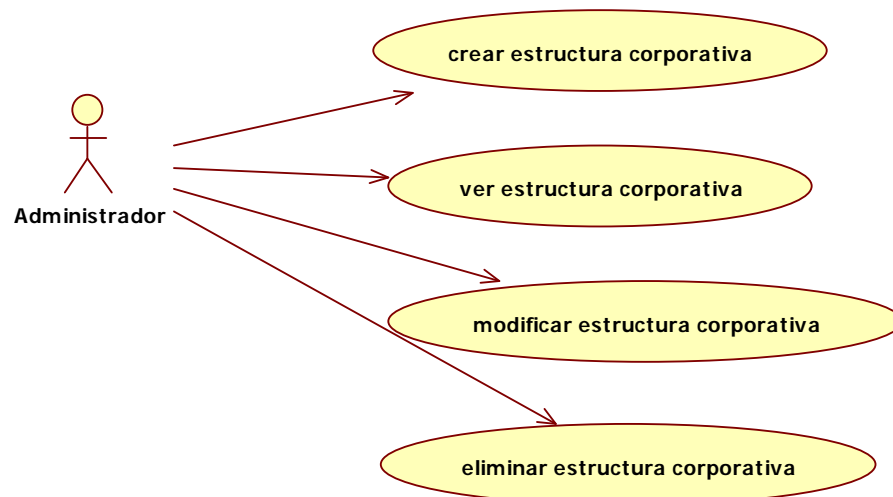


Figura 3.20 Caso de uso: Estructura corporativa

3.4.5 USUARIOS

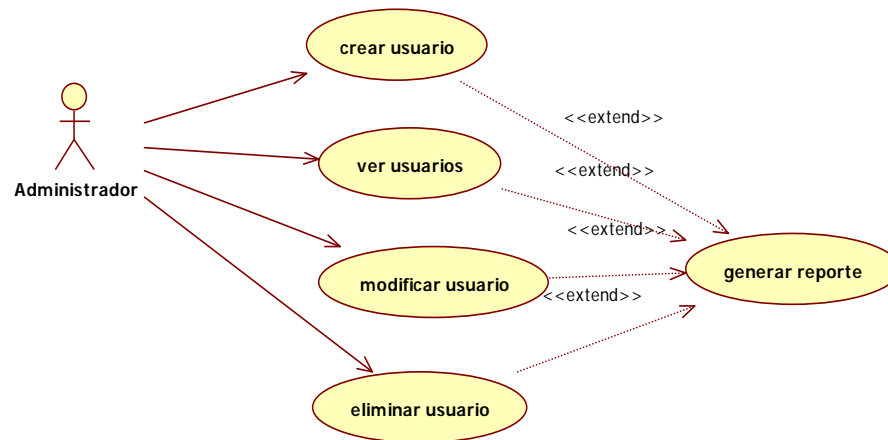


Figura 3.21 Estructura corporativa

3.5 ALCANCE PROYECTO

El proyecto SISPLAGER es un proyecto que es desarrollado en una arquitectura multicapa que permitirá a la UDI llevar una planificación gerencial, que permite a los usuarios la elaboración del análisis ambiental, direccionamiento estratégico y la evaluación y control a través de sus respectivos módulos.

3.5.1 ENTREGABLES DEL PROYECTO

Tabla 3.24 Lista de entregables

LISTA DE ENTREGABLES		
#	Tipo	Descripción
1	Grupo	Documento diseño Proyecto
2	Grupo	Prototipo del software
3	Individual	Manual de usuario
4	Grupo	Manual de técnico

3.5.2 RESTRICCIONES DEL PROYECTO

El proyecto SISPLAGER no es un BPMS por tal motivo no administrará conjunto de servicios y herramientas que facilitan la administración de procesos de negocio. Por administración de procesos se entiende: análisis, definición, ejecución, monitoreo, y control de los procesos.

3.5.3 SUPUESTOS DEL PROYECTO

El proyecto intenta la incorporación de todos los involucrados en la planificación estratégica en la medida de lo posible.

3.5.4 PROPÓSITO

El propósito principal del sistema es tener el control del estado de salud corporativa y la forma como se están encaminando las acciones para alcanzar

la visión. A partir de la visualización y el análisis de los indicadores balanceados, pueden tomarse acciones preventivas o correctivas que afecten el desempeño global de la empresa. Por sus características, el sistema se puede implementar a nivel corporativo o en unidades de negocio con visión y estrategias de negocios definidas y que mantengan cierta autonomía funcional.

3.6 ESTRUCTURA DE DESGLOSE DE TRABAJO

Se trata de un agrupamiento de información que describe los entregables de elementos del proyecto, que organizan y definen el alcance del proyecto.

Es usado para desarrollar o conformar un entendimiento común del alcance del proyecto. Cada nivel de desglose proporciona más detalles de los elementos del proyecto.

En general son presentados en forma de cuadros o diagramas jerárquicos como muestra la Figura 3.22. Cada ítem del EDT es asignado a un identificador único, que se conoce como código de cuenta.

La descripción de cada elemento de trabajo es volcado en el diccionario de la EDT, que incluye descripciones del paquete de trabajo, y otras informaciones de planeamiento como datos de cronograma, presupuesto de costos y asignaciones de recursos.

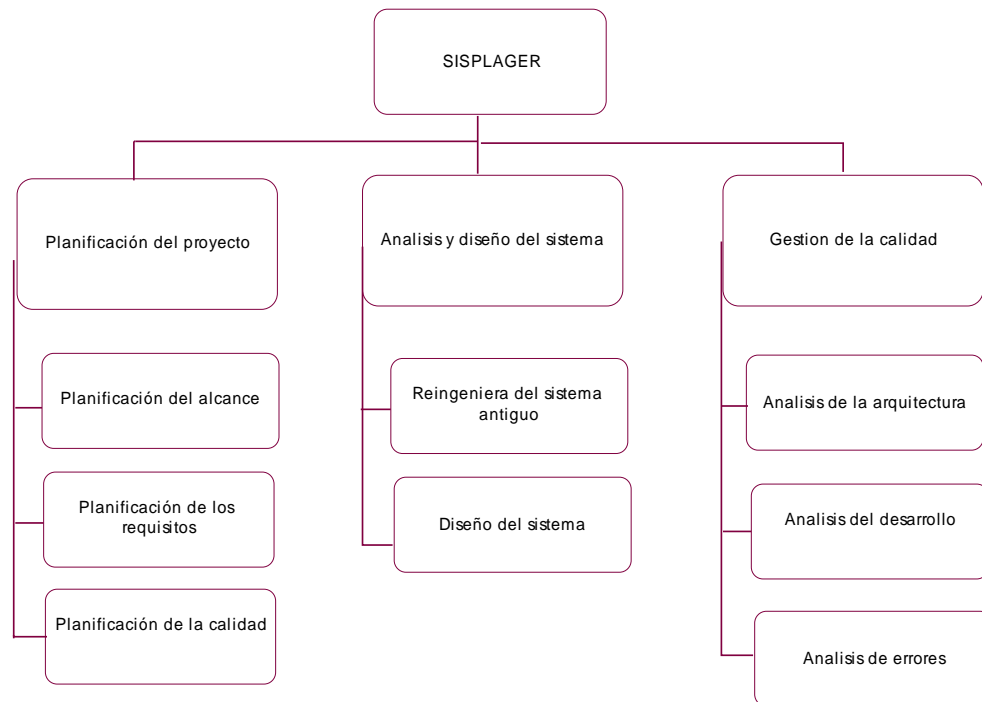


Figura 3.22 Estructura de Desglose del Trabajo

3.7 3. FASES DE PLANEACIÓN

	i	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1			Documentacion Balance Scord Card	5 días	lun 04/06/12	vie 08/06/12	
2			Documentacion Scrum	7 días	lun 11/06/12	mar 19/06/12	1
3			Documentacion EJB 3	7 días	mié 20/06/12	jue 28/06/12	2
4			Documentacion JSF 2	5 días	vie 29/06/12	jue 05/07/12	3
5			Documentacion Arquitectura Aplicación	5 días	vie 06/07/12	jue 12/07/12	4
6			Analisis de Requerimientos	20 días	vie 13/07/12	jue 09/08/12	5
7			Diagrama de Casos de Uso	5 días	vie 10/08/12	jue 16/08/12	6
8			Diagrama N-Tier	2 días	vie 17/08/12	lun 20/08/12	7
9			Diagrama Base Datos	15 días	mar 21/08/12	lun 10/09/12	8
10			desarrollo del Software	90 días	mar 11/09/12	lun 14/01/13	9
11			Conclusion y Recomendacions	5 días	mar 15/01/13	lun 21/01/13	10

Figura 3.23 Planificación del proyecto

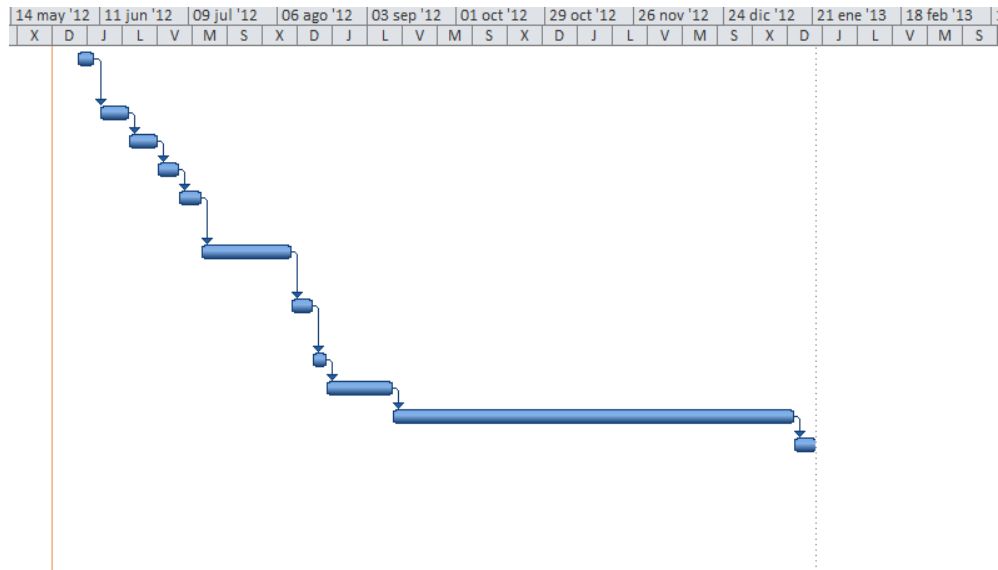


Figura 3.24 Diagrama de secuencia de la planificación del proyecto

3.8 INGENIERÍA INVERSA

3.8.1 MODELO CONCEPTUAL

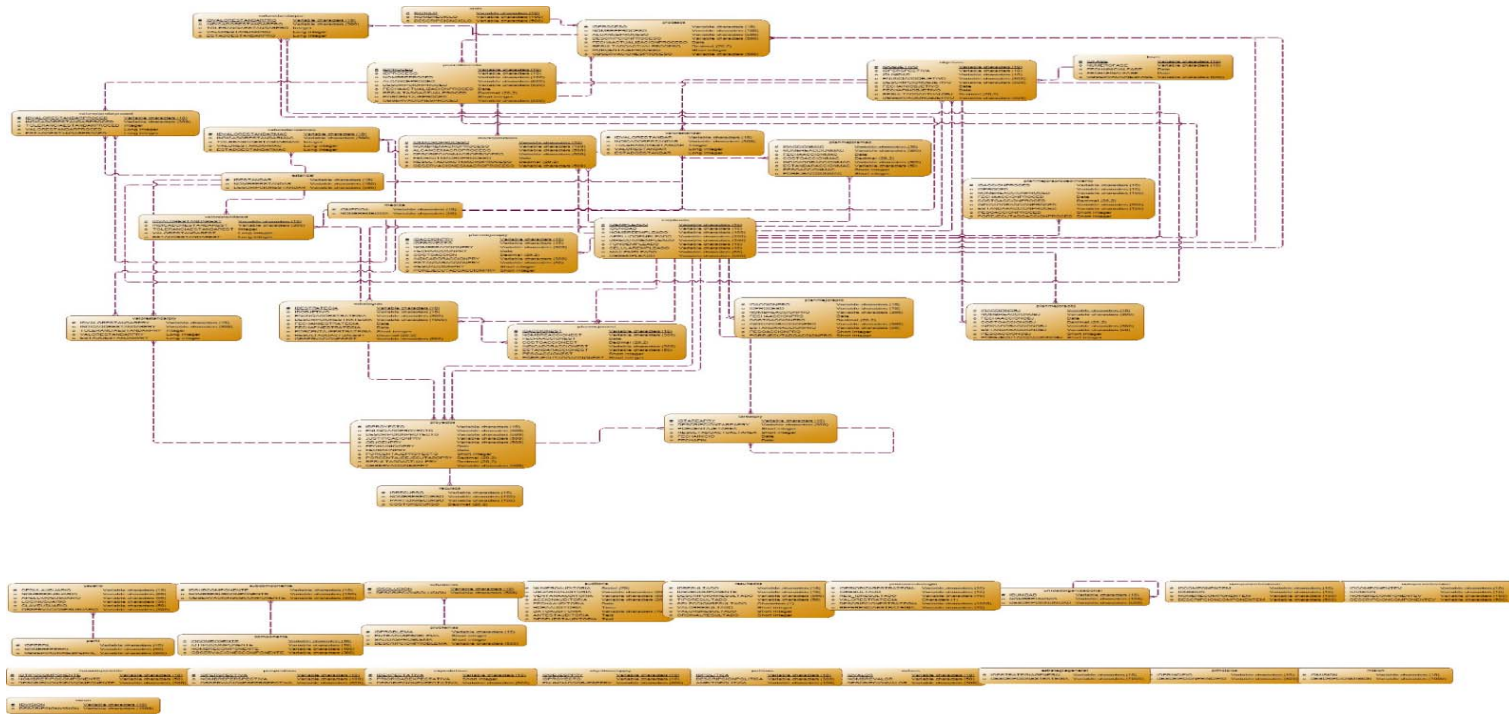


Figura 3.25 Modelo conceptual antiguo sistema

3.8.2 MODELO LÓGICO

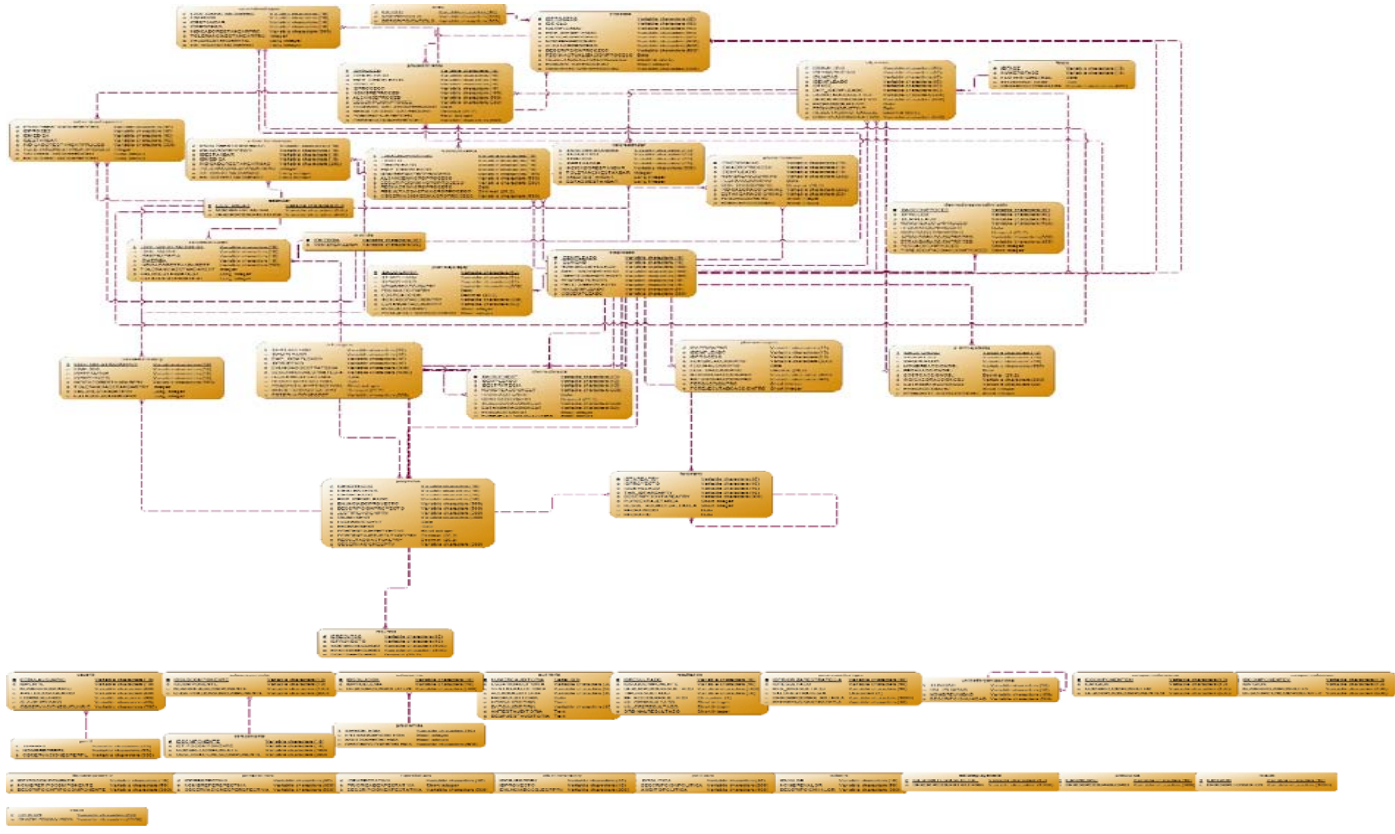


Figura 3.26 Modelo lógico antiguo sistema.

3.8.3 MODELO FÍSICO



Figura 3.27 Diagrama físico antiguo sistema.

3.9 DISEÑO DEL SISTEMA

3.9.1 DIAGRAMA INFRAESTRUCTURA

Se ha diseñado una arquitectura distribuida de hardware como muestra la Figura 3.28, donde se distingue claramente un sistema cuyos componentes se comunican y coordinan sus acciones a través de servicios.

El servidor de aplicaciones JEE esta definidos por capas como muestra la Figura 3.29 donde están alojados los frameworks necesarios para que puede desplegar una aplicación distribuida.

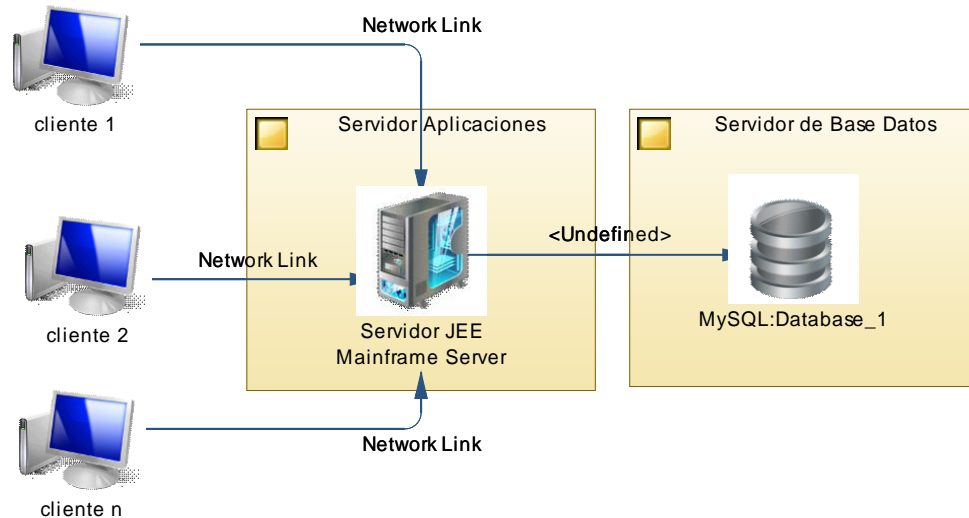


Figura 3.28 Arquitectura de Hardware SISPLAGER

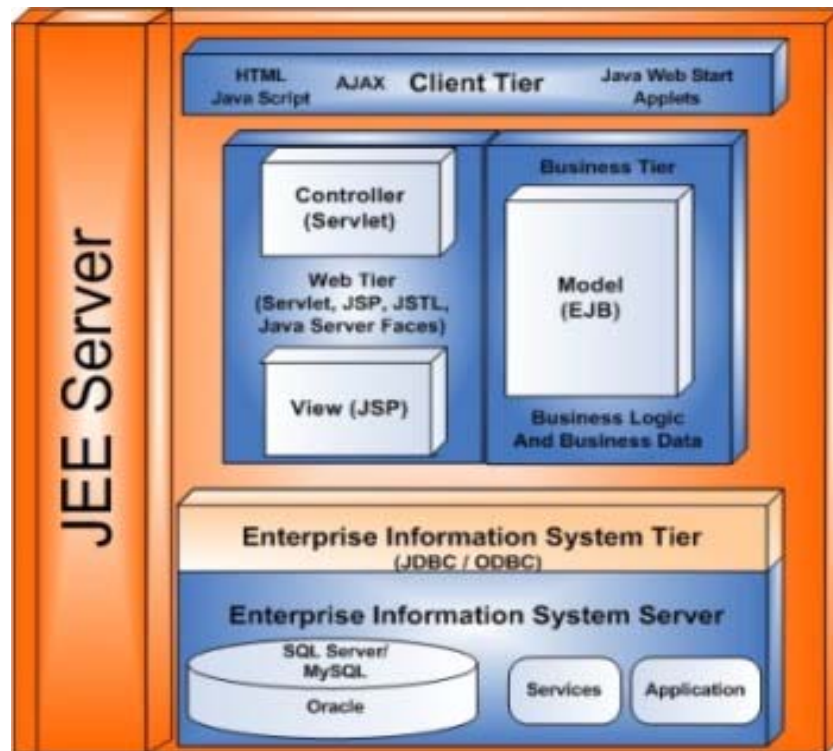


Figura 3.29 Arquitectura de Hardware SISPLAGER. Recuperada <http://docs.oracle.com/javaee/5/tutorial/doc/bnaay.html>

3.9.2 DIAGRAMA DE COMPONENTES

El servidor de aplicaciones se encuentra distribuido en subcomponentes como muestra la Figura 3.29 de la siguiente forma:

a) SisPlaGerWeb

Proyecto Web Java donde se alojan todas las páginas XHTML, JavaScript, CSS, estos elementos se organizan partes que sean independientes.

b) SisPlaGerRecursos

Proyecto Java que contiene plantillas para las paginas XHTML, imágenes y utilitarios para acceder a mensajes de usuario, configurar propiedades web.

c) SisPlaGerCliente

Proyecto Java el cual contiene un conjunto de librerías y APIs [10] que ofrecen acceso a ciertos servicios desde los procesos web.

d) SisPlaGerModelo

Proyecto Java el cual administra toda la lógica de negocio, transacciones, email y gestión de colas. Además este proyecto gestiona todo el proceso de la planificación gerencial.

e) ApplicationUtils

Proyecto Java que permite acceso a propiedades del sistema como en el entorno de desarrollo en el que se encuentra, configuración del idioma.

f) Entidades

Proyecto donde se alojan todas las entidades persistentes, las cuales representan las tablas de la base de datos.

g) QueryUtils

Proyecto Java que permite interpretar SQL a través de entidades persistentes, en el cual permite la extracción, inserción, actualización y eliminación de datos de una forma transparente.

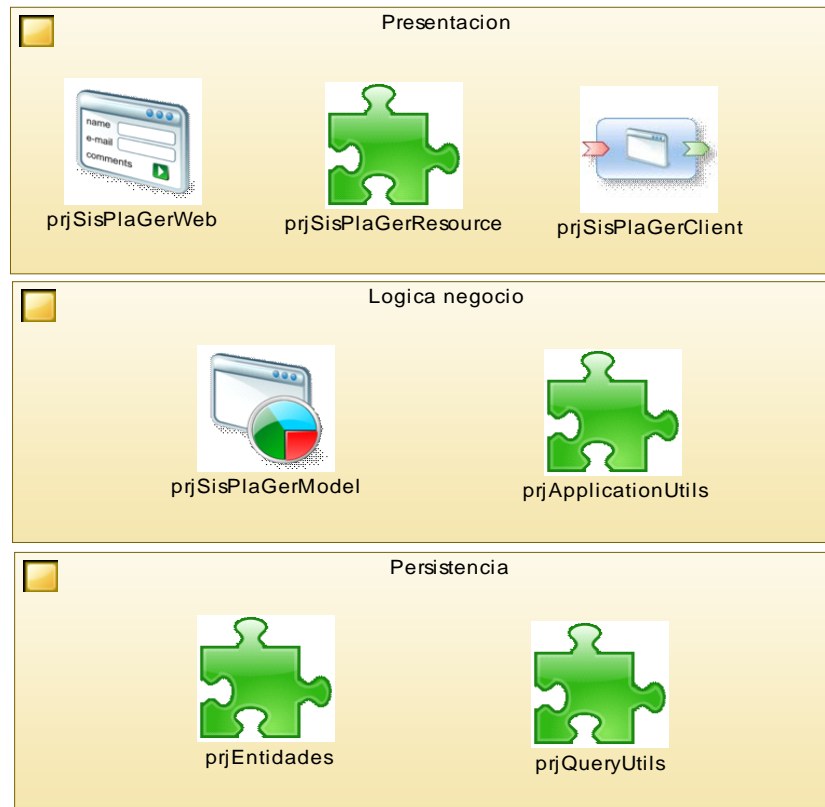


Figura 3.30 Diagrama de componentes SISPLAGER

3.10 DIAGRAMA DE LA BASE DE DATOS

3.10.1 MODELO LÓGICO

a) Usuarios y perfiles

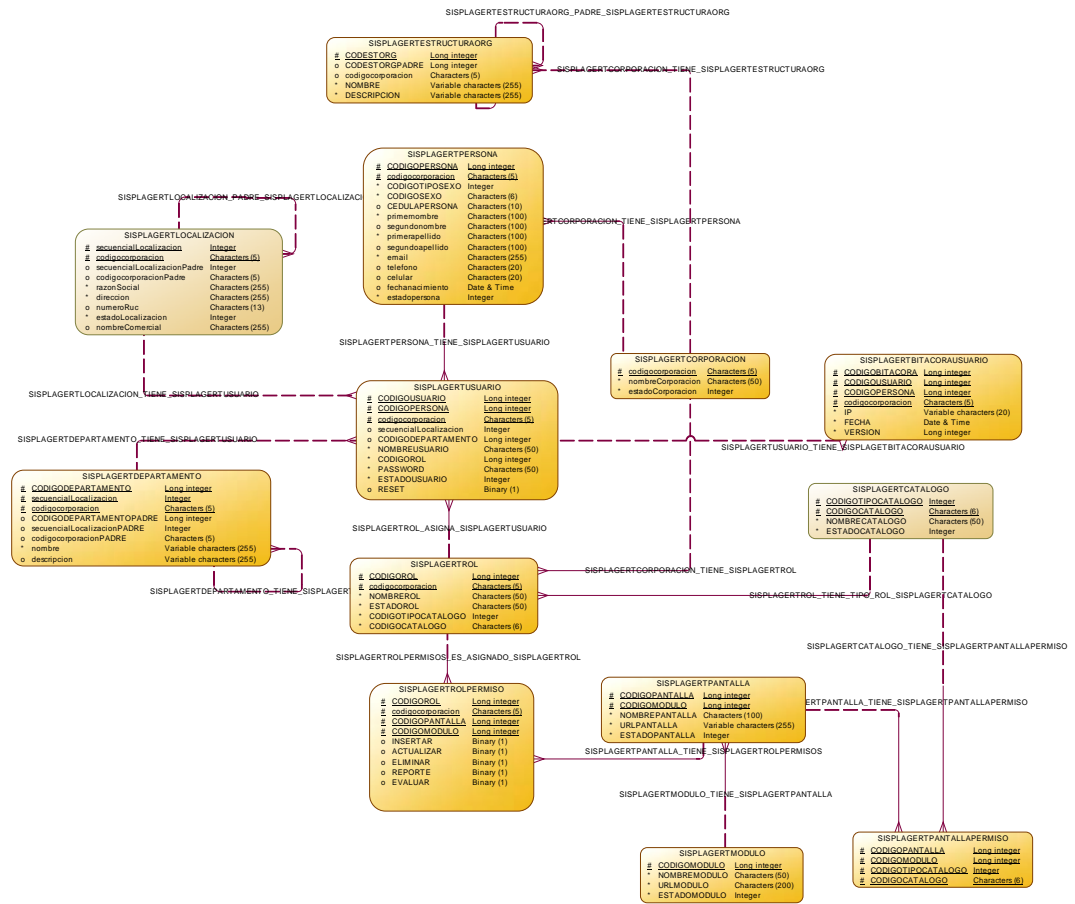


Figura 3.31 Modelo lógico usuarios y perfiles

c) Análisis ambiental

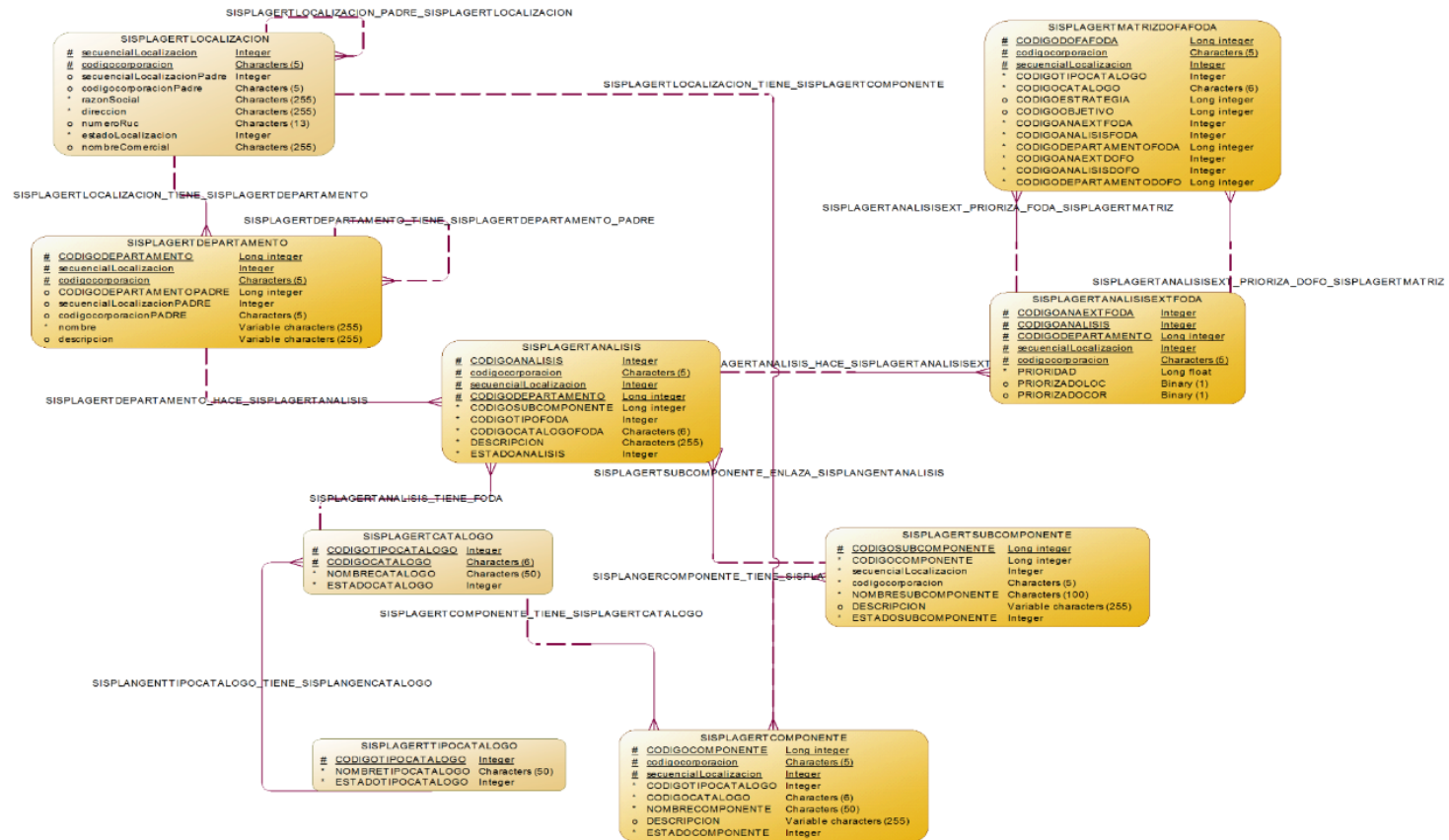


Figura 3.33 Modelo lógico análisis ambiental.

3.11 DIAGRAMA DE CLASES

3.11.1 USUARIOS Y PERFILES

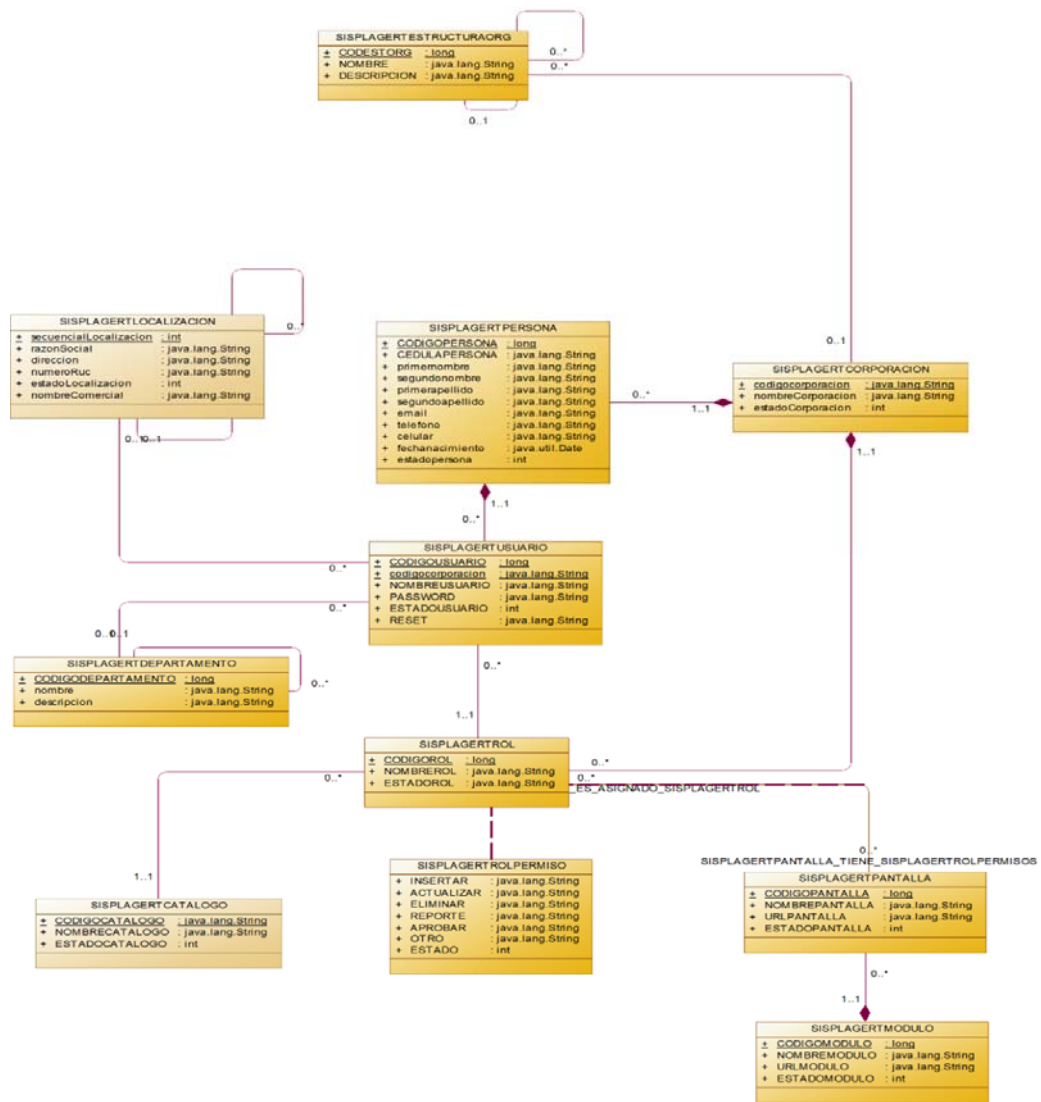


Figura 3.34 Modelo físico usuarios y perfiles

3.11.2 DIRECCIONAMIENTO ESTRATÉGICO

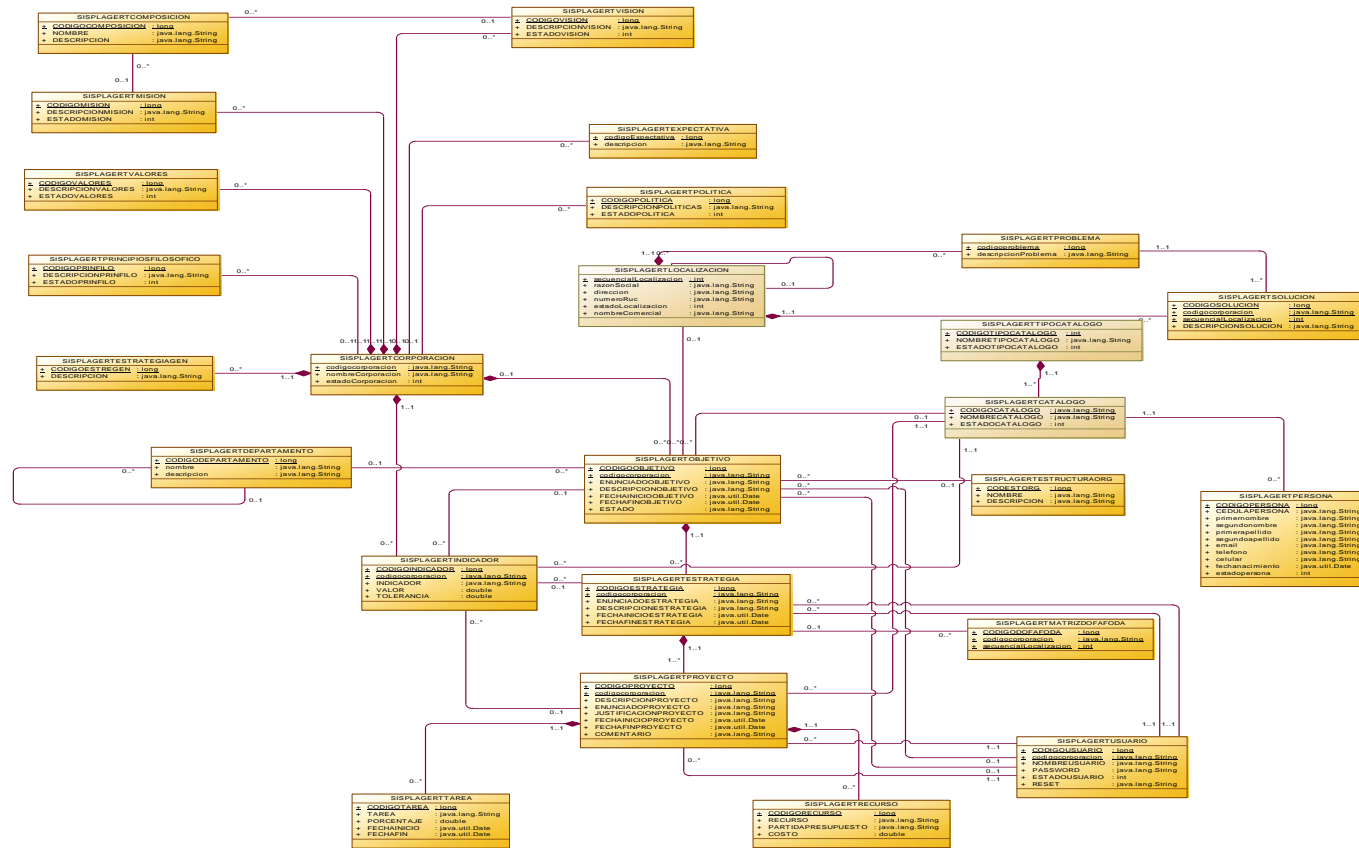


Figura 3.35 Modelo físico direccionamiento estratégico

3.11.3 ANÁLISIS AMBIENTAL

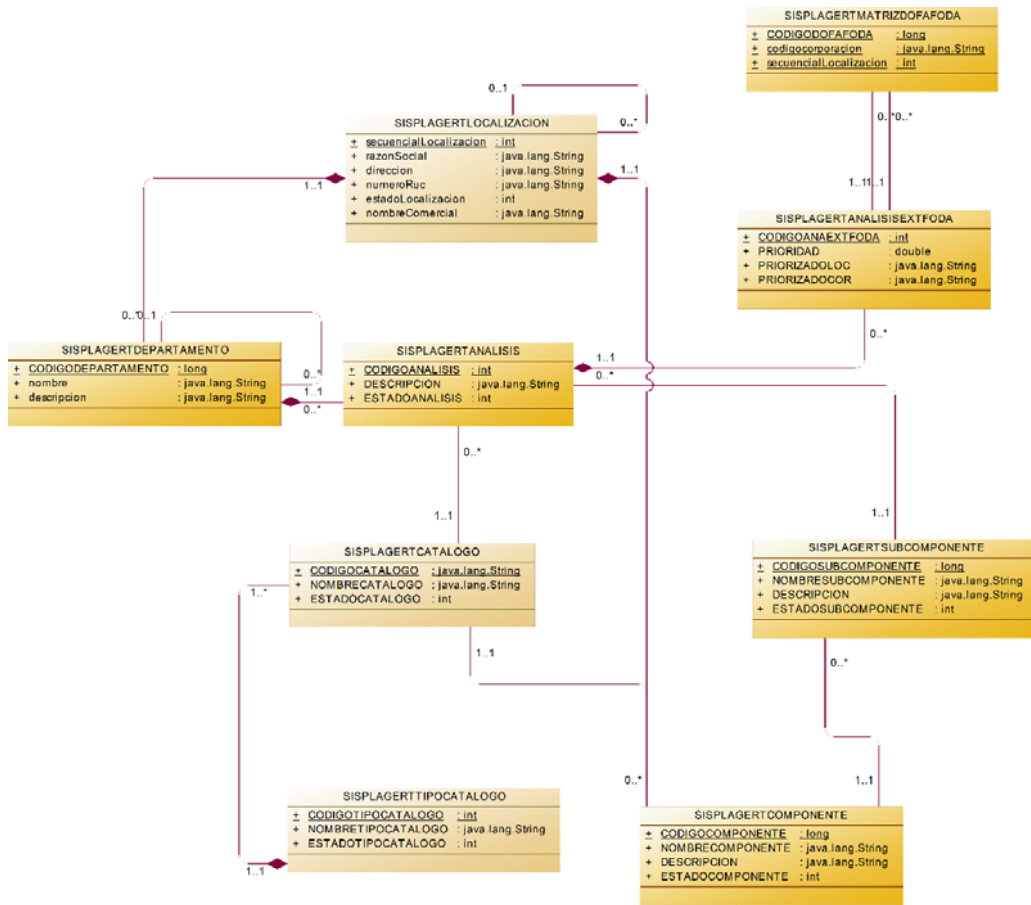


Figura 3.36 Modelo físico análisis ambiental

CAPÍTULO 4

DESARROLLO

4.1 BASE DE DATOS

Las bases de datos no son tan sólo una colección de archivos. Más bien, una base de datos es una fuente central de datos destinados a compartirse entre muchos usuarios para una diversidad de aplicaciones. El corazón de una base de datos lo constituye el sistema de administración de base de datos, el cual permite la creación, modificación y actualización de la base de datos, la recuperación de datos y la generación de informes y pantallas.

4.1.1 ESTÁNDARES

Los estándares de base de base de datos facilitan el desarrollo de un software para asegurar la calidad del mismo.

El uso de estándares de base de datos tiene innumerables ventajas, entre ellas:

- Asegurar la legibilidad del modelo de datos, inclusive para personas que no están relacionadas con el ambiente informático, en etapas de análisis y diseño.
- Facilitar la portabilidad entre motores de bases de datos, plataformas y aplicaciones;

- Facilitar la tarea de los programadores en el desarrollo de los sistemas.

4.1.2 NOMENCLATURA DBMS

En esta sección se presenta la nomenclatura definida para los distintos elementos de una base de datos.

4.1.3 BASE DE DATOS O SCHEMA

La base de datos o los Schemas deberán nombrarse usando la siguiente nomenclatura:

<NOMBRE SISTEMA>_<BD>: ejemplo: SISPLANGER_BD

<NOMBRE SISTEMA>_< SCHEMA >: ejemplo: SISPLANGER_ SCHEMA

a) Tablas

Las tablas deben nombrarse:

- En singular
- Sin utilizar espacios en blanco
- Si el nombre es compuesto solo la última palabra debe ir en plural
- Deben tener el prefijo T

<NOMBRE SISTEMA>< T ><NOMBRE DE LA TABLA>

b) Vistas

Las vistas deben nombrarse con la misma notación definida para nombrar tablas, pero prefijadas usando V.

<NOMBRE SISTEMA>< V ><NOMBRE VISTA>

c) Columnas

Los campos de una tabla corresponden a los atributos de una entidad, describen propiedades de la misma.

Las columnas deben ser nombradas según los lineamientos a continuación:

- Los nombres deben ser simples, representativos e intuitivos.
- Los nombres de las columnas de una tabla deben estar expresados en singular.

d) Foreign keys

Las Foreign keys son usadas para definir vínculos entre tablas relacionadas. Una Foreign key establece una relación entre una o más columnas de una tabla y la clave primaria de la tabla referenciada. Como patrón para la nomenclatura de la Foreign key se elige el siguiente.

FK_<TABLA_QUE_REFERENCIA>+<CAMPO_QUE_REFERENCIA>_<LA_REFERENCIADA>+<CAMPO_REFERENCIADO>

e) Índices

Los índices son un mecanismo para aumentar la eficiencia de localización y acceso de un registro en una tabla en la base de datos, opcionalmente asegurando unicidad de los valores del índice. La definición de índices tiene un impacto positivo en los tiempos de consulta de registro y uno negativo en los de inserción y actualización de los campos del índice.

Los índices están asociados a una tabla y a un conjunto de campos de la tabla, a su vez pueden ser únicos o no y pueden estar definidos en clúster o no. La nomenclatura elegida para nombrarlos es la siguiente:

[IDX_]<TABLA>_<CAMPO>

4.2 DISEÑO DE BASES DE DATOS

4.2.1 NORMALIZACIÓN

La normalización es la transformación de las vistas de usuario complejas y del almacén de datos a un juego de estructuras de datos más pequeñas y estables. Además de ser más simples y estables, las estructuras de datos normalizadas son más fáciles de mantener que otras estructuras de datos.

a) Primera forma normal (1NF)

El primer paso para normalizar una relación es remover los grupos repetitivos.

b) Segunda forma normal (2NF)

En la segunda forma normal, todos los atributos serán funcionalmente dependientes de la clave primaria. Por lo tanto, se remueve todos los atributos parcialmente dependientes y ponerlos en otra relación.

c) Tercera forma normal (3NF)

Una relación normalizada está en tercera forma normal si todos los atributos sin clave son funcionalmente dependientes por completo de la clave primaria y si no hay dependencias transitivas (sin claves).

4.3 ESTÁNDARES DE PROGRAMACIÓN

Este modelo de programación está basado en los estándares recomendados por Sun Microsystems, que han sido difundidos y aceptados ampliamente por toda la comunidad Java, y que han terminado por consolidarse como un modelo estándar de programación de facto.

4.3.1 NOMBRE DE LOS PAQUETES

Las clases en Java se agrupan en paquetes. Estos paquetes organizan de manera jerárquica, de forma que todo código desarrollado este organizado correctamente.

Java sugiere que la URL del sistema se lee de derecha a izquierda para representar los paquetes como muestra el siguiente ejemplo

URL: `www.sisplager.com.ec`

Paquete: `ec.sisplager`

4.3.2 CLASES

Los nombres de las clases deben de poseer un prefijo en conjunto con el nombre propio de la clase para representar el Core del sistema.

`<Nombre de la capa><Nombre>`.

4.3.3 CONSTANTES

Las constantes se dividen en globales y locales, el nombre de la constante posee un prefijo seguido del nombre de la constante definido por el consultor.

Los nombres de las constantes siempre se escribirán en mayúscula, además deben iniciar con un prefijo que encapsule el módulo o área del programa.

4.3.4 ESTÁNDARES DE FUNCIONES/MÉTODOS

Las funciones permiten acceder a los miembros de la clases, según la funcionalidad se debe escribir los métodos en infinito, además si los métodos son invocados por botones, eventos se debe poner el prefijo del componente que lo invoca.

4.4 PATRONES DE DISEÑO

Para construir aplicaciones JEE, de un modo profesional, se debe conocer técnicas avanzadas de análisis, diseño y desarrollo.

Se tiene una serie de patrones de diseño para ayudar a la construcción de las aplicaciones Java. Sin el conocimiento de algunos patrones más, el diseño y desarrollo de aplicaciones puede ser nefasto.

La labor de arquitecto de software es muy duro y complejo, fundamentalmente por el entorno tan cambiante dentro del mundo Java y las nuevas tecnologías en general.

4.4.1 PROXY

Se utiliza como intermediario para acceder a un objeto, permitiendo controlar el acceso a él.

Para ello obliga que las llamadas a un objeto ocurran indirectamente a través de un objeto proxy, que actúa como un sustituto del objeto original, delegando luego las llamadas a los métodos de los objetos respectivos.

4.4.2 SINGLETON

Está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

Se implementa creando en la clase un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con atributos como protegido o privado).

4.4.3 SERVICE LOCATOR

Utilizado para abstraer toda la utilización de JNDI (Interfaz de Nombres y Directorios de Java) y de la creación de cualquier contexto inicial de complejo,

además de para la encapsulación de búsqueda o creación de los objetos Home y EJB.

4.4.4 DATA ACCESS OBJECT

Crea un objeto que es un medio para el acceso a la Base de Datos de la aplicación, de forma que se abstraen y encapsulan todas las operaciones relacionadas con el tratamiento del acceso a datos.

4.5 DESCRIPCIÓN DE MÓDULOS

4.5.1 ANÁLISIS AMBIENTAL

a) Componentes

El módulo de componentes permite al usuario administrar, es decir ingresar, editar, eliminar o modificar un componente. Un componente en una organización son personas, grupos ya sean internas o externas con ciertas características específicas en la cual la organización enfoca y coordina sus esfuerzos productivos y comerciales para producir o para brindar un servicio a los clientes.

b) Subcomponentes

El módulo de subcomponentes permite la administración, es decir ingresar, editar, eliminar o modificar, además detecta si existen componentes creados para poder crear un sub componente.

Un subcomponente describe un proceso, técnicas, situación financiera, estructura organización, índices, tecnología, etc. de una organización.

c) Análisis interno

Este módulo permite al usuario administrar el análisis interno de la organización, donde puede ingresar, editar o eliminar una debilidad, fortaleza de la organización

El análisis interno comprende todas las fuerzas que se encuentran en una organización, es decir recursos internos únicos o particulares, dependiendo de su personal, situación financiera, tecnología, etc. para lograr cumplir un objetivo, además describe los factores en los cuales se tiene una posición desfavorable respecto a la competencia.

- Fortalezas: factores internos positivos que contribuyen al logro de los objetivos de la empresa
- Debilidades: factores internos negativos que inhiben el logro de los objetivos

d) Análisis externo

Este módulo permite al usuario administrar el análisis externo de la organización donde puede ingresar, editar o eliminar una amenaza, oportunidad de la organización.

El análisis externo comprende varios factores en el ambiente de una compañía que afectan la operación de la misma. Algunos de estos factores tienen un impacto positivo, mientras que otros pueden afectar negativamente.

Es más, el mismo factor puede influir de manera positiva en una compañía y negativamente en otra, dependiendo de la misión y objetivos de la compañía.

- Oportunidad: todas aquellas posibilidades externas a la empresa que tienen un impacto favorable en sus actividades.
- Amenaza: todas aquellas fuerzas externas a la empresa que pueden tener una influencia desfavorable en sus actividades.

e) Análisis FODA

Este módulo permite al usuario priorizar las amenazas, debilidades a través de la Matriz Relacional permitiéndole escoger las que obtuvieron la mayor cantidad de salidas, y las oportunidades y fortalezas con la Matriz de Holmes con el fin de obtener las que serán de mayor relevancia en la elaboración de estrategias.

f) Síntesis FODA

Este módulo permite al usuario visualizar la priorización de las oportunidades, fortalezas, debilidades y amenazas.

g) Expectativas

Este módulo permite al usuario la administración de expectativas, es decir ingresar, editar, eliminar o modificar. Una expectativa es una suposición en el futuro, puede o no ser realista.

h) Problemas

Este módulo permite al usuario la administración de problemas organizaciones, es decir ingresar, editar, eliminar o modificar.

4.5.2 DIRECCIONAMIENTO ESTRATÉGICO

a) Misión

Este módulo permite al usuario la administración de la misión, permitiéndole crear y editar. Además permite crear, editar, eliminar componentes de la misión, es decir preguntas que llevan a describir la misión.

La misión es la razón de ser de la empresa, el motivo por el cual existe, así mismo es la determinación de las funciones básicas que la empresa va a desempeñar en un entorno determinado para conseguir tal misión.

b) Visión

Este módulo permite al usuario la administración de la visión, permitiéndole crear y editar. Además permite crear, editar, eliminar componentes de la visión, es decir preguntas que llevan a describir la visión.

La declaración de la visión es el ideal de la organización para el futuro, indica lo que la organización le gustaría ser y como quiere ser vista.

c) Matriz DA

Este módulo permite priorizar las debilidades contra las amenazas a través de la Matriz Relacional calificando con Alta, Media y Baja, al finalizar la priorización se escoge solo las que obtuvieron la calificación Alta.

d) Matriz DO

Este módulo permite priorizar las debilidades contra las oportunidades a través de la Matriz Relacional calificando con Alta, Media y Baja, al finalizar la priorización se escoge solo las que obtuvieron la calificación Alta.

e) Matriz FA

Este módulo permite priorizar las amenazas contra las fortalezas a través de la Matriz Relacional calificando con Alta, Media y Baja, al finalizar la priorización se escoge solo las que obtuvieron la calificación Alta.

f) Matriz FO

Este módulo permite priorizar las fortalezas contra las oportunidades a través de la Matriz Relacional calificando con Alta, Media y Baja, al finalizar la priorización se escoge solo las que obtuvieron la calificación Alta.

g) Políticas generales

El módulo de políticas generales permite la administración, es decir ingresar, editar, eliminar o modificar.

La política de una organización es una declaración de principios generales que la empresa u organización se compromete a cumplir. En ella se dan una serie de reglas y directrices básicas acerca del comportamiento que se espera de sus empleados y fija las bases sobre cómo realizar las actividades de la empresa.

h) Principios filosóficos

El módulo de principios filosóficos permite la administración es decir ingresar, editar, eliminar o modificar. Un primer principio es un principio básico, una proposición fundamental que como tal no admite demostración a partir de principios más básicos, o no necesita demostración por ser auto-evidente. Los primeros principios son abstractos y generales, aplicándose a una gran cantidad de casos.

i) Soluciones

El módulo de soluciones permite la administración, es decir ingresar, editar, eliminar o modificar.

j) Valores institucionales

El módulo de valores institucionales permite la administración de valores institucionales, es decir ingresar, editar, eliminar o modificar un valor institucional.

Los valores institucionales son elementos de la cultura empresarial, propios de cada compañía. Son enunciados que guían el pensamiento y la acción de la gente en una Empresa, que se convierten en creencias arraigadas que definen comportamientos de las personas y les indica claramente pautas para la toma de decisiones.

k) Objetivos

El módulo de objetivos permite la administración de objetivos, es decir ingresar, editar, eliminar o modificar. Los objetivos que se crean desde el nivel más alto se crean con un estado activo, un objetivo también se crea a partir de una estrategia de un nivel superior con un estado pendiente de crear. Los objetivos que se crean a partir de una estrategia no se pueden eliminar, a diferencia de los objetivo del nivel superior. El objetivo creado asignara un responsable para que se cumpla, y un evaluador, demás se podrá administrar indicadores que permitan determinar el cumplimiento de objetivo.

Un objetivo es una meta o finalidad a cumplir para la que se disponen medios determinados. En general, la consecución de un determinado logro lleva implícita la superación de obstáculos y dificultades que pueden hacer naufragar el proyecto o, al menos, dilatar su concreción.

l) Estrategias

Este módulo permite la administración de estrategias, es decir ingresar, editar, eliminar o modificar. Las estrategias que se crean de un nivel superior, crean objetivos en niveles inferiores, para poder crear una estrategia obligadamente debe existir un objetivo. La estrategia creado asignara un responsable para que se cumpla, y un evaluador, además se podrá administrar indicadores que permitan determinar el cumplimiento de la estrategia.

Una estrategia se refiere al conjunto de acciones planificadas anticipadamente, cuyo objetivo es alinear los recursos y potencialidades de una empresa para el logro de sus metas y objetivos de expansión y crecimiento empresarial. Esta se hace evidente a través del modelo de negocio y las herramientas que facilitan su elaboración.

m) Proyectos

Este módulo permite la administración de proyectos, es decir ingresar, editar, eliminar o modificar. Un proyecto creado asignara un responsable para que se cumpla, y un evaluador, además se podrá administrar indicadores que permite determinar el cumplimiento de la estrategia, cronograma que permiten cumplir el proyecto y recurso necesarios para el proyecto.

Para que proyecto comience debe ser aprobado por un nivel superior, como también puede ser rechazado.

Un proyecto se define como la aplicación de conocimientos, herramientas y técnicas para encontrar una respuesta adecuada al planteamiento de una necesidad humana.

4.5.3 USUARIOS

Este módulo permite la administración de usuarios, es decir permite ingresar, editar, eliminar o modificar. Para poder crear un usuario es necesario que previamente este creado un perfil para poder asignar, además este módulo permite registrar los datos básicos de una persona.

4.5.4 PERFILES

Este módulo permite la administración de perfiles, es decir permite ingresar, editar, eliminar o modificar, donde el usuario asigna permisos tales como insertar, actualizar, eliminar, generar reporte a cada módulo del sistema

4.6 GESTIÓN DE LA CALIDAD

4.6.1 ANÁLISIS ARQUITECTURA

a) Sonar

Sonar es una excelente herramienta Open Source que tiene como objetivo analizar y medir la calidad de los proyectos. Para conseguir este propósito hace uso de una gran cantidad de métricas de calidad del software además de un amplio juego de reglas.

4.6.2 ANÁLISIS RENDIMIENTO

Es esta sección se muestra el tiempo de respuesta de los módulos, donde se mide el tiempo en segundos, el número de peticiones mostrando un gráfico de barras.

4.6.3 PRUEBAS DE UNIDAD

Nombre que reciben los procedimientos de pruebas locales a un módulo del sistema. Por definición dichas pruebas cubren la funcionalidad propia del módulo tanto con una perspectiva de caja blanca como de caja negra; pero prestando poca o ninguna atención a la integración con otros módulos.

Las pruebas unitarias se tienen que poder ejecutar sin necesidad de intervención manual. Esta característica posibilita que puedan automatizar su ejecución.

Las pruebas unitarias tienen que poder repetirse tantas veces como uno quiera. Por este motivo, la rapidez de las pruebas tiene un factor clave. Si pasar las pruebas es un proceso lento no se pasarán de forma habitual, por lo que se perderán los beneficios que éstas ofrecen.

4.6.4 PRUEBA CAJA BLANCA

La prueba de caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos.

Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de Caja Blanca como uno de los tipos de pruebas más importantes que se le aplican a los software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

4.6.5 PRUEBA DE CAJA NEGRA

Las pruebas de caja negra se centran principalmente en lo que “se quiere” de un módulo, charter o sección específica de un software, es decir, es una manera de encontrar casos específicos en ese modulo que atiendan a su especificación.

Las pruebas de caja negra se limitan a que el tester pruebe con “datos” de entrada y estudie como salen, sin preocuparse de lo que ocurre en el interior.

Lo más deseable a la hora de realizar pruebas de caja negra es realizar una cobertura completa, pero, en la mayoría de los casos no es suficiente, siempre hay que combinarlas con pruebas de integración, ya que por mucho que funcionen los datos de entrada/salida, por dentro o en terceros sistemas, pueden existir defectos que no se están teniendo en cuenta. Estos defectos pueden no acarrear problemas a corto plazo, pero a lo largo del tiempo aparecerán y como dicen...”es mejor el remedio que la enfermedad”.

4.6.6 PRUEBAS DE INTEGRACIÓN

Las Pruebas de Integración son aquellas que permiten probar en conjunto distintos subsistemas funcionales o componentes del proyecto para verificar que interactúan de manera correcta y que se ajustan a los requisitos especificados (sean estos funcionales o no).

Este tipo de pruebas deberán ejecutarse una vez se haya asegurado el funcionamiento correcto de cada componente implicado por separado, es decir, una vez se hayan ejecutado sin errores las pruebas unitarias de estos componentes.

La aplicación de este tipo de pruebas en un proyecto proporciona una serie de ventajas, especialmente relacionadas con la prevención de la aparición de errores ocasionados por:

- Un mal diseño de los interfaces de los componentes
- Un mal uso de los componentes.

Además, al igual que las pruebas unitarias, las pruebas de integración sirven como documentación del uso de los componentes puesto que en definitiva, de nuevo podrán considerarse como ejemplos de su uso.

4.6.7 PRUEBAS DE ACEPTACIÓN

En la ingeniería del software, los casos de prueba o Test Case son un conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio. Se pueden realizar muchos casos de prueba para determinar que un requisito es completamente satisfactorio. Con el propósito de comprobar que todos los requisitos de una aplicación son revisados, debe haber al menos un caso de prueba para cada requisito a menos que un requisito tenga requisitos secundarios. En ese caso, cada requisito secundario deberá tener por lo menos un caso de prueba. Algunas metodologías como RUP recomiendan el crear por lo menos dos casos de prueba para cada requisito. Uno de ellos debe realizar la prueba positiva de los requisitos y el otro debe realizar la prueba negativa.

Si la aplicación es creada sin requisitos formales, entonces los casos de prueba se escriben basados en la operación normal de programas de una clase similar.

Los casos de prueba escritos, incluyen una descripción de la funcionalidad que se probará, la cual es tomada ya sea de los requisitos o de los casos de

uso, y la preparación requerida para asegurarse de que la prueba pueda ser dirigida.

4.7 MANUAL USUARIO

Ver Anexo A

4.8 MANUAL TÉCNICO

Ver Anexo B

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

Al finalizar el sistema de planificación gerencial se puede indicar que se ha desarrollado con estándares de desarrollo de software, el resultado de la planificación, ejecución y pruebas se obtiene un sistema totalmente genérico para cualquier empresa que tenga una estructura organización bien definida.

5.1 CONCLUSIONES

- Los resultados obtenidos a través de las pruebas demuestran el cumplimiento de las historias de usuario y de su rendimiento que califican al software como Portable, Flexible, Integro.
- El conocimiento de frameworks web como Richfaces y Primefaces, ayuda al usuario final para que su experiencia al interactuar con el sistema sea satisfactoria.
- La versión anterior del sistema, permitió una traducción rápida de requerimientos para la construcción del nuevo sistema y facilitó la medición y prueba de la nueva versión.
- El diseño de usabilidad del sistema permite ofrecer una interfaz intuitiva y de fácil manejo para el usuario, considerando la complejidad intrínseca del proceso.

- Para el desarrollo de software es fundamental el interés que muestren los Stakeholders, puesto que éstos son los que determinan la calidad del software.

5.2 RECOMENDACIONES

- Utilizar el sistema desarrollado como software educativo, para la fase práctica, en asignaturas relacionadas con temas de planificación tanto estratégica como operativa.
- Es necesario cumplir adecuadamente las actividades de planificación del proyecto de ingeniería, con el apoyo de una metodología que permita asegurar la satisfacción de las necesidades del cliente proveyéndoles de software de calidad.
- Es recomendable utilizar estándares, tanto de gestión de proyectos como de desarrollo, para asegurar un adecuado mantenimiento adaptativo del sistema en el futuro.
- Utilizar el software construido en los procesos de planificación de la nueva estructura organizacional de la Universidad de la Fuerzas Armadas ESPE.

BIBLIOGRAFÍA

- BuenasTareas.com. (s.f.). *Análisis Y Validación De Requerimientos*.
Obtenido de <http://www.buenastareas.com/ensayos/Analisis-y-Validacion-De-Requerimientos/6124680.html>
- Burke, B., & Rubinger, A. (2010). *Enterprise JavaBeans 3.1*. O'Reilly Media.
- David, F. R. ((2003)). *Conceptos de Administración Estratégica*. México.: Pearson/Pentrice Hall.
- Debu panda, R. R. (March 2011). *EJB 3 in Action, Second Edition*.
- GEARY, D., & HORSTMANN, C. (2010). core JAVASERVER™ FACES.
- Gestopolis. (s.f.). Sistema ERP. Definición, funcionamiento, ventajas y desventajas.
- Jendrock, E., Cervera-Navarro, R., Evans, I., Gollapudi, D., Haase, K., Markito, W., & Srivathsa, C. (2013). *The Java EE 6 Tutorial*.
- Kendall, K., & Kendall, J. (2005). *Análisis y diseño de sistemas*.
- Kenneth , L., & Laudon, J. (s.f.). *Sistemas de información gerencial* . New York.

- Leatham , B., Fryc , L., & Rogers, S. (s.f.). *Developer Guide*. Obtenido de Developer Guide: <http://www.jboss.org/richfaces>
- Leatham, B., Fryc , L., & Rogers , S. (s.f.). *Component Reference*. Obtenido de Component Reference: http://www.jboss.org/richfaces/docs/4-0_comp_reference
- Leonard, A. (June 2010). *JSF 2.0 Cookbook*.
- Oracle. (2013). *The Java EE 6 Tutorial*. Obtenido de The Java EE 6 Tutorial: The Java EE 6 Tutorial
- Palacio, J., & Ruata , C. (Enero de 2011). *Scrum Manager Gestión de Proyectos*.
- Pham, A., & Pham, P.-V. (2012). *Scrum in action*. Boston.
- Red Hat, Inc. (05 de 2013 de 2013). *Hibernate Reference Documentation*. Obtenido de Hibernate Reference Documentation: <http://docs.jboss.org/hibernate/orm/4.3/manual/en-US/html/>
- Reese., R. M. (s.f.). *EJB 3.1 Cookbook*.
- Schwaber, K. (June 13, 2007). *The Enterprise and Scrum*.
- *Scrum Alliance*. (s.f.). Obtenido de <http://www.scrumalliance.org/why-scrum>

- Sicilia, M.-A. (8 de Jan de 2009). *Qué es la Ingeniería inersa*. Obtenido de Connexions : <http://cnx.org/content/m17432/latest/>
- *Wikipedia*. (19 de oct de 2013). Obtenido de [http://es.wikipedia.org/wiki/Subversion_\(software\)](http://es.wikipedia.org/wiki/Subversion_(software))

