



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

“ANÁLISIS COMPARATIVO DE TECNOLOGÍAS DE APLICACIONES WEB EN EL ENTORNO

JSF Y ADF. CASO PRÁCTICO: IESS DE RIOBAMBA - CHIMBORAZO”

TESIS DE GRADO

Previa a la obtención del título de

INGENIERO EN SISTEMAS INFORMÁTICOS

Presentado por:

PATRICIA ALEXANDRA VALDIVIEZO SERRANO

MARÍA ALEJANDRA GUACHO MINTA

RIOBAMBA-ECUADOR

2011

AGRADECIMIENTO

En el presente trabajo hemos plasmado todos los conocimientos adquiridos durante la carrera, queremos agradecer a Dios, ya que por su bendición hemos logrado nuestro objetivo.

También agradecemos a nuestros maestros quienes nos han sabido dirigir con su sabiduría.

En especial al Ing. Danilo Pastor y al Dr. Julio Santillán quienes siempre han estado pendientes y nos han sabido guiar con sus conocimientos y experiencias.

Queremos también agradecer a la ESPOCH, en la cual compartimos inolvidables momentos y recuerdos que permanecerán siempre con nosotros.

DEDICATORIA

La presente tesis va dedicada a mi querida madre,
por ser la protagonista principal de todos los logros
conseguidos en mi vida.

A la Familia Guacho Minta por el apoyo y cariño
incondicional que me han brindado, en especial a mi
sobrino Abdiel que es mi alegría.

A todas las personas que han confiado en mí y me
han brindado muchos consejos para salir adelante en
la vida, mis amigos que son como mi familia.

Alejandra G.

DEDICATORIA

Este trabajo va dirigido a mis padres
que con su infinito amor, han guiado mis pasos,
y me han dado la mano en aquellos malos
momentos.

Con cariño para ustedes

Patricia V.

NOMBRE	FIRMA	FECHA
Ing. Iván Ménes DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
Ing. Raúl Rosero DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS
Ing. Danilo Pástor DIRECTOR DE TESIS
Dr. Julio Santillán MIEMBRO DEL TRIBUNAL
Tigo. Carlos Rodríguez DIRECTOR CENTRO DE DOCUMENTACIÓN
NOTA DE LA TESIS	

“Nosotros, Patricia Alexandra Valdiviezo Serrano y María Alejandra Guacho Minta, somos responsables de las ideas, doctrinas y resultados expuestos en esta tesis; y, el patrimonio intelectual de la Tesis de Grado pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO”

Patricia Alexandra Valdiviezo Serrano

María Alejandra Guacho Minta

ÍNDICE DE ABREVIATURAS

- ADF:** Application Development Frameworks/ Faces
- BD:** Base de Datos
- CLI:** Common Language Infrastructure
- CLS:** Common Language Specification
- CLR:** Common Language Runtime
- CTS:** Common Type System
- CGI:** Common Gateway Interface
- COM / DCOM:** Component Object Model / Distributed COM
- ESPOCH:** Escuela Superior Politécnica de Chimborazo
- GPL:** General Public License
- GUI:** Interfaz Gráfica de Usuario
- HTML:** Lenguaje de Modelación de Hipertexto
- HTTP:** Hypertext Transfer Protocol
- IESS:** Instituto Ecuatoriano de Seguridad Social
- J2EE:** Java 2 enterprise Edition
- JAR:** Archivos Java
- JDK:** Java Development Kit (kit de desarrollo Java)
- JSF:** JavaServer Faces
- JSP:** JavaServer Pages
- JVM:** Java Virtual Machine (Máquina Virtual de Java)
- MYSQL:** My Structured Query Language
- MVC:** Modelo, Vista, Controlador
- RAD:** Rad application development (Desarrollo rápido de aplicaciones)
- SDK:** Software Development Kit (Kit de desarrollo de software)
- SADIESS:** Sistema de Activos y Deprecación para el IESS.
- UI:** Interfaz de Usuario
- XML:** Extended Markup lenguaje. (Lenguaje de marcado Extendido)

ÍNDICE GENERAL

PORTADA

AGRADECIMIENTO

DEDICATORIA

INDICE DE ABREVIATURAS

INDICE GENERAL

INDICE DE FIGURAS

INDICE DE TABLAS

INTRODUCCIÓN

CAPÍTULO I

MARCO REFERENCIAL 19

1.1	Antecedentes.....	19
1.2	Justificación Del Proyecto De Tesis	22
1.3	Problema	24
1.4	OBJETIVOS.....	24
1.4.1	OBJETIVO GENERAL	24
1.4.2	OBJETIVOS ESPECÍFICOS	25
1.5	Hipótesis	25

CAPITULO II

CONCEPTOS GENERALES DE LAS TECNOLOGIAS JAVA PARA APLICACIONES WEB 26

2.1	Entorno Web.....	26
2.1.1	Historia de las aplicaciones Web	26
2.1.2	Aplicaciones Web.....	28
2.1.2.1	Aplicaciones en capas.....	28
2.1.3	Fundamentos de la Web	30
2.1.4	Tecnología java	32
2.1.4.1	La plataforma Java	32
2.1.4.2	Librerías de Java	33
2.1.4.3	Lenguaje de programación Java	33
2.1.4.4	Introducción a los Java Servlets.....	34
2.1.4.5	Java Server Pages o JSP	35
2.2	Evolución de los Frameworks para componentes Web.....	36
2.2.1	Introducción a Los Frameworks de java	36
2.2.1.1	Que es un Framework.....	36

2.2.1.2	Patrón MVC	37
2.2.1.3	Tipos de Framework Web.	38
2.2.2	Componentes en java	39
2.2.2.1	JRE ("Java Runtime Environment")	40
2.2.2.2	JDK,SDK,J2SE.....	41
2.2.2.3	J2EE (Java 2 Enterprise Edition)	42
2.2.3	Plataforma y arquitectura de J2EE	43
2.2.3.1	Arquitectura J2EE	43
2.2.3.2	Componentes de J2EE	44
2.2.3.3	Containers J2EE.....	45
2.2.3.4	Servicios J2EE	46
2.2.3.5	Elementos de un sistema multi-nivel J2EE	47
2.2.3.6	Organización de un sistema multi-nivel bajo el meta patrón MVC	48
2.3	Servidor de Aplicaciones	50
2.3.1	Servidores de aplicación J2EE.....	50
2.3.2	Características comunes.....	51
2.3.3	Tipos de Servidores de aplicaciones	52
2.4	Estudio de las tecnologías JAVA en cuanto al desarrollo de aplicaciones Web. 55	
2.4.1	Tecnología JSF.	55
2.4.1.1	Características.....	55
2.4.1.2	Ventajas De JSF.....	56
2.4.2	Tecnología ADF	58
2.4.2.1	Características.....	58
2.4.2.2	Ventajas.....	58
2.4.2.3	Modelo Vista Controlador (MVC)	58
2.4.2.4	Ventajas del patrón	60
2.5 ADF	Selección de la herramienta IDE más adecuada para las tecnologías JSF y 61	
2.5.1	Entorno de desarrollo Integrado (IDE)	61
2.5.1.1	Componentes de un IDE	61
2.5.1.2	Software's IDE.....	62
2.5.2	Determinación del IDE	62
2.5.3	Análisis de las IDE's Seleccionadas	66
2.5.3.1	El IDE NetBeans.....	66
2.5.3.2	JDeveloper	69
2.6	Selección de la Base de Datos más adecuada para las tecnologías JSF y ADF 71	
2.6.1	Gestor de base de datos para JSF	71
2.6.2	Gestor de base de datos para ADF	73
2.6.3	Análisis del Gestor de base de datos Seleccionado.....	74

2.6.3.1 Base De Datos Mysql..... 74

CAPITULO III

ANÁLISIS COMPARATIVO DE TECNOLOGÍAS DE APLICACIONES WEB EN EL ENTORNO

JSF Y ADF..... 76

3.1 Introducción 76

3.2 Determinación De Las Tecnologías A Comparar 77

3.3 ANÁLISIS DE LAS TECNOLOGÍAS 80

3.3.1 JSF 80

3.3.1.1 Arquitectura 80

3.3.1.2 MVC con la tecnología JSF 81

3.3.1.3 JSF con su IDE..... 81

3.3.2 ADF 82

3.3.2.1 Arquitectura 82

3.3.2.2 MVC con la tecnología ADF 84

3.3.2.3 ADF con su IDE..... 86

3.4 DETERMINACIÓN DE LOS PARÁMETROS DE COMPARACIÓN. 87

3.4.1 Enumerar todos los parámetros posibles..... 87

3.4.2 Selección de los parámetros a comparar..... 88

3.4.3 Determinación de variables e indicadores a valorar 89

3.4.4 Descripción de los parámetros a valorar..... 89

3.4.5 Determinar la importancia de los parámetros 92

3.5 DESCRIPCIÓN DE LOS MÓDULOS DE PRUEBA..... 92

3.5.1 Modulo 1..... 92

3.5.2 Modulo 2..... 93

3.5.3 Modulo 3..... 93

3.6 DESARROLLO DE LOS MÓDULOS DE PRUEBA 93

3.6.1 DESARROLLO DE LOS MÓDULOS DE PRUEBA CON LA TECNOLOGÍA JSF 93

3.6.1.1 Modulo 1..... 93

3.6.1.2 Modulo 2..... 97

3.6.1.3 Modulo 3..... 99

3.6.2 DESARROLLO DE LOS MÓDULOS DE PRUEBA CON LA TECNOLOGÍA ADF 100

3.6.2.1 Modulo 1..... 100

3.6.2.2 Modulo 2..... 106

3.6.2.3 Modulo 3..... 108

3.7 ANÁLISIS CUALITATIVO Y CUANTITATIVO. 109

3.7.1 Escala de valoración cualitativa. 110

3.7.2 Evaluación del parámetro: Manejo del Patrón MVC 111

3.7.2.1 Variable: Modelo/ Acceso a Datos. 111

3.7.3.2 Variable: A nivel de Aplicación.....	120
3.7.4 Evaluación del parámetro: Producto.....	123
3.7.4.1 Variable: Madurez del Producto.....	123
3.7.4.2 Variable: Instalación.....	125
3.8 Puntajes Alcanzados.....	127
3.8.1 Tabla General de Comparación.....	128
3.8.2 Diagrama General De Resultados.....	129
3.9 Interpretación.....	129
3.10 Resultado del Análisis.....	129
3.11 Conclusiones.....	131

CAPITULO IV

DESARROLLO DEL SISTEMA DEL CONTROL DE ACTIVOS FIJOS Y SU DEPRECIACIÓN EN EL IESS DE CHIMBORAZO..... 132

4.1 INGENIERIA DE LA INFORMACIÓN.....	132
4.1.1 Definición del ámbito.....	132
4.1.2 Identificar Requerimientos.....	133
4.1.3 Estudio de Factibilidad.....	133
4.1.3.1 Factibilidad Económica.....	133
4.1.3.2 Factibilidad Técnica.....	133
4.1.3.3 Factibilidad Operativa.....	134
4.1.3.4 Factibilidad Legal.....	134
4.1.4 Especificación de Requerimientos.....	134
4.2 ANALISIS DEL SISTEMA.....	135
4.2.1 Casos de Uso del Sistema.....	135
4.2.2 Detalle de los Casos de Uso Identificados.....	136
4.2.2.1 Funcionalidad de los Casos de Uso.....	136
4.2.2.2 Diagrama de los Casos de Uso.....	138
4.2.3 Diagramas de Secuencia.....	138
4.2.4 Diagramas de Colaboración.....	141
4.3 DISEÑO.....	143
4.3.1 Casos de Uso Reales.....	143
4.3.2 Definición de Informes e Interfaces de Usuario.....	146
4.3.2.1 Definición de la información de la Interfaz de Usuario.....	146
4.3.2.2 Lenguaje de Comunicación.....	146
4.3.3 Diagramas de Interacción.....	147
4.3.3.1 Diagramas de Secuencia.....	147
4.3.3.1 Diagramas de Colaboración.....	150
4.3.3.3 Diagrama de Calles.....	151
4.3.4 Diagrama de Clases.....	152
4.3.5 Diagrama de Base de Datos.....	153

4.3.6	Diagrama de Despliegue.....	154
4.3.6.1	Diagrama de componentes	154
4.3.6.2	Diagrama de Nodos	154
4.4	Implementación y Pruebas	155
4.4.1	Definición de estándares de Programación	155
4.4.2	Pruebas Unitarias	155
4.4.3	Prueba de módulos y del Sistema	155
CAPITULO V		
ANALISIS DE RESULTADOS..... 157		
5.1	Estadística Inferencial.....	157
5.1.1	Selección Y Determinación De La Muestra	158
5.1.2	Obtención De Los Datos	158
5.1.3	Clasificación Y Organización De Los Datos.....	159
5.1.4	Análisis De Los Datos	160
5.1.5	Representación Gráfica De Datos	162
5.1.6	Validación De La Hipótesis	164
5.1.7	CONCLUSIONES.....	165
CONCLUSIONES		
RECOMENDACIONES		
RESUMEN		
SUMMARY		
ANEXOS		
BIBLIOGRAFIA		

INDICE DE FIGURAS

Figura I.1	Estructura del MVC en JSF	23
Figura II.2	Arquitectura tradicional en tres capas.....	29
Figura II.3	Arquitectura Web en tres capas	30
Figura II.4	Funcionamiento del protocolo HTTP	31
Figura II.5	Aplicación configurable.....	37
Figura II.6	Patrón Modelo-Vista-Controlador	38
Figura II.7	Patrón MVC	38
Figura II.8	Java 2, Standard Edition	42
Figura II.9	Arquitectura J2EE.....	43
Figura II.10	Arquitectura Multinivel.	49
Figura II.11	Arquitectura Web de tres niveles.....	50
Figura II.12	Interface de Usuario con JSF	56
Figura II.13	Estructura ADF	59
Figura II.14	NetBeans vs. otros IDE's.....	63
Figura II.15	Evolución de IDE's.....	65
Figura III.16	Pantalla de la encuesta sobre los frameworks	78
Figura III.17	Arquitectura de una aplicación JSF	80
Figura III.18	Arquitectura de ADF	82
Figura III.19	Bussiness Services.....	84
Figura III.20	Data Binding	84
Figura III.21	ADF Implementa el patrón MVC.....	85
Figura III.22	Panel con el driver de Mysql.....	94
Figura III.23	Conexión visual a una base de datos en Netbeans.....	94
Figura III.24	Ventana de aplicación de base de datos en NetBeans	94
Figura III.25	Inserción de juego de registro en NetBeans.....	95
Figura III.26	Pantalla de Consulta.....	95
Figura III.27	Resultado de la consulta	95
Figura III.28	Crear documento en NetBeans + JSF	96
Figura III.29	Figura Diseño y código	97
Figura III.30	Pantalla con los campos restringidos los espacios vacios	97
Figura III.31	Propiedades de la capa Controlador.	98
Figura III.32	Pantalla Final modulo 2	98
Figura III.33	Pantalla Verificando el inicio de sesión	99
Figura III.34	Pantalla Verificando la utilización de la versión	99
Figura III.35	Utilización del puerto en JSF	100
Figura III.36	Pantalla inicial del IDE NetBeans	100
Figura III.34	Ventana para crear una nueva aplicación	101
Figura III.35	Panel para conexión Visual a una base de datos utilizando JDeveloper.....	101
Figura III.36	Ventana para crear una nueva conexión.	102

Figura III.37	Conexión Visual a una base de datos	102
Figura III.38	Ventana de aplicación de base de datos	103
Figura III.39	Ventana de Insertar una nueva página View	103
Figura III.40	Ventana para ver la división del patrón MVC.....	104
Figura III.41	Crear una nueva página View utilizando la paleta de componentes	105
Figura III.42	Agregando elementos Imagen.....	105
Figura III.43	Paleta de Componentes	106
Figura III.44	Crear una nueva página View	107
Figura III.45	Componentes más Validaciones	107
Figura III.46	Conexión mediante código	108
Figura III.47	Pantalla final del modulo 2/Autenticación.	108
Figura III.48	Comparación Porcentual del Sub-Modulo Modelo	114
Figura III.49	Comparación Porcentual del Sub-Modulo Vista	117
Figura III.50	Comparación Porcentual del Sub-Modulo Controlador	120
Figura III.51	Comparación porcentual del parámetro Seguridades	123
Figura III.53	Diagrama General de Resultados	129
Figura IV.54	Diagrama de Casos de Uso General.....	138
Figura IV.55	Diagrama de Secuencia de Autenticación	139
Figura IV.56	Diagrama de Secuencia de Gestión de Información de Activos Fijos	139
Figura IV.57	Diagrama de Secuencia de Generar depreciación de Activos Fijos.....	140
Figura IV.58	Diagrama de Secuencia de Consultas.....	141
Figura IV.59	Diagrama de Colaboración de Autenticación.....	141
Figura IV.60	Diagrama de Colaboración de Gestión de Información de Activos Fijos	142
Figura IV.61	Diagrama de Colaboración de Generar depreciación de Activos Fijos.....	142
Figura IV.62	Diagrama de Colaboración de Consultas	142
Figura IV.63	Diagrama de caso de Uso de Autenticación.....	143
Figura IV.64	Diagrama de Caso de Uso de Gestionar Información de los Activos Fijos....	144
Figura IV.65	Diagrama de Caso de Uso de Generar Depreciación de los Activos fijos	145
Figura IV.66	Diagrama de Caso de Uso de Generar Consultas	146
Figura IV.67	Nivel Arquitectónico	147
Figura IV.68	Diagrama de Secuencia de Autenticación	147
Figura IV.69	Diagrama de Secuencia de Gestión de Información de Activos Fijos	148
Figura IV.70	Diagrama de Secuencia de Generar depreciación de Activos Fijos.....	149
Figura IV.71	Diagrama de Secuencia de Consultas.....	149
Figura IV.72	Diagrama de Colaboración de Autenticación.....	150
Figura IV.73	Diagrama de Colaboración de Gestión de Información de Activos Fijos	150
Figura IV.74	Diagrama de Colaboración de Generar depreciación de Activos Fijos.....	150
Figura IV.75	Diagrama de Colaboración de Consultas	151
Figura IV.76	Diagrama de Calles	151
Figura IV.77	Diagrama de Clases	152
Figura IV.78	Diagrama de Base de Datos.....	153

Figura IV.79	Diagrama de Componentes.....	154
Figura IV.80	Diagrama de Nodos.....	154
Figura V. 81	Grafico que representa los valores del Chi-Cuadrado.....	162
Figura V.82	Puntajes obtenidos de los enunciados	163
Figura V.83	Relación entre número de votantes y puntaje	164

INDICE DE TABLAS

Tabla I. I	Parámetros de evaluación.....	21
Tabla II.II	Características de un Framework	39
Tabla II.III	Tabulación de NetBeans vs. Otros IDE's.....	64
Tabla II.IV	Herramientas de desarrollo vs. JSF.....	64
Tabla II.V	IDE´s.....	66
Tabla II.VI	Tecnologías y Base de Datos Seleccionadas.....	74
Tabla III.VII	Tabulación de los frameworks.....	78
Tabla III.VIII	Parámetros posibles a comparar	87
Tabla III.IX	Parámetros a Comparar	88
Tabla III.X	Variables e Indicadores a Valorar	89
Tabla III.XI	Importancia de Parámetros a Comparar	92
Tabla III.XII	Análisis Cualitativo y Cuantitativo	110
Tabla III.XIII	Escala de valoración cualitativa.....	110
Tabla III.XIV	Equivalenciaentreelpuntaje deunavariabley suvaloracióncualitativa.....	110
Tabla III.XV	Valoraciones de la Variable 1 acceso a datos	112
Tabla III.XVI	Tabladevaloracionesde la VariableVista	115
Tabla III.XVII	Valoración de la Variable Controlador.....	118
Tabla III.XVIII	Tabla de valoración de la variable A nivel de aplicación.....	121
Tabla III.XIX	Valoraciones de la variable Madurez del producto	124
Tabla III.XX	Tabladevaloracionesde la variable Instalación	126
Tabla III.XXI	Tabla General de Comparación	128
Tabla IV.XXII	Recursos Humanos.....	134
Tabla IV.XXIII	Recursos Software	134
Tabla IV.XXIV	Usuarios.....	135
Tabla IV.XXV	Caso de Uso Autenticación.....	136
Tabla IV.XXVI	Caso de Uso Gestionar Información de los Activos Fijos.....	136
Tabla IV.XXVII	Caso de Uso Generar Depreciación de los Activos fijos	137
Tabla IV.XXVIII	Caso de Uso Generar Consultas.....	137
Tabla IV.XXIX	Caso de Uso Autenticación.....	143
Tabla IV.XXX	Caso de Uso Gestionar Información de los Activos Fijos.....	144
Tabla IV.XXXI	Caso de Uso Generar Depreciación de los Activos fijos	144
Tabla IV.XXXII	Caso de Uso Generar Consultas.....	145
TablaV. XXXIII	Variables a ser analizadas.....	158
TablaV. XXXIV	Respuestasdelosencuestados	159
Tabla V. XXXV	Valores observados.....	160
Tabla V. XXXVI	Valores Esperados	161
TablaV. XXXVII	Chi-cuadrado	161
Tabla V. XXXVIII	Número de votantes por enunciado	163

INTRODUCCIÓN

A medida que las Tecnologías de Aplicaciones Web evolucionan, se requiere obtener mejores ventajas y beneficios en el entorno de programación y de esa manera poder disminuir el tiempo de desarrollo y facilidad para la construcción de sistemas Web.

La tecnología Java, es una tecnología madura, extremadamente eficaz y sorprendentemente versátil, se ha convertido en un recurso inestimable y posee buenas perspectivas de futuro para el desarrollo de Aplicaciones Web.

Por tal razón hemos optado en realizar un análisis comparativo de las Tecnologías de aplicaciones Web en los entornos (JSF, ADF) y aplicar el entorno de desarrollo más productivo en la realización de un sistema Web.

El análisis comparativo entre las tecnologías ADF y JSF, nos permitió desarrollar el sistema Web del departamento de contabilidad del IESS con mejor rapidez y seguridad en la aplicación.

Debido a que los procesos manuales que realizan los empleados del Instituto Ecuatoriano de Seguridad Social de Riobamba-Chimborazo del Departamento de Contabilidad, ocasionan ciertos inconvenientes en cuanto a los responsables y la ubicación de los activos y entre otros, se vio la necesidad de automatizar dichos procesos mediante un sistema Web que maneja la gestión de los activos fijos y las depreciaciones.

El sistema Web de los Activos fijos desarrollado para el IESS – Departamento de Contabilidad brinda en la actualidad a los empleados la posibilidad de obtener detalles amplios sobre la situación de los activos que se encuentran en la Institución.

El contenido de esta tesis está estructurado en 5 capítulos, el **capítulo I** proporciona los antecedentes sobre el desarrollo de Aplicaciones Web en entornos Java, así como también detalla el contexto del análisis que se va realizando, el **capítulo II** realiza una descripción de los conceptos generales de las tecnologías Java para aplicaciones Web, el **capítulo III** presenta el análisis comparativo de las tecnologías de Aplicaciones Web en el entorno JSF y ADF a manera de cuadros comparativos y plasma los resultados obtenidos para finalizar **el capítulo IV** que se encarga de documentar el proceso de desarrollo del sistema de Gestión de Activos Fijos y la depreciación de los mismos en el IESS de Riobamba-Chimborazo, y de esta manera debemos poder realizar la comprobación de la Hipótesis la cual se refleja en el capítulo V.

CAPÍTULO I

MARCO REFERENCIAL

1.1 Antecedentes

A lo largo de los años han surgido distintas tecnologías de Java para el desarrollo Web, permitiendo a los desarrolladores un mejor desenvolvimiento y facilitando las tareas que se encontraban en sus manuales de procesos.

Años atrás, para la construcción de aplicaciones Web un desarrollador recurría a simples editores de textos que dificultaban su trabajo, ya que existía un proceso complejo de programación y tiempo para el desarrollo de una aplicación Web.

Debido a esto, surge la necesidad de analizar alternativas que permitan resolver problemas en cuanto al desarrollo de aplicaciones Web, así como el diseño de formularios, acceso a base de datos, manejo de seguridades y otros que complementan los ambientes de desarrollo Web. Y de esta manera permitir al desarrollador Web agilizar y optimizar su trabajo.

Para el desarrollo de aplicaciones Web en JAVA se hace uso de las tecnologías JSF y ADF, las mismas que se define de la siguiente manera:

JSF (Java Server Faces): Es un Framework que simplifica el desarrollo de aplicaciones Web. Es la evolución de los Struts y es un componente del Framework de Java 2 Enterprise Edition (J2EE).

Características

- ✓ Utiliza un sencillo fichero de configuración para el controlador en formato xml.
- ✓ Simplifica el desarrollo de JSP: Basado en componentes.
- ✓ Es extensible, pudiendo crearse nuevos elementos de la interfaz o modificar los ya existentes.
- ✓ Escribe menos HTML: Lenguaje de marcado de componentes

Una Aplicación JSF es Conjunto de:

- ✓ Páginas JSP.
- ✓ Tags personalizados para representan objetos configurables de la página (opcional).
- ✓ Responsables de valores de los componentes gráficos.
- ✓ Listener de eventos.
- ✓ Archivo de configuración de recursos.
- ✓ Reglas de navegación.
- ✓ Descriptor de deployment (web.xml).

ADF (Application Development Framework): Esta basado en el patrón de diseño (MVC) y en las mejores prácticas de J2EE. El patrón de diseño MVC separa la arquitectura de la aplicación en tres capas.

Características

- ✓ La programación de esta Tecnología se basa en la técnica arrastrar y soltar.
- ✓ Contiene un conjunto de componentes inteligentes de software que cooperan entre sí.
- ✓ Frameworks de desarrollo basados en estándares integrados con Herramientas + Tecnología Web 2.0, AJAX.
- ✓ Reduce la necesidad de escribir código.

ANÁLISIS

Dentro del análisis de las tecnologías de aplicaciones Web JSF y ADF se encuentran definidos ciertos parámetros que nos permitirán enmarcarnos en el desarrollo del sistema Web.

Tabla I. I Parámetros de evaluación

PARAMETROS
Patrón de diseño MVC
Seguridad
Producto

Es importante tomar en cuenta estos parámetros, y de esta manera centrarnos en el estudio de las tecnologías JSF y ADF dando a conocer las técnicas que usan en su entorno de desarrollo Web.

DELIMITACIÓN

Lugar de Aplicación: “Instituto Ecuatoriano de Seguridad Social de Riobamba-Chimborazo Regional Centro- Dirección General-Departamento de Contabilidad”

Alcance

Este tema de tesis se enmarca en el análisis comparativo de las tecnologías de aplicaciones Web en entornos JSF y ADF en base a los parámetros establecidos en la [Tabla I.] lo cual permitirá desarrollar una aplicación Web para el control de inventarios y depreciación de activos fijos en el IEES.

Limitación

Para la parte aplicativa nos limitaremos en los siguientes módulos: El sistema Web podrá visualizar:

- ✓ Control de Inventarios
 - Gestión de asignaciones de los bienes al personal de la Dirección General del IEES.

- ✓ Depreciación de activos Fijos
 - Registro de bienes y sus detalles.
 - Calculo de la depreciación por el método de la línea recta.
 - Resúmenes Generales por año, mes y día de bienes depreciados.
 - Valor por depreciarse un bien.
 - Listado de bienes que han sido dados de baja.

1.2 Justificación Del Proyecto De Tesis

JUSTIFICACIÓN TEÓRICA

A medida que las Tecnologías de aplicaciones Web evolucionan, se requiere obtener mejores ventajas y beneficios en el entorno de programación y de esa manera poder disminuir el tiempo de desarrollo y facilidad para la construcción del sistema Web.

La tecnología Java, es una tecnología madura, extremadamente eficaz y sorprendentemente versátil, se ha convertido en un recurso inestimable y posee buenas perspectivas de futuro para el desarrollo de Aplicaciones Web.

Los parámetros que se encuentran en la [**Tabla I.I**] son tomados en cuenta debido a la necesidad de seguridad que se requiere en las páginas Web dinámicas, pues estas deben interactuar con una base de datos que requiere del uso de una interfaz grafica amigable que brinde confianza a los usuarios.

Por tal razón hemos optado en realizar un análisis comparativo de las Tecnologías de aplicaciones Web JSF y ADF ya que los procesos manuales que se realiza en el departamento de contabilidad del IESS, podrían ser solucionados mediante la automatización de los activos fijos y el control de inventarios.

JSF

Java Server Faces permite simplificar el desarrollo de aplicaciones Web, es un Framework que permite manejar el estado de los componentes, procesar los datos, validar la entrada del usuario, y manejar eventos.

Ventajas

- ✓ Separación de la presentación y el comportamiento del patrón de Diseño MVC.

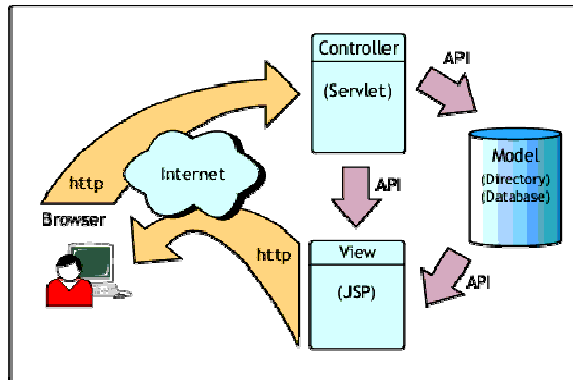


Figura I.1 Estructura del MVC en JSF

- ✓ Separación de roles, eliminación de la complejidad en el desarrollo de la interface de usuario y división de tareas.
- ✓ Permite construir aplicaciones Web que implementan una separación entre el comportamiento y la presentación tradicionalmente ofrecida por arquitectura de la interface de Usuario del lado del cliente.
- ✓ Estandarización: Los más grandes proveedores de herramientas de desarrollo (Sun, ORACLE, IBM, Apache) colaboran con su desarrollo y mantenimiento.
- ✓ La interface de usuario con componentes reutilizables y extensibles.

ADF

El ecosistema de desarrollo Web es un mundo con múltiples opciones, para cada escenario habrá un pequeño subconjunto de opciones que son las recomendadas para cada caso.

Ventajas

- ✓ A grandes rasgos trata de separar el modelo, de la vista y del controlador del patrón de diseño MVC.
- ✓ En caso de aplicaciones destinadas a internet, sólo será necesario usar un subconjunto del framework ADF para lograr los resultados esperados, pero pudiendo usar el mismo equipo de desarrollo para todo tipo de aplicaciones, y por tanto ganando en flexibilidad y rentabilizando al máximo la inversión.

JUSTIFICACIÓN PRÁCTICA

Se ha tomado como parte práctica la realización de un sistema Web para el IESS, puesto que para un mejor servicio y crecimiento de dicha institución es necesario contar con medios tecnológicos que agilicen los procesos que se ejecutan en el mismo.

El Instituto Ecuatoriano de Seguridad Social de Chimborazo tiene un número considerable de aplicaciones bajo el entorno de Java, lo cual nos brinda las facilidades respectivas de desarrollar en ambientes ADF y JSF, pues incluso actualmente esta institución cuenta con el servidor de aplicaciones JBoss,

1.3 Problema

En el Departamento de contabilidad de la dirección general del IEES de Riobamba-Chimborazo Regional Centro se han presentado ciertos problemas en **tiempo, recurso y eficiencia** de los procesos, debido a que algunos trámites se dan de forma manual y otras están realizadas en programas que no son apropiadas, dificultando así las actividades que se llevan a cabo.

Actualmente los procesos de Control de Inventarios y la Depreciación de activos fijos están realizados en Microsoft Excel y otras se realizan manualmente ocasionando inconformismo en los empleados de la Institución, es por ello que requieren automatizar estos procesos mediante una aplicación Web.

Solución

Automatizar los procesos descritos en el problema para obtener calidad mediante el uso de un sistema Web para el control de inventarios y la depreciación de activos fijos así como el registro del bien y de todos sus detalles para que estos puedan ser utilizados y visualizados mediante la Intranet. Generando así reportes en base a los permisos otorgados por el administrador en el IESS de Chimborazo.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Realizar un análisis comparativo de las tecnologías de aplicaciones Web en los entornos (JSF, ADF) y aplicar el entorno de desarrollo más productivo en la realización de un sistema Web para el control de inventarios y la depreciación de activos fijos en el IESS de Riobamba-

Chimborazo Regional Centro-Dirección General del Departamento de Contabilidad.

1.4.2 OBJETIVOS ESPECÍFICOS

- ✓ Estudiar los conceptos sobre las tecnologías de Java JSF y ADF para Desarrollo Web.
- ✓ Establecer variables e indicadores para los parámetros: Patrón de diseño MVC, Seguridad y Producto.
- ✓ Realizar el análisis comparativo entre las tecnologías de Java JSF y ADF mediante la creación módulos para su caracterización en base a los parámetros establecidos.
- ✓ Definir el entorno de desarrollo óptimo para la construcción de la aplicación, con los resultados obtenidos del análisis comparativo.
- ✓ Desarrollar el sistema Web del Control de Inventarios y Depreciación de activos fijos en el “Departamento de contabilidad de la Dirección General del IESS.

1.5 Hipótesis

El análisis comparativo entre las tecnologías ADF y JSF, para el desarrollo Web permitirá desarrollar el sistema Web del departamento de contabilidad del IESS con mejor rapidez y seguridad en la aplicación.

CAPITULO II

CONCEPTOS GENERALES DE LAS TECNOLOGIAS JAVA PARA APLICACIONES WEB

2.1 Entorno Web

2.1.1 Historia de las aplicaciones Web

Inicialmente la Web era simplemente una colección de páginas estáticas, documentos, etc., que podían consultarse o descargarse.

El siguiente paso en su evolución fue la inclusión de un método para confeccionar páginas dinámicas que permitiesen que lo mostrado fuese dinámico (generado o calculado a partir de los datos de la petición). Dicho método fue conocido como CGI (common Gateway interface) y definía un mecanismo mediante el cual podíamos pasar información entre el servidor HTTP y programas externos. Los CGI siguen siendo muy utilizados, puesto que la mayoría de los servidores Web los soportan debido a su sencillez. Además, nos proporcionan total libertad a la hora de escoger el lenguaje de programación para desarrollarlos.

El esquema de funcionamiento de los CGI tenía un punto débil: cada vez que recibíamos una petición, el servidor Web lanzaba un proceso que ejecutaba el programa CGI. Como, por otro

lado, la mayoría de CGI estaban escritos en algún lenguaje interpretado (Perl, Python, etc.) o en algún lenguaje que requería un entorno de ejecución (VisualBasic, Java, etc.), esto implicaba una gran carga para la máquina del servidor. Además, si la Web tenía muchos accesos al CGI, esto suponía problemas graves.

Por ello se empiezan a desarrollar alternativas a los CGI para solucionar este grave problema de rendimiento. Las soluciones vienen principalmente por dos vías. Por un lado se diseñan sistemas de ejecución de módulos más integrados con el servidor, que evitan que éste tenga que instanciar y ejecutar multitud de programas. La otra vía consiste en dotar al servidor de un intérprete de algún lenguaje de programación (RXML, PHP, VBScript, etc.) que nos permita incluir las páginas en el código de manera que el servidor sea quien lo ejecute, reduciendo así el tiempo de respuesta.

A partir de este momento, se vive una explosión del número de arquitecturas y lenguajes de programación que nos permiten desarrollar aplicaciones Web. Todas ellas siguen alguna de las dos vías ya mencionadas. De ellas, las más útiles y las que más se utilizan son aquellas que permiten mezclar los dos sistemas, es decir, un lenguaje de programación integrado que permita al servidor interpretar comandos que “incrustemos” en las páginas HTML y un sistema de ejecución de programas más enlazado con el servidor que no presente los problemas de rendimiento de los CGI.

Continuando con, Sun Microsystems con su sistema Java, que está integrada por dos componentes; a saber, un lenguaje que permite incrustar código interpretable en las páginas HTML y que el servidor traduce a programas ejecutables, JSP (Java server pages) y un mecanismo de programación estrechamente ligado al servidor, con un rendimiento muy superior a los CGI convencionales, llamado Java Servlet.

2.1.2 Aplicaciones Web

Se denomina **Aplicación Web** a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor Web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores Web (HTML, JavaScript, Java, asp.net, etc.) en la que se confía la ejecución al navegador.

Las aplicaciones Web son populares debido a lo práctico del navegador Web como cliente ligero, así como a la facilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los Web mails, wikis, Weblogs, tiendas en línea y la propia Wikipedia que son ejemplos bien conocidos de aplicaciones Web.

Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

2.1.2.1 Aplicaciones en capas

La estrategia tradicional de utilizar aplicaciones compactas causa gran cantidad de problemas de integración en sistemas de aplicaciones complejos como pueden ser los sistemas de gestión de una empresa o los sistemas de información integrados consistentes en más de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad e integración. Para solventar estos problemas se ha generalizado la división de las aplicaciones en capas que normalmente serán tres: una capa que servirá para guardar los datos (base de datos), una capa para centralizar la lógica de negocio (modelo) y por último una interfaz gráfica que facilite al usuario el uso del sistema (presentación).

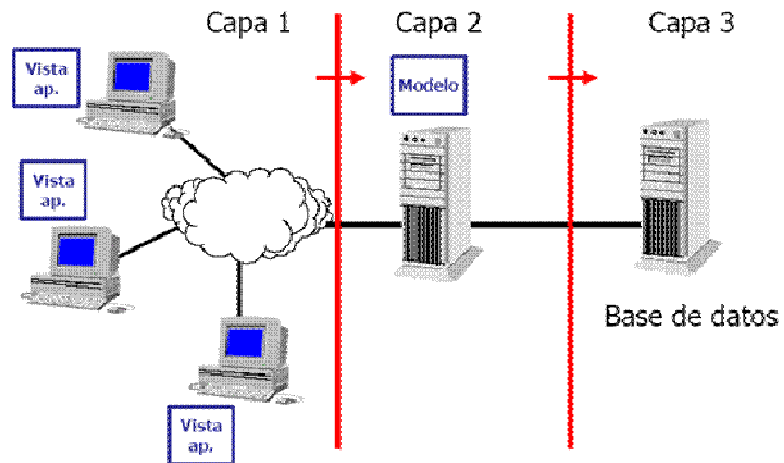


Figura II.2 **Arquitectura tradicional en tres capas**

Si se establece una separación entre la capa de interfaz gráfica (cliente), replicada en cada uno de los entornos de usuario, y la capa del modelo, que quedaría centralizada en un servidor de aplicaciones, según el diagrama que podemos ver en la Figura 1.1, obtenemos una potente arquitectura que otorga algunas ventajas:

- ✓ Centralización de los aspectos de seguridad y transaccionalidad, que serían responsabilidad del modelo.
- ✓ No replicación de lógica de negocio en los clientes: esto permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costos de mantenimiento.
- ✓ Mayor sencillez de los clientes.

Si intentamos aplicar esto a las aplicaciones Web, debidas a la obligatoria sencillez del software cliente que será un navegador Web, nos encontramos con una doble posibilidad:

Crear un modelo de cuatro capas, separando cliente, servidor Web, modelo y almacén de datos. Esto permite una mayor extensibilidad en caso de que existan también clientes no Web en el sistema, que trabajarían directamente contra el servidor del modelo.

Sin embargo, la gran mayoría de las aplicaciones Web comunes utilizan una arquitectura basada en la de tres capas extendida a las particularidades de la Web.

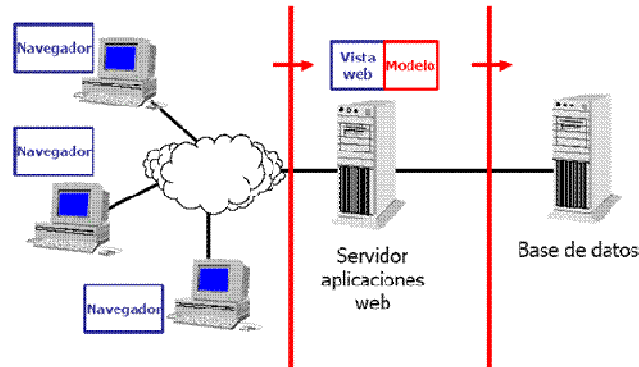


Figura II.3 **Arquitectura Web en tres capas**

2.1.3 Fundamentos de la Web

El éxito espectacular de la Web se basa en dos puntales fundamentales:

El protocolo HTTP y el lenguaje HTML. Uno permite una implementación simple y sencilla de un sistema de comunicaciones que nos permite enviar cualquier tipo de ficheros de una forma fácil, simplificando el funcionamiento del servidor y permitiendo que servidores poco potentes atiendan miles de peticiones.

El otro nos proporciona un mecanismo de composición de páginas enlazadas simple y fácil, altamente eficiente y de uso muy simple.

El protocolo HTTP

El protocolo HTTP (*hypertext transfer protocol*) es el protocolo base de la WWW. Se trata de un protocolo simple, orientado a conexión y sin estado. La razón de que esté orientado a conexión es que emplea para su funcionamiento un protocolo de comunicaciones (TCP, *transport control protocol*) de modo conectado.

Existe una variante de HTTP llamada HTTPS (S por *secure*) que utiliza el protocolo de seguridad SSL (*secure socket layer*) para cifrar y autenticar el tráfico entre cliente y servidor,

siendo ésta muy usada por los servidores Web de comercio electrónico, así como por aquellos que contienen información personal o confidencial.

HTTP utiliza el puerto 80 (equivalente de alguna forma al identificador de conexión o de servicio TCP) para todas las conexiones por defecto (podemos utilizar otros puertos diferentes del 80)-

EL HTTPS utiliza por defecto el puerto 443.



Figura II.4 Funcionamiento del protocolo HTTP

De manera esquemática, el funcionamiento de HTTP es el siguiente:

El cliente establece una conexión TCP hacia el servidor, hacia el puerto HTTP (o el indicado en la dirección de conexión), envía un comando HTTP de petición de un recurso (junto con algunas cabeceras informativas) y por la misma conexión el servidor responde con los datos solicitados y con algunas cabeceras informativas.¹

El lenguaje HTML

El otro puntal del éxito del WWW ha sido el lenguaje HTML (hypertext mark-up language). Se trata de un lenguaje de marcas (se utiliza insertando marcas en el interior del texto) que nos permite representar de forma rica el contenido y también referenciar otros recursos (imágenes, etc.), enlaces a otros documentos (la característica más destacada del WWW), mostrar formularios para posteriormente procesarlos, etc.

¹<http://www.scribd.com/doc/4605108/Desarrollo-de-Aplicaciones-Web>.

El lenguaje HTML empieza a proporcionar funcionalidades más avanzadas para crear páginas más ricas en contenido. Además se ha definido una especificación compatible con HTML, el XHTML (*extensible hypertext markup language*) que se suele definir como una versión XML validable de HTML, proporcionándonos un XML Schema contra el que validar el documento para comprobar si está bien formado, etc.

2.1.4 Tecnología java

2.1.4.1 La plataforma Java

Java es el nombre de un entorno o plataforma de computación originaria de *Sun Microsystems*, capaz de ejecutar aplicaciones desarrolladas usando el Lenguaje de programación Java u otros lenguajes que compilen a bytecode y un conjunto de herramientas de desarrollo. En este caso, la plataforma no es un hardware específico o un sistema operativo, sino más bien una máquina virtual encargada de la ejecución de aplicaciones, y un conjunto de librerías estándar que ofrecen funcionalidad común.

La plataforma así llamada incluye:

- ✓ Edición Estándar (*Java Platform, Standard Edition*), o Java SE (antes J2SE)
- ✓ Plataforma Java, Edición Empresa (*Java Platform, Enterprise Edition*), o Java EE (antes J2EE)
- ✓ Plataforma Java, Edición Micro (*Java Platform, Micro Edition*), o Java ME (antes J2ME)

Desde 2006, la versión actual de la Plataforma *Java Standard Edition* se le conoce como Java SE 6 como versión externa, y 1.6 como versión interna. Sin embargo, se prefiere el término versión 6.

La Plataforma Java se compone de un amplio abanico de tecnologías, cada una de las cuales ofrece una parte del complejo de desarrollo o del entorno de ejecución en tiempo real. Por ejemplo, los usuarios finales suelen interactuar con la máquina virtual de Java y el conjunto estándar de bibliotecas. Además, las aplicaciones Java pueden usarse de forma variada, como

²por ejemplo ser incrustadas en una página Web. Para el desarrollo de aplicaciones, se utiliza un conjunto de herramientas conocidas como JDK (*Java Development Kit*, o herramientas de desarrollo para Java).

2.1.4.2 Librerías de Java

En la mayoría de los sistemas operativos actuales, se ofrece una cantidad de código para simplificar la tarea de programación. Este código toma la forma, normalmente, de un conjunto de librerías dinámicas que las aplicaciones pueden llamar cuando lo necesiten. Pero la Plataforma Java está pensada para ser independiente del sistema operativo subyacente, por lo que las aplicaciones no pueden apoyarse en funciones dependientes de cada sistema en concreto. Lo que hace la Plataforma Java, es ofrecer un conjunto de librerías estándar, que contiene mucha de las funciones reutilizables disponibles en los sistemas operativos actuales.

Las librerías de Java tienen tres propósitos dentro de la Plataforma Java. Al igual que otras librerías estándar, ofrecen al programador un conjunto bien definido de funciones para realizar tareas comunes, como manejar listas de elementos u operar de forma sofisticada sobre cadenas de caracteres. Además, las librerías proporcionan una interfaz abstracta para tareas que son altamente dependientes del hardware de la plataforma destino y de su sistema operativo. Tareas tales como manejo de las funciones de red o acceso a ficheros, suelen depender fuertemente de la funcionalidad nativa de la plataforma destino. En el caso concreto anterior, las librerías `java.net` y `java.io` implementan el código nativo internamente, y ofrecen una interfaz estándar para que aplicaciones Java puedan ejecutar tales funciones. Finalmente, no todas las plataformas soportan todas las funciones que una aplicación Java espera. En estos casos, las librerías bien pueden emular esas funciones usando lo que esté disponible, o bien ofrecer un mecanismo para comprobar si una funcionalidad concreta está presente.

2.1.4.3 Lenguaje de programación Java

¹⁸ <http://www.scribd.com/doc/4605108/Desarrollo-de-Aplicaciones-Web>.

La palabra Java, por sí misma, se refiere habitualmente al lenguaje de programación Java, que fue diseñado para usar con la Plataforma Java. Los lenguajes de programación se encuentran fuera del ámbito de lo que es una "plataforma", aunque el lenguaje de programación Java es uno de los componentes fundamentales de la propia plataforma. El propio lenguaje y el entorno en tiempo de ejecución suelen considerarse una única entidad.

2.1.4.4 Introducción a los Java Servlets

Los Servlets de Java son la propuesta de la tecnología Java para el desarrollo de aplicaciones Web. Un Servlet es un programa que se ejecuta en un servidor Web y que construye una página Web que es devuelta al usuario. Esta página, construida dinámicamente, puede contener información procedente de bases de datos, ser una respuesta a los datos introducidos por el usuario, etc.

Los Servlets Java presentan una serie de ventajas sobre los CGI, el método tradicional de desarrollo de aplicaciones Web. Éstos son más portables, más potentes, mucho más eficientes, más fáciles de usar, más escalables, etc.

Eficiencia

Con el modelo tradicional de CGI, cada petición que llega al servidor dispara la ejecución de un nuevo proceso. Si el tiempo de vida del CGI (el tiempo que tarda en ejecutarse) es corto, el tiempo de instanciación (el tiempo de arrancar un proceso) puede superar al de ejecución.

Con el modelo de Servlets, la máquina virtual de Java, el entorno donde se ejecutan, se arranca al iniciar el servidor, permaneciendo arrancada durante toda la ejecución del mismo. Para atender cada petición no se arranca un nuevo proceso, sino un *thread*, un proceso ligero de Java, mucho más rápido (de hecho, casi instantáneo).

Además, si tenemos x peticiones simultáneas de un CGI, tendremos x procesos simultáneos en memoria, consumiendo así x veces el espacio de un CGI (que, en caso de ser interpretado,

como suele ocurrir, implica el consumo de x veces el intérprete). En el caso de los Servlets, hay determinada cantidad de *threads*, pero sólo una copia de la máquina virtual y sus clases.

El estándar de Servlets también nos ofrece más alternativas que los CGI para optimizaciones: caches de cálculos previos, pools de conexiones de bases de datos, etc.

Facilidad de uso

El estándar de Servlets nos ofrece una magnífica infraestructura de desarrollo de aplicaciones Web, proporcionándonos métodos para análisis automático y decodificación de los datos de los formularios de HTML, acceso a las cabeceras de las peticiones HTTP, manejo de cookies, seguimiento, control y gestión de sesiones, entre otras muchas facilidades.

Potencia

Los Servlets Java permiten hacer muchas cosas que son difíciles o imposibles de realizar con los CGI tradicionales. Los Servlets pueden compartir los datos entre sí, permitiendo compartir datos, conexiones a bases de datos, etc. Asimismo, pueden mantener información de solicitud en solicitud, facilitando tareas como el seguimiento de las sesiones de usuario, etc.

Portabilidad

Los Servlets están escritos en Java y se rigen por un API estándar bien documentado. Como consecuencia de ello, los Servlets pueden ejecutarse en todas las plataformas que nos ofrezcan soporte de Java Servlets, sin tener que recompilar, modificarse, etc., sean estas plataformas Apache, iPlanet, IIS, etc., y además, con independencia de sistema operativo, arquitectura *hardware*, etc.

2.1.4.5 Java Server Pages o JSP

Las Java Server Pages (JSP) son una tecnología que nos permite mezclar HTML estático con HTML generado dinámicamente mediante código Java embebido en las páginas. Cuando programamos aplicaciones Web con CGI, gran parte de la página que generan los CGI es estática y no varía de ejecución en ejecución. Por su parte, la parte variable de la página es realmente dinámica y muy pequeña.

Tanto los CGI como los Servlet nos obligan a generar la página por completo desde nuestro código de programa, dificultando así las tareas de mantenimiento, diseño gráfico, comprensión del código, etc.

2.2 Evolución de los Frameworks para componentes Web

2.2.1 Introducción a Los Frameworks de java

Hoy en día, con la aparición de nuevas metodologías y herramientas en el ámbito de la ingeniería de software, el desarrollo de aplicaciones Web ha tomado un cauce un tanto distinto a lo que se tenía hace unos años atrás; esto representa que, si antes con el desarrollo y herramientas tradicionales demorábamos semanas y/o meses en prototiparlas, meses o hasta años en desarrollarlas completamente. Hoy en día con la aparición de los frameworks éste tiempo se ha reducido considerablemente, ya que ahora con la ayuda de estas nuevas herramientas como son los frameworks y teniendo como aliadas a las metodologías ágiles de desarrollo, podemos prototipar nuestras aplicaciones en cuestión de minutos u horas, y terminar su desarrollo en una par de semanas.

El objetivo de este trabajo es explicar de forma clara y sencilla en qué consiste un framework para sistemas Web y las características generales de estos frameworks.

2.2.1.1 Que es un Framework

El concepto Framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito que pueda ocurrírsenos.

En general, con el término Framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un Framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

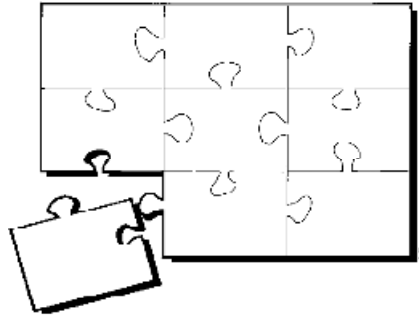


Figura II.5 **Aplicación configurable**

Uno de los frameworks más utilizados en desarrollo Web bajo plataforma Java son los Struts.

Los objetivos principales que persigue un Framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Un Framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

2.2.1.2 Patrón MVC

Para comprender como trabajan los frameworks Web existentes es imprescindible conocer el patrón MVC.

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, **el primero** es un modelo que representa los datos de la aplicación y sus reglas de negocio, **el segundo** es un conjunto de vistas que representa los formularios de entrada y salida de información, **el tercero** es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema.

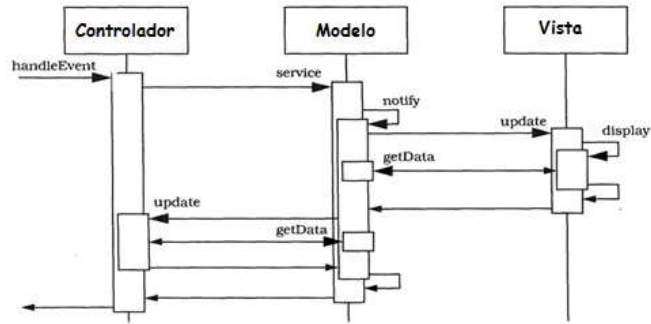


Figura II.6 Patrón Modelo-Vista-Controlador

El patrón MCV ha cobrado una notable importancia, sobre todo por su cualidad de dividir en capas el desarrollo de un proyecto o aplicación Web.

- ✓ **Modelo:** Encargado de modelar y contener la lógica del dominio de la aplicación.
- ✓ **Vista:** Encargada de contener la lógica de presentación.
- ✓ **Controlador:** Encargado de ser el engranaje entre el modelo y la vista; y también contener la lógica del negocio.

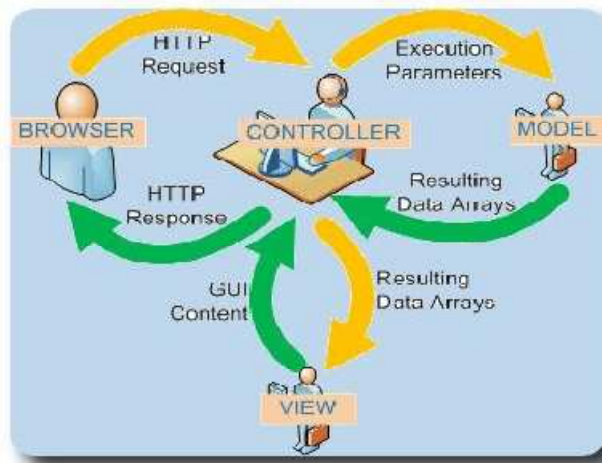


Figura II.7 Patrón MVC

2.2.1.3 Tipos de Framework Web.

Existen varios tipos de frameworks Web: orientados a la interfaz de usuario, como Java Server Faces, orientados a aplicaciones de publicación de documentos, como Cocoon, orientados a la parte de control de eventos, como Struts y algunos que incluyen varios elementos como

Tapestry. La mayoría de frameworks Web se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC y JSP, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación.

Características:

A continuación enunciamos una serie de características que podemos encontrar en prácticamente todos los frameworks existentes.

Tabla II.II Características de un Framework

<i>Abstracción de URLs y sesiones.</i>	No es necesario manipular directamente las URLs ni las sesiones, el Framework ya se encarga de hacerlo.
<i>Acceso a datos.</i>	Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en BBDD, XML, etc..
<i>Controladores.</i>	La mayoría de frameworks implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.
<i>Autenticación y control de acceso.</i>	Incluyen mecanismos para la identificación de usuarios mediante login y password y permiten restringir el acceso a determinadas páginas a determinados usuarios

2.2.2 Componentes en java

Además de la amplia terminología individual que existe en Java, existen diversos componentes ("KITS") que agrupan estas funcionalidades, y son estos componentes con los que se trabaja "día a día" en proyectos Java.

2.2.2.1 JRE ("Java Runtime Environment")

Java Runtime Environment es un programa destinado a la Plataforma Java necesita dos componentes en el sistema donde se va a ejecutar: una máquina virtual de Java (*Java Virtual Machine, JVM*), y un conjunto de librerías para proporcionar los servicios que pueda necesitar la aplicación. La JVM que proporciona Sun Microsystems, junto con su implementación de las librerías estándar, se conocen como *Java Runtime Environment (JRE)* o Entorno en tiempo de ejecución para Java. El JRE es lo mínimo que debe contener un sistema para poder ejecutar una aplicación Java sobre el mismo.

En el concepto de máquina virtual se encierra el concepto común de un procesador "virtual" que ejecuta programas escritos en el lenguaje de programación Java. En concreto, ejecuta el código resultante de la compilación del código fuente, conocido como bytecode. Este "procesador" es la máquina virtual de Java o JVM, que se encarga de traducir (interpretar o compilar al vuelo) el bytecode en instrucciones nativas de la plataforma destino. Esto permite que una misma aplicación Java pueda ser ejecutada en una gran variedad de sistemas con arquitecturas distintas, siempre que con una implementación adecuada de la JVM. Este hecho es lo que ha dado lugar a la famosa frase: "*write once, run anywhere*" (escriba una vez, ejecute en cualquier parte). La condición es que no se utilicen llamadas nativas o funciones específicas de una plataforma y aún así no se asegura completamente que se cumpla una verdadera independencia de plataforma.

Desde la versión 1.2 de JRE, la implementación de la máquina virtual de Sun incluye un compilador JIT (Just In Time). De esta forma, en vez de la tradicional interpretación del código bytecode, que da lugar a una ejecución lenta de las aplicaciones, el JIT convierte el bytecode a código nativo de la plataforma destino. Esta segunda compilación del código penaliza en cuanto a tiempo, pero el código nativo resultante se ejecuta de forma más eficaz y rápida que si fuera interpretado. Otras técnicas de compilación dinámica del código durante el tiempo de ejecución permiten optimizar más aún el código, dejando atrás el estigma que caía sobre Java

en cuanto a su lentitud y en sus últimas versiones la JVM se ha optimizado a tal punto que ya no se considera una plataforma lenta en cuanto a ejecución de aplicaciones.

Java no fue la primera plataforma basada en el concepto de una máquina virtual, aunque es la que de más amplia difusión ha gozado. El empleo de máquinas virtuales se había centrado principalmente en el uso de emuladores para ayudar al desarrollo de hardware en construcción o sistemas operativos, pero la JVM fue diseñada para ser implementada completamente en software, y al mismo tiempo hacer que fuera portable a todo tipo de hardware.

Como su nombre lo indica este ambiente ("KIT") es utilizado solo para *ejecutar* ("Runtime") programas en Java. Esta situación se da cuando empresas de Software diseñan alguna interface gráfica o aplicación en Java para su producto. Cabe mencionar que muchos productos que utilizan Java para su interface gráfica o instalación *ya incluyen un JRE* para evitarle la molestia de instalarlo, uno de estos es Oracle; sin embargo, muchos productos requieren que usted posea este ambiente, si se pregunta: Porque me hacen que obtenga forzosamente este JRE ? Una de las principales razones es *costo* ya que las empresas deben pagar a *Sun* por *distribuir* este ambiente en su producto.

2.2.2.2 JDK,SDK,J2SE

"Java Development Kit"(JDK),"Standard Development Kit" (SDK) y "Java 2 Standard Edition" (J2SE) son nombres para el mismo componente e incluyen: El API de Java, el JRE (JVM), compilador de Java y otras funcionalidades definidas por Sun.

Java 2, Standard Edition (J2SE) es la edición principal de la plataforma Java sobre la cual se basan las demás ediciones. Provee las capacidades de desarrollo y ejecución de software escrito en lenguaje Java. Esta constituido de dos módulos principales (ver figura 2.):

- ✓ *Software Development Kit (J2SE SDK)*, conocido inicialmente como *Java Development Kit (JDK)*, proporciona el software necesario para desarrollar programas en Java como es el compilador, el debugger y las bibliotecas con las funcionalidades del lenguaje.
- ✓ *Java Runtime Environment (JRE)*, contiene sólo el ambiente necesario y las bibliotecas principales para ejecutar software escrito en Java.

J2SE incluye herramientas y APIs para desarrollar aplicaciones con interfaz gráfica, acceso a base de datos, acceso a directorios, seguridad, entrada/salida, programación en red y varias otras funcionalidades.

Las bibliotecas principales son clases que se encuentran dentro de los paquetes *java.** y las bibliotecas con extensiones estándares se encuentran como clases dentro de los paquetes *javax.**. Para poder nombrar bibliotecas de aplicaciones desarrollados por terceros, se utiliza la convención de invertir el nombre de dominio de Internet del desarrollador y luego separar por funcionalidades, así existe por ejemplo *org.w3c.** y *org.apache.** entre otros.

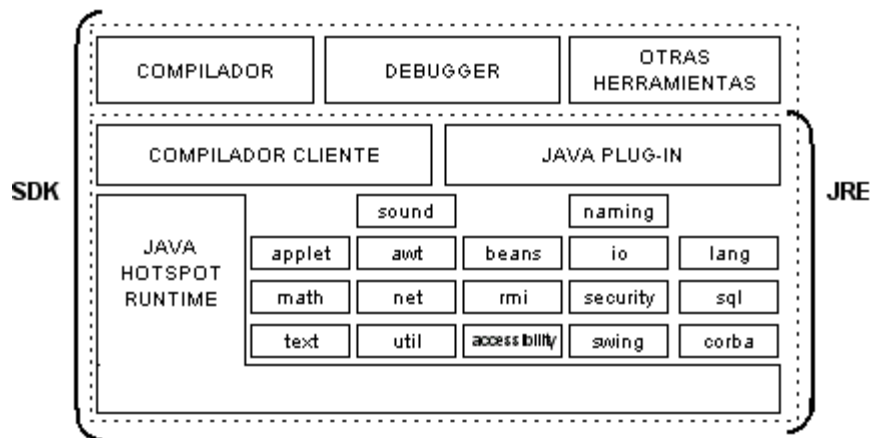


Figura II.8 Java 2, Standard Edition ³

2.2.2.3 J2EE (Java 2 Enterprise Edition)

J2EE es un grupo de especificaciones diseñadas por Sun que permiten la creación de aplicaciones empresariales, esto sería: acceso a base de datos (JDBC), utilización de directorios distribuidos (JNDI), acceso a métodos remotos (RMI/CORBA), funciones de correo electrónico (JavaMail), aplicaciones Web(JSP y Servlets)...etc. Aquí es **importante notar que J2EE es solo una especificación**, esto permite que diversos productos sean diseñados alrededor de estas especificaciones algunos son Tomcat y Weblogic.

³<http://www.dcc.uchile.cl/~jbarrios/J2EE/node12.html>

2.2.3 Plataforma y arquitectura de J2EE

La plataforma Java 2 Enterprise Edition (J2EE) es perfecta para desarrollar sólidas aplicaciones empresariales que permiten un desarrollo acelerado.

2.2.3.1 Arquitectura J2EE

La especificación de J2EE define su arquitectura basándose en los conceptos de *capas*, *containers*, *componentes*, *servicios* y las características de cada uno de éstos. Las aplicaciones J2EE son divididas en cuatro capas: la capa cliente, la capa Web, la capa negocio y la capa datos. La figura representa estas capas y las componentes relacionadas.

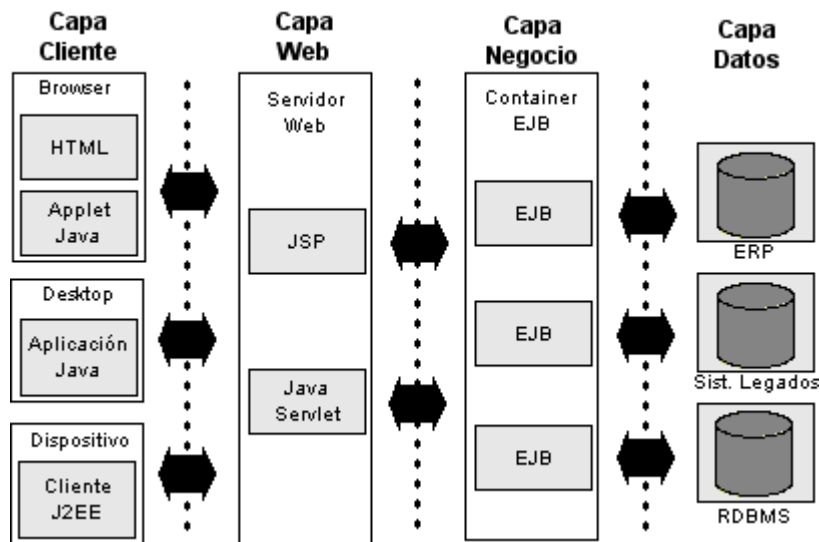


Figura II.9 Arquitectura J2EE⁴

Capa Cliente

Esta capa corresponde a lo que se encuentra en el computador del cliente. Es la interfaz gráfica del sistema y se encarga de interactuar con el usuario. J2EE tiene soporte para diferentes tipos de clientes incluyendo clientes HTML, applets Java y aplicaciones Java.

⁴<http://www.dcc.uchile.cl/~jbarrios/J2EE/node14.html>

Capa Web

Se encuentra en el servidor Web y contiene la lógica de presentación que se utiliza para generar una respuesta al cliente. Recibe los datos del usuario desde la capa cliente y basado en éstos genera una respuesta apropiada a la solicitud. J2EE utiliza en esta capa las componentes *Java Servlets* y *JavaServer Pages* para crear los datos que se enviarán al cliente.

Capa Negocio

Se encuentra en el servidor de aplicaciones y contiene el núcleo de la lógica del negocio de la aplicación. Provee las interfaces necesarias para utilizar el servicio de componentes del negocio. Las componentes del negocio interactúan con la capa de datos y son típicamente implementadas como componentes EJB.

Capa Datos

Esta capa es responsable del sistema de información de la empresa o *Enterprise Information System (EIS)* que incluye bases de datos, sistema de procesamiento datos, sistemas *legados* y sistemas de planificación de recursos. Esta capa es el punto donde las aplicaciones J2EE se integran con otros sistemas no J2EE o con sistemas legados.

2.2.3.2 Componentes de J2EE

Cada componente de J2EE es una unidad de software independiente y funcional que cumple con las condiciones de interfaz definidas por la especificación de la componente y sólo tiene dependencias explícitas con su entorno de ejecución o *container*. Una componente puede estar compuesta por una única clase, o lo más común, por un conjunto de clases, interfaces y recursos.

Las componentes principales en la plataforma J2EE son cuatro:

1. Aplicaciones cliente, son programas nativos escritos en Java que en general poseen su propia interfaz gráfica y que se ejecutan en un proceso independiente en un computador personal. Son ejecutados dentro del container de aplicación dado por el JRE y tienen acceso a todas las capacidades de la capa media J2EE.

2. Applets, son componentes que se ejecutan típicamente en un browser Web y proporcionan una interfaz Web mejorada para aplicaciones J2EE. En general se ejecutan en un container de applets de un browser, pero pueden ejecutarse en una variedad de otras aplicaciones o dispositivos que proporcionen soporte para el container. Son utilizados como alternativa a interfaces más limitadas basadas en HTML.
3. Java Servlets y JavaServer Pages, son llamados colectivamente con el nombre de *componentes Web*. Se ejecutan en un servidor Web para responder a solicitudes HTTP desde clientes y pueden generar páginas HTML, que en general corresponde a la interfaz de usuario de una aplicación, o puede generar XML u otro formato de datos que será utilizado por otras componentes de la aplicación.
4. Enterprise JavaBeans, son componentes que contienen la lógica del negocio para una aplicación J2EE. Se ejecutan en un ambiente distribuido y que soporta transacciones. Encapsulan el acceso al EIS a través de la utilización de objetos que proveen la funcionalidad de manejo de transacciones y persistencia.

2.2.3.3 Containers J2EE

Un *container* es un servicio que proporciona la infraestructura necesaria a una componente para ser ejecutada, para proveer sus servicios a un cliente y para dar comunicación con otras componentes. Las componentes de una aplicación J2EE no interactúan directamente entre ellas, si no que deben utilizar los protocolos y métodos dados por el *container* para ese fin. Un *container* usualmente provee sus servicios a las componentes como un *Java Runtime Environment (JRE)*.

Al existir un *container* entre las componentes y los servicios de J2EE se tiene la posibilidad de agregar transparentemente servicios como manejo de transacciones, chequeos de seguridad, administración de recursos y manejo de estados.

Un producto J2EE típico proveerá un *container* para cada tipo de componente de la aplicación: *container* de la aplicación cliente, *container* de applets, *container* de componentes Web y *container* de EJB.

2.2.3.4 Servicios J2EE

J2EE especifica los siguientes servicios estándares, junto con las APIs necesarias para la utilización por parte de cada componente. Algunos de estos servicios actualmente son provistos por J2SE.

1. *HTTP y HTTPS*: Protocolos estándares utilizados para comunicaciones Web y para comunicaciones seguras sobre *Secure Socket Layer (SSL)*, respectivamente. La API para clientes está definida por el paquete *java.net.** y la API para servidor está definida por las clases de servlets y JSP.
2. *JDBC*: Una API estándar para acceder a los recursos de una base de datos relacional de una forma independiente del proveedor. Esta API consta de dos partes, una interfaz para ser utilizada por las componentes y una interfaz de proveedores para definir drivers específicos. Oficialmente JDBC no es un acrónimo, aunque comúnmente se utiliza el nombre de *Java Database Connectivity*.
3. *JavaMail*: Una API que permite crear aplicaciones Java para mensajería y envío de correo electrónico en forma independiente de la plataforma y del protocolo a utilizar.
4. *JavaBeans Activation Framework (JAF)*: API que proporciona un Framework de activación que es utilizado por otros paquetes, como JavaMail. Los desarrolladores pueden utilizar JAF para determinar el tipo de un trozo arbitrario de datos, accederlo, descubrir las operaciones disponibles en él e instanciar el bean apropiado para ejecutar esas operaciones. Por ejemplo, JavaMail usa JAF para determinar qué tipo de objeto instanciar dependiendo del *Mime-Type* del objeto.
5. *Remote Method Invocation-Internet Inter-ORB Protocol (RMI-IIOP)*: Está compuesto de APIs que permiten la programación distribuida a través de Java RMI. Los protocolos utilizados para la comunicación pueden ser JRMP (protocolo nativo de RMI), o IIOP (protocolo de CORBA).
6. *Java Interface Definition Language (JavaIDL)*: Permite a las aplicaciones actuar como clientes de servicios CORBA, invocando objetos externos utilizando el protocolo IIOP.

7. *Java Transaction API (JTA)*: Permite el manejo de transacciones. Las aplicaciones pueden utilizar JTA para iniciar, cerrar o abortar transacciones. Además permite al container comunicarse con monitores transaccionales y administradores de recursos.
8. *Java Message Service (JMS)*: Es una API que se utiliza para comunicarse con un *Message-Oriented Middleware (MOM)* en una forma independiente al proveedor para permitir mensajería del tipo punto a punto y del tipo publicar/subscribir entre sistemas.
9. *Java Naming and Directory Interface (JNDI)*: Una API estándar para el registro y acceso de servicios y objetos. Incluye soporte para *LDAP (Lightweight Directory Access Protocol)*, *COS (CORBA Object Services)*, Naming Service y Java RMI Registry.
10. *Java API for XML Parsing (JAXP)*: Permite a las aplicaciones la utilización de documentos XML a través de las APIs estándares SAX, DOM y XSLT.
11. *J2EE Connector Architecture (JCA)*: Una API de J2EE que permite agregar recursos nuevos a cualquier producto J2EE. La arquitectura Connector define un contrato entre un servidor J2EE y un adaptador de recursos para permitir esto.
12. *Java Authentication and Authorization Service (JAAS)*: Proporciona una forma para la identificación de usuarios y su autorización para acceder a recursos de la aplicación. Implementa una versión en Java del estándar *Pluggable Authentication Module (PAM)*.

2.2.3.5 Elementos de un sistema multi-nivel J2EE

Desde la programación procedimental hasta el Modelo de desarrollo MVC, las diferentes metodologías de desarrollo han evolucionado, conservando un fin común: la total integración de componentes reutilizables y la separación de los niveles de programación.

Existen bastantes métodos de desarrollo multinivel en la actualidad. Muchos fueron pensados originalmente para la programación de aplicaciones tradicionales orientadas a un tipo de Sistema Operativo en particular.

Posteriormente, fueron adaptadas para su implementación en entornos de desarrollo orientados al Web. Sin duda, algunos de ellos son eficaces en cualquier plataforma. Pero el

hecho de que hayan sido concebidas primeramente para entornos tradicionales, no permite explotar el potencial que se tiene, al programar para la Web.

Sistema multinivel

La denominación de multinivel hace referencia al hecho de que podemos trabajar en el nivel que queramos, desde el más bajo como sería directamente el código fuente, hasta el más alto y abstracto, como sería el nivel de modelo. Con lo que enfocamos el primer punto fuerte de esta herramienta, que es básicamente el hecho de que el desarrollador no necesita tener amplia experiencia en Java. Por supuesto, se debe estar familiarizado con la programación orientada a objetos; debemos ser capaces de realizar un diseño basado en clases, pero no es necesario que nos peguemos con las particularidades de ninguna sintaxis estricta. A golpe de ratón y ventanas formulario, iremos creando toda la estructura de nuestro sitio Web.

2.2.3.6 Organización de un sistema multi-nivel bajo el meta patrón MVC

Aplicaciones Multinivel

Al hablar del desarrollo de aplicaciones Web resulta adecuado presentarlas dentro de las aplicaciones multinivel. Los sistemas típicos cliente/servidor pertenecen a la categoría de las aplicaciones de dos niveles. La aplicación reside en el cliente mientras que la base de datos se encuentra en el servidor. En este tipo de aplicaciones el peso del cálculo recae en el cliente, mientras que el servidor hace la parte menos pesada, y eso que los clientes suelen ser máquinas menos potentes que los servidores. Además, está el problema de la actualización y el mantenimiento de las aplicaciones, ya que las modificaciones a la misma han de ser trasladada a todos los clientes.

Para solucionar estos problemas se ha desarrollado el concepto de arquitecturas de tres niveles:

1. Interfaz de presentación
2. Lógica de la aplicación
3. y los datos.

La capa intermedia es el código que el usuario invoca para recuperar los datos deseados. La capa de presentación recibe los datos y los formatea para mostrarlos adecuadamente. Esta división entre la capa de presentación y la de la lógica permite una gran flexibilidad a la hora de construir aplicaciones, ya que se pueden tener múltiples interfaces sin cambiar la lógica de la aplicación.

La tercera capa consiste en los datos que gestiona la aplicación. Estos datos pueden ser cualquier fuente de información como una base de datos o documentos XML.

Convertir un sistema de tres niveles a otro multinivel es fácil ya que consiste en extender la capa intermedia permitiendo que convivan múltiples aplicaciones en lugar de una sola (véase la Figura II.9).

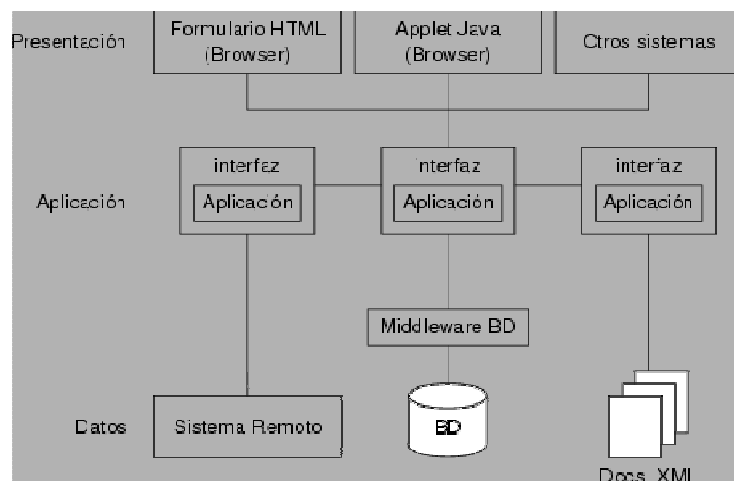


Figura II.10 Arquitectura Multinivel.

La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles (véase la Figura II.10). El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor Web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o script. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución. Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que

ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).

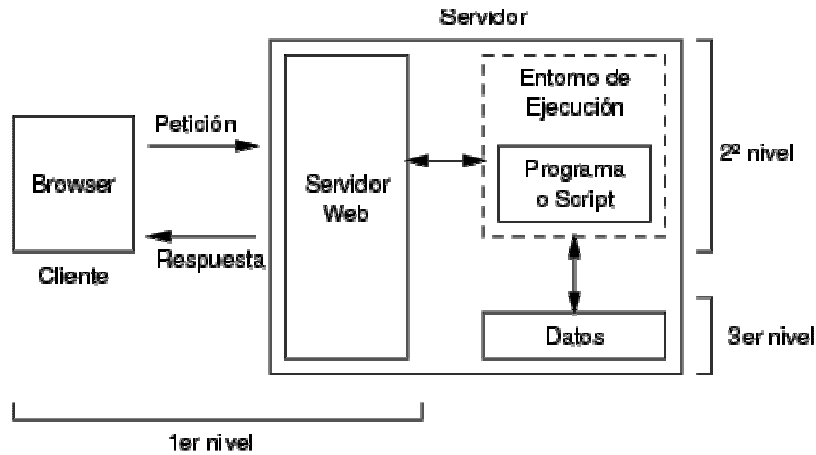


Figura II.11 Arquitectura Web de tres niveles.

2.3 Servidor de Aplicaciones

En informática, se denomina servidor de aplicaciones a un servidor en una red de computadores que ejecuta ciertas aplicaciones.

Usualmente se trata de un dispositivo de software que proporciona servicios de aplicación a las computadoras cliente. Un servidor de aplicaciones generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negocio y de acceso a los datos de la aplicación. Los principales beneficios de la aplicación de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones. Si bien el término es aplicable a todas las plataformas de software, hoy en día el término *servidor de aplicaciones* se ha convertido en sinónimo de la plataforma Java EE (antes J2EE) de Sun Microsystems.

2.3.1 Servidores de aplicación J2EE

Como consecuencia del éxito del lenguaje de programación Java, el término *servidor de aplicaciones* usualmente hace referencia a un servidor de aplicaciones Java EE. WebSphere (IBM) y WebLogic (Oracle, antes BEA Systems) están entre los servidores de aplicación Java

EE privativos más conocidos. EAServer (Sybase Inc.) es también conocido por ofrecer soporte a otros lenguajes diferentes a Java, como PowerBuilder. El servidor de aplicaciones JOnAS, desarrollado por el consorcio ObjectWeb, fue el primer servidor de aplicaciones libre en lograr certificación oficial de compatibilidad con J2EE. JBoss es otro servidor de aplicaciones libre y muy popular en la actualidad, así como el GlassFish de SUN. Mucha gente confunde Tomcat (The Apache Software Foundation) como un servidor de aplicaciones; sin embargo, es solamente un contenedor de servlets

Java EE provee estándares que permiten a un servidor de aplicaciones servir como "contenedor" de los componentes que conforman dichas aplicaciones. Estos componentes, escritos en lenguaje Java, usualmente se conocen como Servlets, Java Server Pages (JSPs) y Enterprise JavaBeans (EJBs) y permiten implementar diferentes capas de la aplicación, como la interfaz de usuario, la lógica de negocio, la gestión de sesiones de usuario o el acceso a bases de datos remotas.

La portabilidad de Java también ha permitido que los servidores de aplicación Java EE se encuentren disponibles sobre una gran variedad de plataformas, como Unix, Microsoft Windows y GNU/Linux.

2.3.2 Características comunes

Los servidores de aplicación típicamente incluyen también *middleware* (o software de conectividad) que les permite comunicarse con variados servicios, para efectos de confiabilidad, seguridad, etc. Los servidores de aplicación también brindan a los desarrolladores una Interfaz para Programación de Aplicaciones (API), de tal manera que no tengan que preocuparse por el sistema operativo o por la gran cantidad de interfaces requeridas en una aplicación Web moderna.

Los servidores de aplicación también brindan soporte a una gran variedad de estándares, tales como HTML, XML, IIOP, JDBC, SSL, etc., que les permiten su funcionamiento en ambientes Web (como Internet) y la conexión a una gran variedad de fuentes de datos, sistemas y dispositivos.

Usos

Un ejemplo común del uso de servidores de aplicación (y de sus componentes) son los portales de Internet, que permiten a las empresas la gestión y divulgación de su información, y un punto único de entrada a los usuarios internos y externos. Teniendo como base un servidor de aplicación, dichos portales permiten tener acceso a información y servicios (como servicios Web) de manera segura y transparente, desde cualquier dispositivo.

2.3.3 Tipos de Servidores de aplicaciones

Contamos con Servidores de aplicaciones como:

- ✓ BEA WebLogic
- ✓ GlassFish
- ✓ JBoss
- ✓ JOnAS
- ✓ Jetty
- ✓ NetWeaver
- ✓ Oracle WebLogic
- ✓ Sun Java System Application Server

Analizaremos algunos de los Servidores:

Bea WebLogic: Es un servidor de aplicaciones J2EE y también un servidor WebHTTP de BEA Systems de San José, California, para Unix, Linux, Microsoft Windows, y otras plataformas.

WebLogic puede utilizar Oracle, DB2, Microsoft SQL Server, y otras bases de datos que se ajusten al estándar JDBC. El servidor WebLogic es compatible con WS-Security y cumple con los estándares de J2EE 1.3 desde su versión 7 y con la J2EE 1.4 desde su versión 9 y JEE 5 para las versiones 9.2 y 10.x

GlassFish: Es un servidor de aplicaciones desarrollado por Sun Microsystems que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. La versión comercial es denominada Sun GlassFish Enterprise Server. Es

gratuito y de código libre, se distribuye bajo un licenciamiento dual a través de la licencia CDDL y la GNU GPL.

GlassFish está basado en el código fuente donado por Sun y Oracle Corporation, éste último proporcionó el módulo de persistencia. GlassFish tiene como base al servidor *Sun Java System Application Server* de Sun Microsystems, un derivado de Apache Tomcat, y que usa un componente adicional llamado Grizzly que usa Java NIO para escalabilidad y velocidad.

JBoss: Es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado J2EE 1.4, disponible en el mercado, ofreciendo una plataforma de alto rendimiento para aplicaciones de e-business. Combinando una arquitectura orientada a servicios revolucionaria con una licencia de código abierto, JBoss AS puede ser descargado, utilizado, incrustado y distribuido sin restricciones por la licencia. Por este motivo es la plataforma más popular de middleware para desarrolladores, vendedores independientes de software y, también, para grandes empresas.

Las características destacadas de JBoss incluyen:

- ✓ Producto de licencia de código abierto sin costo adicional.
- ✓ Cumple los estándares.
- ✓ Confiable a nivel de empresa
- ✓ Incrustable, orientado a arquitectura de servicios.
- ✓ Flexibilidad consistente
- ✓ Servicios del middleware para cualquier objeto de Java
- ✓ Ayuda profesional 24x7 de la fuente
- ✓ Soporte completo para JMX

JOnas: Es un servidor de aplicaciones J2EE de código abierto implementado en Java.

JOnAS forma parte de la iniciativa de código abierto de ObjectWeb, la cual fue lanzada en colaboración con varios socios, entre los que se encuentran Bull, France Télécom e INRIA.

Oracle WebLogic: Es un servidor de aplicaciones Java EE y también un servidor WebHTTP desarrollado por BEA Systems posteriormente adquirida por Oracle Corporation. Se ejecuta en Unix, Linux, Microsoft Windows, y otras plataformas.

WebLogic puede utilizar Oracle, DB2, Microsoft SQL Server, y otras bases de datos que se ajusten al estándar JDBC. El servidor WebLogic es compatible con WS-Security y cumple con los estándares de J2EE 1.3 desde su versión 7 y con la J2EE 1.4 desde su versión 9 y Java EE para las versiones 9.2 y 10.x

Oracle WebLogic Server es parte de Oracle WebLogic Platform. Los demás componentes de esta plataforma son:

- ✓ Portal, que incluye el servidor de comercio y el servidor de personalización (construido sobre un motor de reglas producido también por Bea, Rete),
- ✓ Weblogic Integration,
- ✓ Weblogic Workshop, una IDE para Java, y
- ✓ JRockit, una máquina virtual Java (JVM) para CPUs de Intel

WebLogic Server incluye interoperabilidad .NET y admite las siguientes capacidades de integración nativa:

- ✓ Mensajería nativa JMS a escala de empresa
- ✓ J2EE Connector Architecture
- ✓ Conector WebLogic/Tuxedo
- ✓ Conectividad COM+
- ✓ Conectividad CORBA
- ✓ Conectividad IBM WebSphere MQ

Oracle WebLogic Server Process Edition también incluye Business Process Management y funcionalidad de mapeo de datos.

WebLogic admite políticas de seguridad administradas por Security Administrators. El modelo de seguridad de WebLogic Server incluye:

- ✓ Separar la lógica de aplicaciones de negocio del código de seguridad
- ✓ El rango completo de cobertura de seguridad tanto para los componentes J2EE y no J2EE.

2.4 Estudio de las tecnologías JAVA en cuanto al desarrollo de aplicaciones Web.

2.4.1 Tecnología JSF.

JavaServer Faces (JSF) es una tecnología para aplicaciones Java basadas en Web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías.

JSF incluye:

- ✓ Un conjunto de APIs para: Representar componentes de Interfaz de Usuario (UI) y gestionar su estado.
 - Manejar eventos, validar en el servidor y conversión de datos.
 - Definir la navegación de páginas.
 - Soporte de internacionalización y accesibilidad
- ✓ Un conjunto por defecto de componentes para la interfaz de usuario.
- ✓ Dos librerías de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz Java Server Faces dentro de una página JSP.
- ✓ Un modelo de eventos en el lado del servidor.
- ✓ Administración de estados.
- ✓ Beans administrados.

2.4.1.1 Características

Este modelo de programación bien definido y la librería de etiquetas para componentes UI facilita de forma significativa la tarea de la construcción y mantenimiento de aplicaciones Web con UIs del lado del servidor. Con un mínimo esfuerzo, podemos:

- ✓ Conectar eventos generados en el cliente a código de la aplicación en el lado del servidor.
- ✓ Mapear componentes UI a una página de datos del lado del servidor.

- ✓ Construir un UI con componentes reutilizables y extensibles.
- ✓ Grabar y restaurar el estado del UI más allá de la vida de las peticiones de servidor.

Como se puede apreciar en la siguiente figura, el interface de usuario que creamos con la tecnología JavaServer Faces (representado por myUI en el gráfico) se ejecuta en el servidor y se renderiza en el cliente.

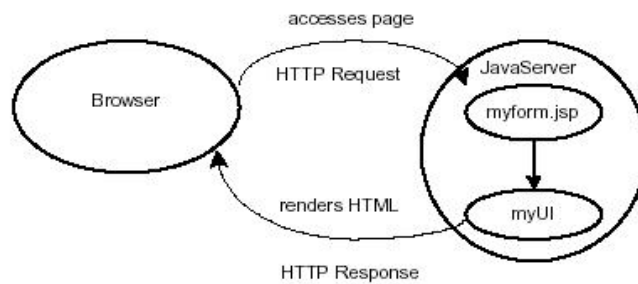


Figura II.12 Interface de Usuario con JSF

La página JSP, (myform.jsp), dibuja los componentes del interface de usuario con etiquetas personalizadas definidas por la tecnología JavaServer Faces. El UI de la aplicación Web (representado por myUI en la imagen) maneja los objetos referenciados por la página JSP:

- ✓ Los objetos componentes que mapean las etiquetas sobre la página JSP.
- ✓ Los oyentes de eventos, validadores, y los conversores que están registrados en los componentes.
- ✓ Los objetos del modelo que encapsulan los datos y las funcionalidades de los componentes específicos de la aplicación.

2.4.1.2 Ventajas De JSF

Existen numerosas ventajas que hacen que JSF sea una tecnología apropiada para el desarrollo de aplicaciones Web:

- ✓ Una de las grandes ventajas de la tecnología JavaServer Faces es que ofrece una clara separación entre el comportamiento y la presentación. Las aplicaciones Web construidas con tecnología JSP conseguían parcialmente esta separación. Sin

embargo, una aplicación JSP no puede mapear peticiones HTTP al manejo de eventos específicos de los componentes o manejar elementos UI como objetos con estado en el servidor.

- ✓ La tecnología JavaServer Faces permite construir aplicaciones Web que implementan una separación entre el comportamiento y la presentación tradicionalmente ofrecida por arquitectura UI del lado del cliente. JSF se hace fácil de usar al aislar al desarrollador del API de Servlet.
- ✓ La separación de la lógica de la presentación también le permite a cada miembro del equipo de desarrollo de una aplicación Web enfocarse en su parte del proceso de desarrollo, y proporciona un sencillo modelo de programación para enlazar todas las piezas.
- ✓ Aunque la tecnología JavaServer Faces incluye una librería de etiquetas JSP personalizadas para representar componentes en una página JSP, los APIs de la tecnología JavaServer Faces se han creado directamente sobre el API JavaServlet. Esto permite hacer algunas cosas: usar otra tecnología de presentación junto a JSP, crear componentes propios personalizados directamente desde las clases de componentes, y generar salida para diferentes dispositivos cliente. Así, se podrán encapsular otras tecnologías como Ajax en componentes JSF, haciendo su uso más fácil y productivo, al aislar al programador de ellas.
- ✓ JavaServer Faces ofrece una gran cantidad de componentes opensource para las funcionalidades que se necesiten. Los componentes Tomahawk de MyFaces y ADFFaces de Oracle son un ejemplo. Además, también existe una gran cantidad de herramientas para el desarrollo IDE en JSF al ser el estándar de JAVA.
- ✓ La tecnología JavaServer Faces proporciona una rica arquitectura para manejar el estado de los componentes, procesar los datos, validar la entrada del usuario, y manejar eventos.
- ✓ Además, ofrece una rápida adaptación para nuevos desarrolladores.

2.4.2 Tecnología ADF

Oracle Application Development Framework, generalmente llamado Oracle ADF, es un framework comercial de Java para la creación de aplicaciones empresariales. Es una herramienta del tipo RAD que se basa en patrones de diseño listos para usar. Provee un enfoque visual y declarativo para el desarrollo de aplicaciones J2EE

Oracle ADF está basado en la arquitectura MVC (Model-View-Controller o Modelo-Vista-Controlador).

2.4.2.1 Características

- ✓ La programación de esta Tecnología se basa en la técnica arrastrar y soltar.
- ✓ Contiene un conjunto de componentes inteligentes de software que cooperan entre sí.
- ✓ Frameworks de desarrollo basados en estándares integrados con Herramientas + Tecnología Web 2.0, AJAX.
- ✓ Reduce la necesidad de escribir código.

2.4.2.2 Ventajas

- ✓ En caso de aplicaciones destinadas a internet, sólo será necesario usar un subconjunto del Framework ADF para lograr los resultados esperados, pero pudiendo usar el mismo equipo de desarrollo para todo tipo de aplicaciones, y por tanto ganando en flexibilidad y rentabilizando al máximo la inversión.
- ✓ A grandes rasgos trata de separar el modelo, de la vista y del controlador del patrón de diseño MVC.

2.4.2.3 Modelo Vista Controlador (MVC)

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión

de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

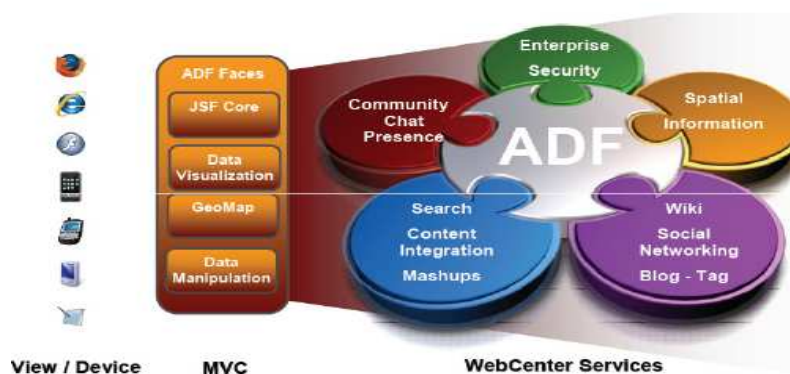


Figura II.13 Estructura ADF⁵

- ✓ **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos;

Entidades

Un objeto entidad (Entity) de ADF representa una entidad de negocio. Esto normalmente se traduce en una fila de datos de una tabla, y cada atributo de la entidad representa una columna de esa tabla.

En ADF, la entidad maneja otras tareas, como las validaciones de negocio y las máscaras de formato. En la mayoría de los casos, el modelo es el lugar más lógico para dirigir esos conceptos porque permitirá una máxima reutilización. Las validaciones, predicadas por requerimientos de negocio, se supone que son importantes sin importar como se utilice el modelo. Por ejemplo, la verificación de que un empleado siempre está asignado a un y sólo un departamento no cambiará si la aplicación se convierte desde una aplicación Web a una aplicación Swing. ADF propagará esas validaciones y formatos al controlador y a la vista, aliviando la necesidad de corregir esos problemas en otros lugares.

⁵http://www.bbr.cat/presentaciones/PDF/Noticias_EventosBbr/ADF_Enterprise_2_0/ADF.%20EI%20framework%20de%20desarrollo%20de%20Fusion%20v3.0.pdf

✓ **Vistas**

Un objeto vista de ADF expone un objeto entidad a los usuarios de la aplicación. Se puede acceder a una entidad desde varios sitios, pero el contexto en el que se utiliza variará. Se pueden establecer filtrados, ordenaciones y relaciones entre objetos mediante un objeto vista, asegurando que el modelo se representa en la forma en la que tiene más sentido.

Módulo de aplicación

El módulo de aplicación ADF empaqueta varios componentes vista mientras proporciona acceso a su capa de servicios de negocio, que podría utilizar tecnologías como EJB para manejar su base de datos. El módulo de aplicación proporciona vistas ADF con soporte de transacciones, y otros servicios importantes centrados en los datos.

Vista y Controlador

Nada en su modelo debería dictar una tecnología de vista particular. Un módulo de aplicación puede servir como base para una aplicación Swing o para una aplicación Web basada en Struts. Si desarrolla una aplicación ligera, los componentes Swing se podrían comunicar directamente con el módulo de aplicación. Cuando desarrolle una aplicación basada en la Web, su aplicación usará un controlador claramente definido para manejar la interacción entre el modelo y la vista.

✓ **Controlador**

En versiones anteriores, ADF proporcionaba su propio controlador. Ese controlador ha sido reemplazado en favor de Struts. Struts utiliza un fichero XML, llamado struts-config.xml, para manejar las peticiones de la aplicación y despacharlas a los manejadores adecuados. Los desarrolladores con experiencia en Struts apreciarán la posibilidad de utilizar esos conocimientos.

2.4.2.4 Ventajas del patrón

MVC nos aporta una construcción de software muy mantenible, en la que se pueden localizar de forma ágil los errores. Supone un diseño modular, y muy poco acoplado, favoreciendo la reutilización.

Por ejemplo, podemos realizar una interfaz gráfica de escritorio, y una Web, que compartirían las capas Controlador y Modelo, y solo trataríamos como desarrollos distintos las dos capas Vista.⁶

2.5 Selección de la herramienta IDE más adecuada para las tecnologías JSF y ADF

2.5.1 Entorno de desarrollo Integrado (IDE)

Un **entorno de desarrollo integrado** o, en inglés, Integrated Development Environment ('IDE'), es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, etc.

Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación.

2.5.1.1 Componentes de un IDE

- ✓ Un editor de texto.
- ✓ Un compilador.
- ✓ Un intérprete.

⁶<http://caraballomaestre.blogspot.com/2009/02/arquitectura-j2ee-patron-mvc.html>

- ✓ Herramientas de automatización.
- ✓ Un depurador.
- ✓ Posibilidad de ofrecer un sistema de control de versiones.
- ✓ Factibilidad para ayudar en la construcción de interfaces gráficas de usuarios.

2.5.1.2 Software's IDE

Algunos ejemplos de *entornos integrados de desarrollo* (IDE) son los siguientes:

- ✓ Emacs
- ✓ Eric (software) (IDE para Python)
- ✓ Qt Creator
- ✓ Anjuta (GTK, pensado para escritorio GNOME)
- ✓ C++Builder y TurboC++ Explorer de Borland (ahora, de CodeGear)
- ✓ Clarion
- ✓ Code::Blocks
- ✓ CodeWarrior
- ✓ Delphi y Turbo Delphi de Borland(ahora de CodeGear)
- ✓ Eclipse
- ✓ JBuilder de Borland (ahora de CodeGear)
- ✓ JDeveloper de Oracle_Corporation
- ✓ JCreator
- ✓ KDevelop (QT, pensado para escritorio KDE)
- ✓ Lazarus
- ✓ MS Visual Studio .NET y Visual Studio Express de Microsoft
- ✓ NetBeans
- ✓ Turbo Pascal de Borland
- ✓ Watcom C/C++ Compiler y OpenWatcom
- ✓ Xcode de Apple

2.5.2 Determinación del IDE

La elección de un IDE para trabajar con Java es una decisión muy importante. El hecho de que Sun ponga a disposición gratuita el JDK de Java, y otras descargas, ha permitido aflorar un conjunto de aplicaciones de apoyo a los desarrolladores Java, incluyendo los IDEs. La mayoría de los IDE's son propietarios de compañías como Sun, Borland o IBM. Esto es un inconveniente, tanto monetario si quieres operar con una licencia legal.

Se encontraron varios sitios donde se analiza las IDE's existentes:

En una encuesta realizada en el sitio Web sobre ¿Qué IDE utilizas a menudo para programar en java?, ubicado en⁷ nos muestra que NetBeans es la que más utilizan los desarrolladores, ya que el gran numero de plugins facilita el desarrollo.

La comunidad que participó en la misma menciona las IDE's como: NetBeans, Eclipse, IntelliJ, JCreator. En esta encuesta participaron 232 integrantes, la cual muestra los siguientes puntajes.



Figura II.14 NetBeans vs. otros IDE's

Tabulación:

⁷<http://www.javamexico.org/node/351/results>

Tabla II.III Tabulación de NetBeans vs. Otros IDE's

IDE's	Porcentaje
NetBeans	53 %
Eclipse	31 %
Intelij	1 %
JCreator	11 %
Otros	5 %

Según el documento de BenchMark sobre Tecnologías Faces ubicado en el sitio: ⁸nos indica que JSF puede trabajar con: NetBeans, Eclipse entre otras y para trabajar con ADF la herramienta de desarrollo que recomienda es JDevelopers, como se muestra en la siguiente tabla.

Tabla II.IV Herramientas de desarrollo vs. JSF

	Herramientas de desarrollo vs jsf		
	jsf	ADFaces	Myfaces
Jdeveloper 10.1.2, 10.1.3, 11G		x	
Eclipse	x		x
NetBeans	x		x

Según el Artículo ubicado en el sitio:⁹nos da a conocer la evolución de uso de las IDE's JDevelopers y NetBeans desde los años 2004 a 2009. Esta comparación se da debido a que en la actualidad JDeveloper y NetBeans son las más utilizadas por sus ventajas.

⁸http://www.postgradoinformatica.edu.bo/enlaces/investigacion/pdf/INGSW3_50.pdf?PHPSESSID=3ab62098bf2d8aae88430bb18791f9ac,

⁹<http://willyxoft.wordpress.com/2009/11/13/netbeans-vs-jdeveloper/>

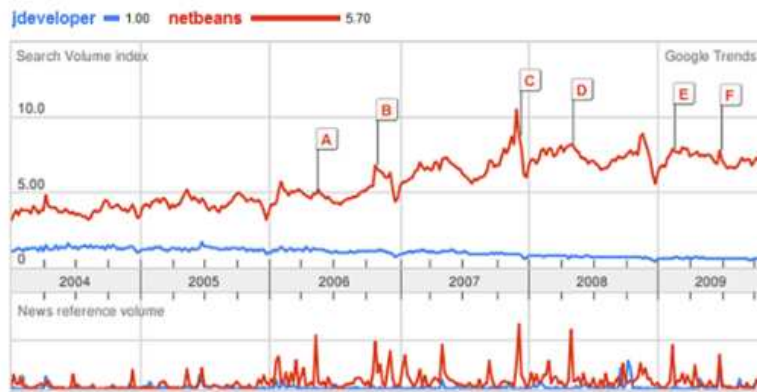


Figura II.15 Evolución de IDE's

Según el foro por JAVA Sun Microsystems ubicado en¹⁰, Nos informa que JDeveloper, es gratuito, pero no open source. Tiene una licencia Oracle Technology Network (OTN).

Además JDeveloper es un IDE de Oracle, sin embargo es una herramienta de desarrollo gratuita, intuitiva, versátil y fácil de utilizarla y conseguir resultados rápidos. También se puede desarrollar con uno u otros FrameWorks.

Por todo lo arriba mencionado para el análisis comparativo se ha seleccionado las IDE's NetBeans para JSF y JDeveloper para ADF por los siguientes motivos:

- ✓ Se eligió la IDE NetBeans para JSF, ya que es gratuitas y mucho más fácil de conseguir que las aplicaciones que son de pago. Y JDeveloper para ADF ya que esta IDE es gratuito y exclusivo para ADF.
- ✓ NetBeans tiene un conjunto de componentes de **Java Server Faces**, creado para facilitar notablemente su uso, cuenta con:
 - Un Editor Visual para JSF.
 - Componentes que son utilizados a través de uniones (bindings) a la base de datos.
 - Un conjunto de componentes útiles como Tabs, botones de Imágenes, calendarios, etc.

¹⁰<http://osum.sun.com/forum/topics/2181626:Topic:258627?page=2&commentId=2181626%3AComment%3A1012235&x=1#2181626Comment1012235>

- Un esquema de eventos fácil de usar.
- Un editor para las reglas de navegación de la aplicación, el cual se pueden ligar los eventos invocados por los componentes y el conjunto de páginas creadas en la aplicación.
- ✓ JDeveloper es una herramienta de desarrollo de ADF, las cuales presentan las siguientes ventajas de uso:
 - ✓ Soporta todo el ciclo de desarrollo en J2EE
 - Diferentes formas de despliegue
 - Aplicación Java, applet, JSP, servlet, EJB
 - ✓ Fácil desarrollo, depuración y despliegue
 - ✓ Framework de acceso a datos: Business Components (BC4J y ADF BC)

2.5.3 Análisis de las IDE's Seleccionadas

En este apartado describiremos y analizaremos los siguientes IDE's seleccionados anteriormente.

Tabla II.V IDE's

IDE	TECNOLOGIA
NetBeans	JSF
JDeveloper de Oracle_Corporation	ADF

2.5.3.1 El IDE NetBeans

Se refiere a una plataforma para el desarrollo de aplicaciones de escritorio usando Java y a un entorno de desarrollo integrado (IDE) desarrollado usando la Plataforma NetBeans.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos

pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

NetBeans es muy flexible y basado en la calidad, entorno de desarrollo integrado (IDE), por escrito, en el lenguaje de programación Java. NetBeans El proyecto consta de un IDE de código abierto y una plataforma de aplicaciones, que puede ser usado como un marco aplicable a generar algún tipo de aplicación. NetBeans IDE 6.1 El apoyo a la JSF y es muy fácil de crear las aplicaciones JSF con NetBeans.

Historia

NetBeans comenzó como un proyecto estudiantil en República Checa (originalmente llamado Xelfi), en 1996 bajo la tutoría de la Facultad de Matemáticas y Física en la Universidad de Charles en Praga. La meta era escribir un entorno de desarrollo integrado (IDE) para Java parecida a la de Delphi. Xelfi fue el primer entorno de desarrollo integrado escrito en Java, con su primer pre-release en 1997.

Xelfi fue un proyecto divertido para trabajar, ya que las IDEs escritas en Java eran un territorio desconocido en esa época. El proyecto atrajo suficiente interés, por lo que los estudiantes, después de graduarse, decidieron que lo podían convertir en un proyecto comercial. Prestando espacios Web de amigos y familiares, formaron una compañía alrededor de esto. Casi todos ellos siguen trabajando en NetBeans.

Tiempo después, ellos fueron contactados por Roman Stanek, un empresario que ya había estado relacionado con varias iniciativas en la República Checa. Él estaba buscando una buena idea en que invertir, y encontró en Xelfi una buena oportunidad. Ellos se reunieron, y el negocio surgió.

El plan original era desarrollar unos componentes JavaBeans para redes. Jarda Tulach, quien diseñó la arquitectura básica de la IDE, surgió con la idea de llamarlo NetBeans, con el fin de describir lo que ellos harían. Cuando las especificaciones de los Enterprise JavaBeans salieron, ellos decidieron trabajar con este estándar, ya que no tenía sentido competir con él, sin embargo el nombre de NetBeans se quedó.

En la primavera de 1999, Netbeans DeveloperX2 fue lanzado, soportando Swing. Las mejoras de rendimiento que llegaron con el JDK 1.3, lanzado en otoño de 1999, hicieron a NetBeans una alternativa realmente viable para el desarrollo de herramientas. En el verano de 1999, el equipo trabajó duro para rediseñar a DeveloperX2 en un NetBeans más modular, lo que lo convirtió en la base de NetBeans hoy en día.

Algo más paso en el verano de 1999. Sun Microsystems quería una mejor herramienta de desarrollo de Java, y comenzó a estar interesado en NetBeans. En otoño de 1999, con la nueva generación de NetBeans en Beta, el acuerdo fue realizado.

Sun adquirió otra compañía de herramientas al mismo tiempo, Forté, y decidió renombrar NetBeans a Forté for Java. El nombre de NetBeans desapareció de vista por un tiempo.

Seis meses después, se tomó la decisión de hacer a NetBeans open source. Mientras que Sun había contribuido considerablemente con líneas de código en varios proyectos de código abierto a través de los años, NetBeans se convirtió en el primer proyecto de código abierto patrocinado por ellos. En Junio del 2000 NetBeans.org fue lanzado.

NetBeans Hoy

Un proyecto de código abierto no es nada más ni nada menos que un proceso. Toma tiempo encontrar el equilibrio. El primer año, fue crucial como inicio. Los dos años siguientes, se orientó hacia código abierto. Como muestra de lo abierto que era, en los primeros dos años había más debate que implementación.

Con NetBeans 3.5 se mejoró enormemente en desempeño, y con la llegada de NetBeans 3.6, se reimplementó el sistema de ventanas y la hoja de propiedades, y se limpió enormemente la interfaz. NetBeans 4.0 fue un gran cambio en cuanto a la forma de funcionar del IDE, con nuevos sistemas de proyectos, con el cambio no solo de la experiencia de usuario, sino del reemplazo de muchas piezas de la infraestructura que había tenido NetBeans anteriormente. NetBeans IDE 5.0 introdujo un soporte mucho mejor para el desarrollo de nuevos módulos, el nuevo constructor intuitivo de interfaces Matisse, un nuevo y rediseñado soporte de CVS, soporte a Sun ApplicationServer 8.2, Weblogic9 y JBoss 4.

Con NetBeans 6.01 y 6.5 Se dio soporte a frameworks comerciales como son Struts, Hibernate.

La Plataforma NetBeans

Durante el desarrollo del NetBeans IDE ocurrió una cosa interesante. La gente empezó a construir aplicaciones usando el NetBeans core runtime con sus propios plug-ins, de hecho, esto se convirtió en un mercado bastante grande.

La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:

- ✓ Administración de las interfaces de usuario (ej. menús y barras de herramientas)
- ✓ Administración de las configuraciones del usuario
- ✓ Administración del almacenamiento (guardando y cargando cualquier tipo de dato)
- ✓ Administración de ventanas
- ✓ Framework basado en asistentes (diálogos paso a paso)

2.5.3.2 JDeveloper

JDeveloper: es un ambiente integrado para desarrollo (IDE), construido en Java para el desarrollo de proyectos Java. El ciclo de desarrollo es completamente atendido, es decir, en JDeveloper es posible modelar (UML), codificar, “debugar”, testear y “deployar”. El está preparado para el desarrollo en Web, XML, Web Services, SQL, J2EE y Oracle ADF. Provee funcionalidades nativas para las herramientas y *frameworks* abiertos mas populares, como: Struts, Ant, JUnit e CVS. Según Oracle, el JDeveloper es la herramienta seleccionada por los

clientes de Forms, Reports y Designer, esto porque el JDeveloper utiliza un modelo de desarrollo similar.

Una de las ventajas más grandes es que para el desarrollador el ambiente, los menús, etc, son los mismos; no se requiere de utilizar otras herramientas, el IDE se ajusta de acuerdo al tipo de tecnologías que vayas a utilizar.

La última pieza que diferencia a JDeveloper de los demás es ADF o Application Development Framework, un framework end-to-end para construir aplicaciones Java EE. Utilizando este framework en conjunto con JDeveloper se pueden crear aplicaciones basadas en MVC de manera productiva y basada en los estándares de la industria tales como EJB3, JPA, JSF

Vista

JDeveloper hace sencillo diseñar páginas visualmente utilizando JSP y JSTL. ADF también proporciona su propio marco de trabajo para vistas, llamado UIX, que ofrece un conjunto de elementos HTML para generar rápidamente páginas de aspecto consistente. Los componentes UIX se convertirán en la base de la implementación de JSF (JavaServer Faces) en futuras versiones.

Es el editor para aplicaciones desarrolladas en Java de Oracle, dentro de sus principales características cuenta con:

- ✓ El soporte de Oracle
- ✓ Es gratuito
- ✓ Fácil de usar, excelente para desarrolladores no experimentados en Java
- ✓ Permite desarrollar aplicaciones tanto para Desktop como para Web
- ✓ Excelente integración con Oracle 9i y 10g
- ✓ Permite generar diagramas de UML
 - Casos de uso – Especificación de Requerimientos
 - Diagramas de Actividad – Definición de procesos
 - Diagramas de Secuencia – Facilitan el modelado de flujos que siguen los componentes del sistema

- Diagrama de Clases – Representan la organización y las relaciones de las clases y componentes del sistema
- ✓ Permite generar diagramas de Entidad Relación para Bases de datos
- ✓ Permite la integración con servidores de control de versiones.

2.6 Selección de la Base de Datos más adecuada para las tecnologías JSF y ADF

2.6.1 Gestor de base de datos para JSF

Existen un sin número de bases de datos para interactuar con aplicaciones Web, sin embargo es importante resaltar que para un IDE de código libre sería conveniente usar un manejador de base de datos libre, es por eso que para las tecnologías Java encontramos disponibles las siguientes bases de datos más utilizadas:

- ✓ MySQL
- ✓ Postgresql

Encontramos una comparación de estas dos bases de datos la misma que nos parece importante analizar en la siguiente dirección, ¹¹, en un documento redactado por Daniel Pecos.

Lo mejor de PostGreSQL

Las características positivas que posee este gestor según las opiniones más comunes en Internet, son:

1. Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.
2. Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en las que MySQL no podría.
3. Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle.

¹¹http://danielpecos.com/docs/mysql_postgres/index.html

Desventajas

Por contra, los mayores inconvenientes que se pueden encontrar a este gestor son:

1. Consume gran cantidad de recursos.
2. Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
3. Es de 2 a 3 veces más lento que MySQL.

Lo mejor de MySQL

Es evidente que la gran mayoría de gente usa este gestor en Internet, por lo que encontrar opiniones favorables no ha resultado en absoluto complicado:

1. Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
2. Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
3. Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
4. Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.
5. El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro (Barrapunto.com) y de buscadores de aplicaciones (Freshmeat.net).

Desventajas

Debido a esta mayor aceptación en Internet, gran parte de los inconvenientes se exponen a continuación, han sido extraídos de comparativas con otras bases de datos:

1. Carece de soporte para transacciones, rollback's y subconsultas.

2. El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.
3. No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.

Cada uno de estos gestores es idóneo para ciertos campos, e intentar utilizar el otro acarrearía una pérdida de productividad del programa, como también grandes quebraderos de cabeza.

Ninguno de estos dos gestores son totalmente perfectos, por lo que no hay que obcecarse en la elección única y fanática, como se suele hacer en muchos casos de alguno de ellos. Simplemente se trata de escoger el más conveniente en cada caso. Éstos son los grandes inconvenientes y a la vez las grandes maravillas que conlleva el mundo OpenSource.

En base al análisis de los foros escogeremos Mysql para trabajar con JSF.

En base a bloggers encontrados en la Web, el que sobresale lo mencionamos a continuación. Sin embargo es importante saber que JSF se acopla a cualquier base de datos, esto se muestra en el foro de Java Hispana en la siguiente dirección¹², en la cual demuestran que depende de las características de la aplicación para poder interactuar con la base de datos.

Nosotros requerimos una combinación por lo cual usaremos para JSF el gestor de base de datos Mysql.

2.6.2 Gestor de base de datos para ADF

Para la tecnología ADF se puede usar también el gestor de base de datos Mysql, con tan solo incluir el driver en el IDE bastara, eso se demuestra en el siguiente apartado¹³

Así como en las características del modelo vista controlador es trivial la interacción entre ADF que usa su IDE y el gestor de base de datos Mysql. Se demuestra en la siguiente dirección¹⁴

¹²http://www.javahispano.org/forum/persistencia/es/base_de_datos_para_jsf_de_java/

¹³<http://www.zonaoracle.com/foro/index.php?topic=973.0>

¹⁴http://www.programacion.com/articulo/jap_adf_mysql/

2.6.3 Análisis del Gestor de base de datos Seleccionado.

En este apartado describiremos y analizaremos el Gestor de base de datos a usar correspondientes a las tecnologías que corresponden a la investigación y que fue seleccionado anteriormente.

Tabla II.VI Tecnologías y Base de Datos Seleccionadas

TECNOLOGIA	IDE	ODBC
JSF	NetBeans	MySQL
ADF	JDeveloper	MySQL

2.6.3.1 Base De Datos Mysql

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Historia de MySQL

MySQL surgió como un intento de conectar el gestor mSQL a las tablas propias de MySQL AB, usando sus propias rutinas a bajo nivel. Tras unas primeras pruebas, vieron que mSQL no era lo bastante flexible para lo que necesitaban, por lo que tuvieron que desarrollar nuevas

funciones. Esto resultó en una interfaz SQL a su base de datos, con una interfaz totalmente compatible a mSQL.

Por un lado dicen que sus librerías han llevado el prefijo 'my' durante los diez últimos años. Por otro lado, la hija de uno de los desarrolladores se llama My. No saben cuál de estas dos causas (aunque bien podrían tratarse de la misma), han dado lugar al nombre de este conocido gestor de bases de datos.

La versión estable de este gestor a días de hoy es la 3.23.49.

Características de MySQL

Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
4. Gran portabilidad entre sistemas.
5. Soporta hasta 32 índices por tabla.
6. Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

CAPITULO III

ANÁLISIS COMPARATIVO DE TECNOLOGÍAS DE APLICACIONES WEB EN EL ENTORNO JSF Y ADF.

3.1 Introducción

El auge de las tecnologías Java crece a medida que el tiempo transcurre, por lo cual es importante el desarrollo de aplicaciones Web utilizando este entorno.

Al momento de desarrollar un sistema informático uno de los puntos más importantes que deben tomar en cuenta los desarrolladores, es la selección de la tecnología de desarrollo puesto que esta determinara en gran parte el tiempo, facilidad, costo, y otros aspectos para la implementación del sistema.

El presente capitulo se refiere al análisis comparativo de las tecnologías de Aplicaciones Web en Java, que consiste en determinar cuál es la más apta para la implementación de aplicaciones Web, las personas dedicadas a la programación orientada a la Web deben tomar

una decisión importante en cuanto a la selección de tecnología que utilizará, esta elección se debe realizar a través de un análisis y criterios o parámetros de comparación, que permitan cumplir con las normas que requieren los sistemas Web y los requerimientos que demandan las empresas en la actualidad.

El objetivo de este capítulo es tener una base que permita la selección de la tecnología más idónea, para el desarrollo de aplicaciones Web.

3.2 Determinación De Las Tecnologías A Comparar

En el mercado podemos encontrar varias tecnologías para el desarrollo de Aplicaciones Web en Java, en la cual cada una de ellas brinda ciertas ventajas.

Diversos sitios Web que tienen que ver con el desarrollo de aplicaciones Web, que lo realizan utilizando Java, citan tecnologías como: JSF y ADF las mismas que permiten alcanzar una solución global y que actualmente son las más preferidas por parte de los desarrolladores.

Según el sitio Web de la Comunidad de desarrolladores Mexicanos ubicado en¹⁵, Encuestas realizadas sobre ¿Qué Framework MVC prefieres para desarrollar?, nos habla que la tecnología con mayor preferencia es JSF.

La comunidad que participó en la misma menciona diferentes tecnologías, en esta encuesta participaron 665 integrantes, la cual presentaron los siguientes datos.

¹⁵http://www.javamexico.org/poll/que_framework_mvc_prefieres_para_desarrollar

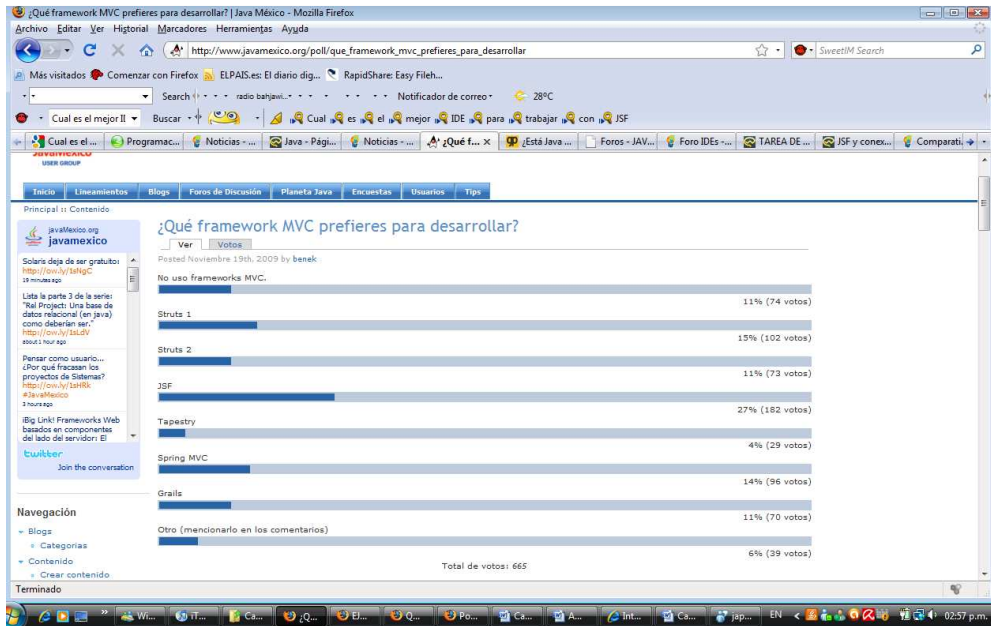


Figura III.16 Pantalla de la encuesta sobre los frameworks

Tabulación:

Tabla III.VII Tabulación de los frameworks.

Framework	Porcentaje
No Uso Framework MVC	11 %
Struts 1	15 %
Struts 2	11 %
JSF	27 %
Tapestry	4%
Spring MVC	14 %
Grails	11 %
otros	6 %

Según el blog ubicado en el sitio¹⁶, trata sobre son las 10 Razones para elegir ADF, en este blog nos muestra las ventajas de utilizar la tecnología ADF.

10 razones para elegir ADF Faces. Estas son:

¹⁶<http://blog.pucp.edu.pe/item/8284>

- 1.- Simplicidad
- 2.- Facilidad
- 3.- Flexibilidad
- 4.- Reusabilidad
- 5.- Arquitectura orientada a Servicios (SOA)
- 6.- Usabilidad
- 7.-Colaboración
- 8.- Set de Herramientas completo
- 9.- Confiabilidad
- 10.- Compatibilidad

Según el apartado ubicado en la siguiente dirección¹⁷, podemos notar que la filosofía de java es surgir y permitir al desarrollador un ambiente más rápido para el desarrollo, veremos entonces que jsf, adf y struts son técnicas de programación diferentes.

Muchas ventajas se menciona al trabajar con entornos java, es así que JSP, JSF, y ADF es una evolución de la tecnología proporcionando grandes ventajas como se muestra en el apartado¹⁸.

Por todo lo arriba mencionado para el análisis comparativo se ha seleccionado las tecnologías JSF y ADF por los siguientes motivos:

- ✓ **JAVA** es un lenguaje que nos permitirá introducirnos en el ambiente del software libre utilizando herramientas (IDEs y Servidores “free”).
Nos permitirá programar sistemas simples, complejos y de gran envergadura para escritorio, Web y con tecnología cliente-servidor.
- ✓ La estructura que JSF y ADF manejan dentro de su arquitectura se enmarca en el patrón MVC, lo cual permite brindar facilidad en la interacción de las capas. Dando al desarrollador un mejor entorno de trabajo.

¹⁷<http://www.temadictos.com/558662313-jsf-y-ADF-o-struts>

¹⁸http://www.bbr.cat/index2.php?option=com_content&do_pdf=1&id=171

- ✓ Estas tecnologías a diferencia de JSP permite reutilizar componentes con la técnica de drag and Drop, y disminuir por lo tanto la escritura de código.

3.3 ANÁLISIS DE LAS TECNOLOGÍAS

En este apartado describiremos y analizaremos las tecnologías a comparar para posteriormente establecer la más óptima a ser utilizada en nuestro sistema.

3.3.1 JSF

JSF es un marco de trabajo de componentes de interface de usuario del lado del servidor para aplicaciones Web basadas en Java. JSF contiene un API para representar componentes UI y manejar sus estados, manejar sus eventos, la validación del lado del servidor, y la conversión de datos, definir la navegación entre páginas, soportar internacionalización y accesibilidad; y proporcionar extensibilidad para todas estas características. También contienen dos librerías de etiquetas JSP personalizadas para expresar componentes UI dentro de una página JSP y para conectar componentes a objetos del lado del servidor.¹⁹

3.3.1.1 Arquitectura

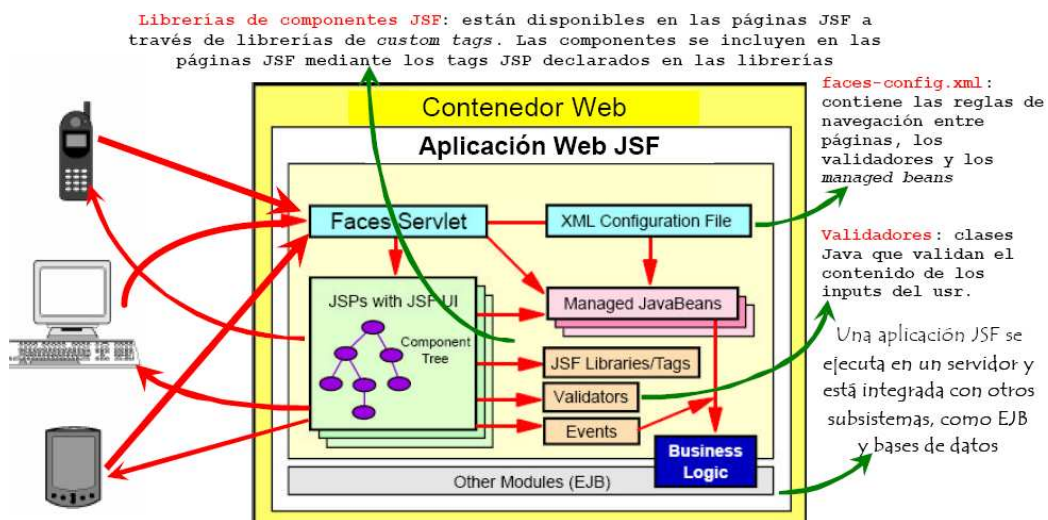


Figura III.17 Arquitectura de una aplicación JSF²⁰

¹⁹http://www.programacion.com/tutorial/jap_jsfwork/3/

²⁰<http://www.ing.unp.edu.ar/wicc2007/trabajos/ISBD/109.pdf>

De manera esquemática, el funcionamiento de la arquitectura de JSF es el siguiente:

La Figura [III.17] muestra los componentes principales del Framework JSF que intervienen en la construcción de una aplicación Web J2EE, y también su flexibilidad para aceptar peticiones provenientes de diferentes clientes. JSF implementa el patrón Front-Controller, que centraliza el manejo de peticiones provenientes de los clientes. En el caso de JSF, este controlador central, es un objeto servlet, llamado **FacesServlet**.

3.3.1.2 MVC con la tecnología JSF

MVC es el patrón de diseño arquitectural recomendado para aplicaciones interactivas Java. MVC separa los conceptos de diseño, y por lo tanto decrementa la duplicación de código, el centralizamiento del control y hace que la aplicación sea más extensible. MVC también ayuda a los desarrolladores con diferentes habilidades a enfocarse en sus habilidades principales y a colaborar a través de interfaces claramente definidos. MVC es el patrón de diseño arquitectural para la capa de presentación.

JSF encaja bien en la arquitectura de la capa de presentación basada en MVC. Ofrece una clara separación entre el comportamiento y la presentación. Une los familiares componentes UI con los conceptos de la capa-Web sin limitarnos a una tecnología de script o lenguaje de marcas particular.

Los beans que hay tras JSF son la capa de modelo (en una sección posterior hablaremos más de estos beans). También contienen acciones, que son una extensión de la capa del controlador y delegan las peticiones del usuario a la capa de la lógica-de-negocio. Observe, que desde la perspectiva de la arquitectura general de la aplicación, también se puede referir a la capa de lógica-de-negocio como la capa del modelo. Las páginas JSP con etiquetas JSF personalizadas son la capa de la vista. El Servlet `Faces` proporciona la funcionalidad del controlador.

3.3.1.3 JSF con su IDE

Una vez configurada la base de datos, creadas las entidades y las librerías, es importante crear la vista de la aplicación, para después ligarla con el modelo que acabamos de crear.

Netbeans tiene un conjunto de componentes de **Java Server Faces**, creado para facilitar notablemente su uso, cuenta con:

- ✓ Un Editor Visual para JSF.
- ✓ Componentes que son utilizados a través de uniones (bindings) a la base de datos.
- ✓ Un conjunto de componentes útiles como Tabs, botones de Imágenes, calendarios, etc.
- ✓ Un esquema de eventos fácil de usar.
- ✓ Un editor para las reglas de navegación de la aplicación, el cual se pueden ligar los eventos invocados por los componentes y el conjunto de páginas creadas en la aplicación.
- ✓ Control de los beans utilizados por los JSF que se encuentran en los scopes de:
 - Request
 - Session
 - Application

3.3.2 ADF

ADF es el Framework de desarrollo J2EE ofrecido por Oracle para la construcción rápida de aplicaciones Web robustas, reduciendo la codificación por medio de librerías y asistentes.

3.3.2.1 Arquitectura

ADF (Application Developer Framework) es la tecnología que engloba a ADF Business Component, ADF Binding, ADF Struts, etc.

ADF presenta la siguiente Arquitectura

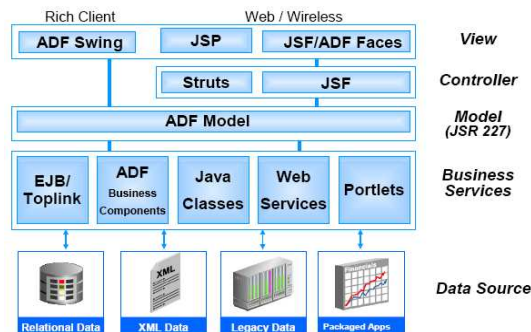


Figura III.18 Arquitectura de ADF

La capa de Business Service contempla:

- ✓ Acceso a datos
 - O/R Mapping
 - Queries/DML
- ✓ Lógica de negocio
 - Validación de datos
 - Lógica de negocios
- ✓ Elecciones de implementación
 - EJB, POJO, Web Services, XML, TopLink, ADF

La capa de View contempla:

- ✓ Elección de tecnologías
 - JavaServer Faces (JSF)
 - Swing
 - JSP
 - JSP/Struts
- ✓ Editor visual de layout
- ✓ Fácil data binding

La Capa de Modelado ADF contempla:

- ✓ ADF model
 - JSR-227 Data Controls
- ✓ Interfaz orientada a servicios
 - Implementación abstracta desde clientes
 - Acoplamiento ligero entre servicio y aplicaciones
- ✓ Más reusable
 - Descubre y comparte servicios
- ✓ Más productividad
 - Drag and Drop Data Binding

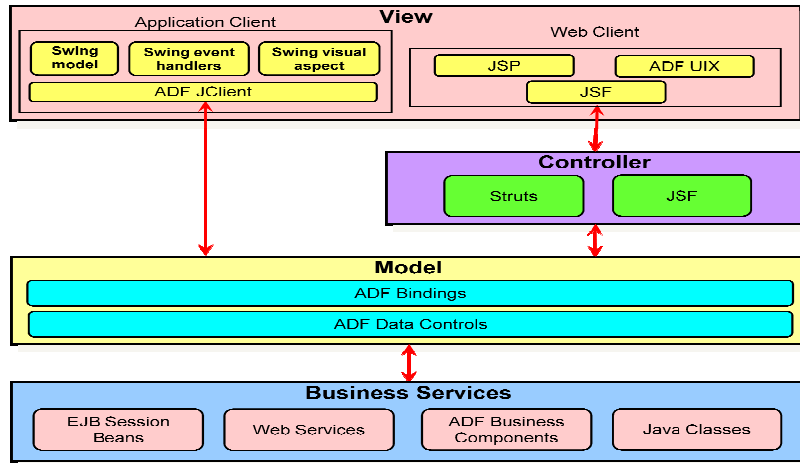


Figura III.19 Business Services

ADF Data Binding: Arrastrar y soltar servicios de negocio a la vista y al controlador.

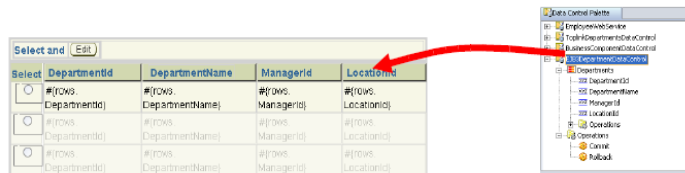


Figura III.20 Data Binding

3.3.2.2 MVC con la tecnología ADF

La tecnología ADF permite crear aplicaciones J2EE basada en el patrón MVC, en muy poco tiempo, sin escribir una sola línea de código Java. Utilizando una base de datos MySQL, ADF le ha permitido crear un sistema de reservas totalmente funcional que se ha desplegado en el servidor OC4J de Oracle.

Las aplicaciones desarrolladas por ADF no están confinadas al servidor de aplicaciones de Oracle. JDeveloper proporciona soporte para instalar las librerías ADF y para empaquetar y desplegar una completa aplicación ADF en otro servidor de aplicaciones como JBoss.

ADF Implementa el patrón MVC

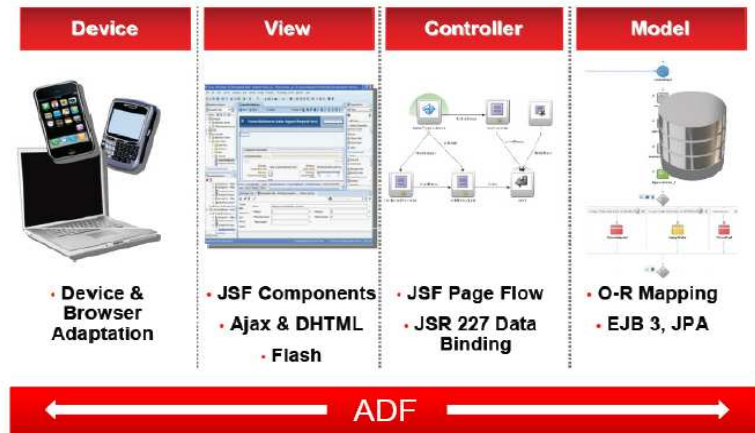


Figura III.21 ADF Implementa el patrón MVC

Modelo

Entidades: Un objeto entidad (*Entity*) de ADF representa una entidad de negocio.

En ADF, la entidad maneja otras tareas, como las validaciones de negocio y las máscaras de formato. En la mayoría de los casos, el modelo es el lugar más lógico para dirigir esos conceptos porque permitirá una máxima reutilización. Las validaciones, mostradas por requerimientos de negocio, se supone que son importantes sin importar como se utilice el modelo.

Vista

Un objeto vista de ADF expone un objeto entidad a los usuarios de la aplicación. Se puede acceder a una entidad desde varios sitios, pero el contexto en el que se utiliza variará. Se pueden establecer filtrados, ordenaciones y relaciones entre objetos mediante un objeto vista, asegurando que el modelo se representa en la forma en la que tiene más sentido.

Módulo de aplicación

El módulo de aplicación ADF empaqueta varios componentes vista mientras proporciona acceso a su capa de servicios de negocio, que podría utilizar tecnologías como EJB para manejar su base de datos. El módulo de aplicación proporciona vistas ADF con soporte de transacciones, y otros servicios importantes centrados en los datos.

Vista y Controlador Nada en su modelo debería dictar una tecnología de vista particular. Un módulo de aplicación puede servir como base para una aplicación Swing o para una aplicación Web basada en Struts. Si desarrolla una aplicación ligera, los componentes Swing se podrían comunicar directamente con el módulo de aplicación. Cuando desarrolle una aplicación basada en la Web, su aplicación usará un controlador claramente definido para manejar la interacción entre el modelo y la vista.

Controlador

En versiones anteriores, ADF proporcionaba su propio controlador. Ese controlador ha sido reemplazado en favor de Struts. Struts utiliza un fichero XML, llamado `struts-config.xml`, para manejar las peticiones de la aplicación y despacharlas a los manejadores adecuados. Los desarrolladores con experiencia en Struts apreciarán la posibilidad de utilizar esos conocimientos. Mas en estas tecnologías, la capa controlador es creada y manejada internamente.

3.3.2.3 ADF con su IDE

JDeveloper es un entorno de desarrollo integrado desarrollado por Oracle Corporation para los lenguajes Java, HTML, XML, SQL, PL/SQL, Javascript, PHP, Oracle ADF, UML y otros.

El IDE JDeveloper es una herramienta de desarrollo de ADF, las cuales presentan las siguientes ventajas de uso:

- ✓ Soporta todo el ciclo de desarrollo en J2EE
 - Diferentes formas de despliegue
 - Aplicación Java, applet, JSP, servlet, EJB
- ✓ Fácil desarrollo, depuración y despliegue
- ✓ Framework de acceso a datos: Business Components (BC4J y ADF BC)

3.4 DETERMINACIÓN DE LOS PARÁMETROS DE COMPARACIÓN.

3.4.1 Enumerar todos los parámetros posibles.

Tabla III.VIII Parámetros posibles a comparar

Parámetros Generales		
Rendimiento	<ul style="list-style-type: none"> ✓ Rendimiento Independiente ✓ Rendimiento con Base de Datos 	<ul style="list-style-type: none"> ○ <i>Uso de Recursos</i> ○ <i>Capacidad de Respuesta</i> ○ <i>Requerimientos Mínimos</i> ○ <i>Escalabilidad.</i>
Desarrollo	<ul style="list-style-type: none"> ✓ Implementación de código embebido ✓ Interfaz de Usuario Accesibilidad ✓ Uso de Componentes 	
Producto	<ul style="list-style-type: none"> ✓ Madurez del producto(Versiones) ✓ Licencia Libre. ✓ Costos 	
Compatibilidad.	<ul style="list-style-type: none"> ✓ Portabilidad ✓ Navegadores Web 	
Patrón de Diseño	<ul style="list-style-type: none"> ✓ Modelo MVC 	
Cumplimiento de estándares	<ul style="list-style-type: none"> ✓ Especificaciones JSR 	
Arquitectura	<ul style="list-style-type: none"> ✓ Soporte de Servidores Web ✓ Soporte de Multicapa 	
Manejo de Seguridad	<ul style="list-style-type: none"> ✓ A nivel de la Aplicación ✓ A nivel de Base de Datos ✓ A nivel del Servidor Web 	
Otros	<ul style="list-style-type: none"> ✓ Simplicidad ✓ Facilidad 	

	<ul style="list-style-type: none">✓ Flexibilidad✓ Reusabilidad✓ Arquitectura orientada a Servicios (SOA✓ Usabilidad✓ Colaboración✓ Set de Herramientas completo✓ Confiabilidad✓ Compatibilidad
--	--

3.4.2 Selección de los parámetros a comparar.

Para realizar el análisis cuantitativo y cualitativo de Tecnologías JSF y ADF, que permiten la construcción de aplicaciones Web, se ha tomado en cuenta varios aspectos relevantes, del listado mostrado anteriormente procedemos a seleccionar en lo que se ha considerado los siguientes parámetros de comparación.

Tabla III.IX Parámetros a Comparar

Parámetro	Justificación
<i>Patrón de Diseño MVC</i>	El patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos, es el mismo que caracteriza las tecnologías JSF y ADF.
<i>Seguridad</i>	Es importante mantener seguridades en las aplicaciones Web, para de esta manera incrementar la confiabilidad del usuario en la navegación por nuestra página. Ciertos datos necesitan ser privados.
<i>Producto</i>	Es fundamental, analizar el entorno usado para el desarrollo de las aplicaciones Web, para ponerlas en producción.

3.4.3 Determinación de variables e indicadores a valorar

Tabla III.X Variables e Indicadores a Valorar

Parámetros	Variables	Indicadores
Patrón de Diseño MVC	Modelo	Conexión Visual a datos Soporte para múltiples Bases de Datos. Manipulación de base de datos Desempeño con la base de datos
	Vista	Diseñador GUI Visual Facilidad de aprendizaje y uso de los elementos gráficos Existencia de herramientas de ayuda y consulta Herramientas de navegación de forma visual y grafica Fusión de código y diseño.
	Controlador	Generación automática de etiquetas de formulario. Validaciones de Formulario. Manejo visual de componentes de un formulario. Manejo de controles personalizados.
Seguridad	A nivel de Aplicación	Variable Uso desesiones. Manejo de Cookies. Encriptación de datos. Validación de datos.
Producto	Madurez del Producto	Especificaciones Licenciamiento Costos
	Instalación	JDK Contenedor de aplicaciones Entorno de Desarrollo Versionamiento

3.4.4 Descripción de los parámetros a valorar

MANEJO DEL PATRÓN DE DISEÑO MVC

Es un patrón de arquitectura de desarrollo software que separa las aplicaciones en tres capas diferenciadas: datos (Modelo), interfaz de usuario (Vista) y lógica de control (Controlador). Esto

permite una separación entre la interfaz de usuario y la lógica que hace que el mantenimiento de las aplicaciones Web sea sencillo. [8]

Para la evaluación del manejo de Patrón Modelo Vista y Controlador se ha tomado como referencia las siguientes variables:

- ✓ **Modelo:** Esta es la representación específica de la información con la cual el sistema opera, esta capa típicamente representa objetos almacenados en nuestras fuentes de datos, por lo que se realizara un análisis sobre la conectividad que consiste en la conexión con bases de datos dentro de una aplicación Java usando Java Database Connection (JDBC).
JDBC es una especificación de un conjunto de clases y métodos de operación que permiten a cualquier programa Java acceder a sistemas de bases de datos de forma homogénea. [16]
- ✓ **Vista:** Manejo de los componentes de Interfaz de Grafica para el diseño de páginas.
- ✓ **Controlador:** Responde a eventos del usuario e invoca cambios en el modelo y probablemente en la vista.

SEGURIDAD

- ✓ **A Nivel de Aplicación:** Permite que la aplicación proporcione confianza al usuario, y de esta manera no existan ataques a la funcionalidad del sistema. Mediante el manejo de roles dinámicos en la aplicación (es decir solo usuarios privilegiados tienen acceso a ciertas características de la aplicación, y estas pueden ser auditadas).
Mediante la Autenticación que es la que explota el método de validación de la identidad de un usuario, servicio o aplicación.
- ✓ **A Nivel de Base de Datos:** La seguridad es algo que nunca se debe dejar pasar por alto en una aplicación web puesto que la información que se maneja está a disposición de los usuarios de la web, por lo cual es de suma importancia mantener un nivel de seguridad para evitar pérdidas o violaciones en nuestra información. Para evaluar este parámetro tomaremos en cuenta los principales aspectos que intervienen en la seguridad a nivel de Base de datos.

Hoy en día, las bases de datos son componentes cardinales de cualquier aplicación basada en web, permitiendo que los sitios web provean contenido dinámico, debido a que la información considerablemente sensible o secreta puede ser almacenada en tales bases de datos, se debe considerar seriamente la forma de protegerlas. Existen varias formas de seguridad por lo que nos centraremos en la creación de login para la Base de datos, usuario con derecho a crear, borrar y modificar objetos y que además puede conceder privilegios a otros usuarios sobre los objetos que ha creado o usuario con derecho a consultar, o actualizar, y sin derecho a crear o borrar objetos.

PRODUCTO

- ✓ **Madurez del Producto:** Se basa en las especificaciones que cada tecnología usa, pues día a día surgen nuevas librerías o ideas para poder mejorar su funcionalidad.

Una licencia es aquella autorización formal con carácter contractual que un autor de un software da a un interesado para ejercer "actos de explotación legales". Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el licenciatarlo.

La distribución de los productos, derechos de uso de las herramientas, tecnologías y formas de licenciamiento son elementos que deben ser considerados en función del alcance la solución, dato que este es un tema que afecta directamente el presupuesto de los proyectos de desarrollo.

Los Costos criterio hace referencia directa al precio de las licencias que debe pagar el usuario por la utilización de las herramientas de desarrollo, para la creación de aplicaciones bajo los estándares J2EE. Si se considera la realización de paquetes independientes para el nivel del usuario.

- ✓ **Instalación:** JDK, representa las versiones de Java. El contenedor de aplicaciones o servidor, nos permitirá ejecutar la aplicación, el mismo que debe soportar las versiones de las tecnologías utilizadas, pues no cualquier servidor las acepta.

Un servidor de aplicaciones generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negocios y de acceso a los datos de la aplicación.

Para el desarrollo de las aplicaciones Web con determinadas tecnologías es necesario el uso de las IDEs que son entornos de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

3.4.5 Determinar la importancia de los parámetros

Tabla III.XI Importancia de Parámetros a Comparar

Parámetro	Valoración
<i>Patrón de Diseño MVC</i>	40%
<i>Seguridad</i>	30%
<i>Producto</i>	30%

3.5 DESCRIPCIÓN DE LOS MÓDULOS DE PRUEBA

3.5.1 Modulo 1

El modulo 1 será desarrollado para probar el parámetro: *Manejo del Patrón MVC*.

Modelo:

Conectividad con la Base de Datos, se analizará el uso de las directivas que se emplean para la conexión respectiva. Pues en este sub módulo se va a crear una página que desplegará el listado de los Tipos de Activos fijos, con los que se calculan las depreciaciones del departamento de contabilidad del IESS.

Mediante la consulta a una base de datos en MySQL, esto nos permitirá valorar el manejo de acceso y presentación de datos de cada una de las tecnologías que se están analizando.

Vista

La capa de presentación, nos mostrará el soporte de los componentes de IU.

El desarrollo de este sub modulo comprende la creación una página del sistema SADIESS, la cual consta de varios componentes de interfaz de usuario como imágenes, botones, etc. mediante la construcción de este modulo podremos evaluar la facilidad que brinda cada tecnología para el desarrollo Web.

Controlador

Este submodulo consta de una página la cual permite apreciar las validaciones de los campos vacios, pues sin su respectivo control no podrá acceder a la acción.

3.5.2 Modulo 2

El modulo2 será desarrollado para probar el parámetro: *Seguridad*.

Este modulo consta de la creación de un login para los usuarios que maneja la aplicación Web mediante el uso de sesiones, protección de datos que viajan en la Web como las contraseñas, lo cual nos permitirá evaluar el manejo y tipos de seguridades que proporcionan las tecnologías que están siendo evaluadas.

3.5.3 Modulo 3

El modulo 3 será desarrollado para probar el parámetro: *Producto*

Este modulo consta de cuadros comparativos que reflejan las características de las Tecnologías. Lo cual nos permitirá determinar las ventajas de producción final.

3.6 DESARROLLO DE LOS MÓDULOS DE PRUEBA

3.6.1 DESARROLLO DE LOS MÓDULOS DE PRUEBA CON LA TECNOLOGÍA JSF

3.6.1.1 Modulo 1

MODELO

En este modulo vamos a utilizar los elementos que la tecnología JSF nos proporciona para el acceso y manipulación de base de datos.

Para lo cual creamos un nuevo proyectoJSF en Netbeans, a continuación realizamos la conexión a la base de datos, NetBeans permite una conexión visual a la base de datos mysql usando los elementos que se encuentran en la ventana de aplicación/base de datos.

El driver apunta directamente al puerto 3306, que es de Mysql.

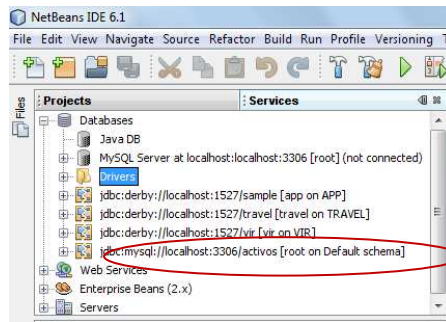


Figura III.22 Panel con el driver de Mysql

Escogemos el servidor el usuario y a la base de datos a la cual nos queremos conectar y elegimos la opción aceptar.

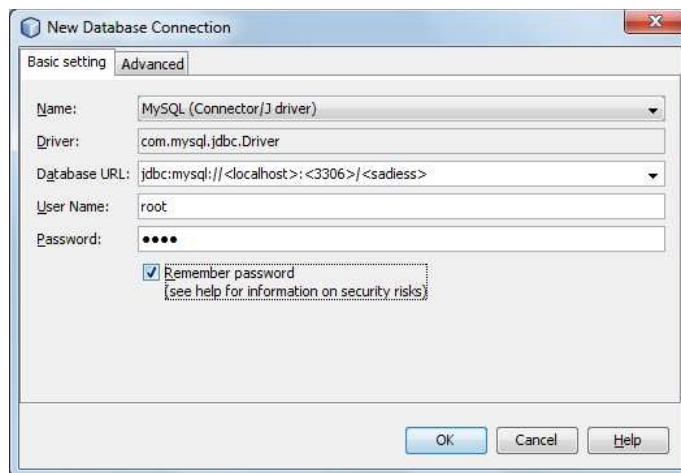


Figura III.23 Conexión visual a una base de datos en Netbeans

Una vez que haya realizado el ítem anterior podemos acceder directamente en la ventana de aplicación/base de datos a los todos los elementos de la base de datos seleccionada.

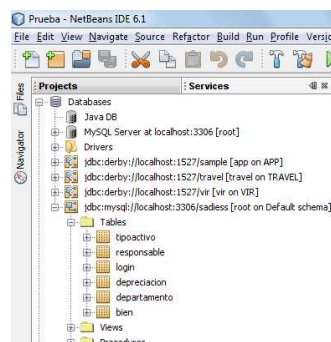


Figura III.24 Ventana de aplicación de base de datos en NetBeans

A continuación insertamos un nuevo registro haciendo uso de NetBeans para proceder a seleccionar los tipos de activos fijos y los años de Vida útil.

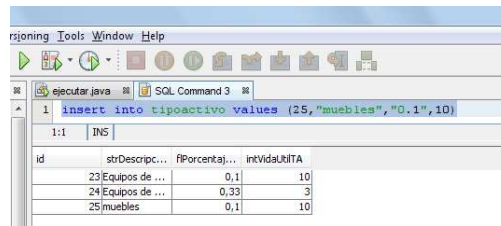


Figura III.25 Inserción de juego de registro en NetBeans

A continuación mediante el uso de una sentencia sql, usando código java desde la clase Ejecutar obtendremos la consulta, con la selección de los Tipos de Activos fijos y los años de vida útil.

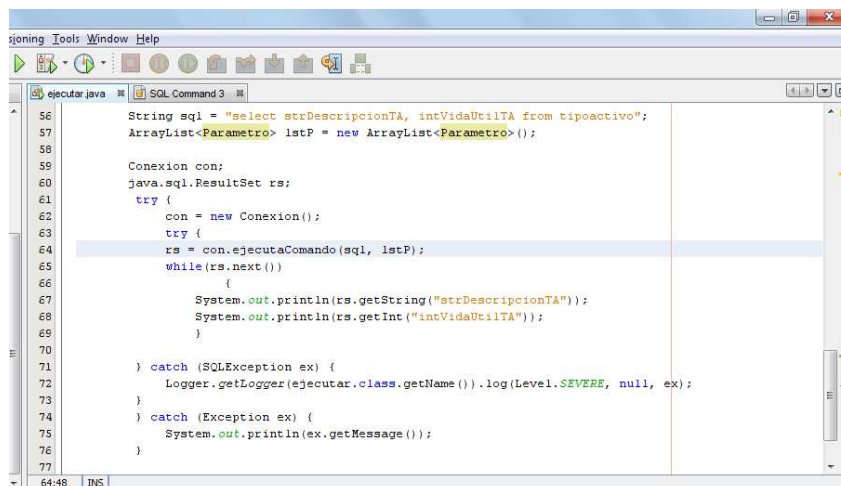


Figura III.26 Pantalla de Consulta

Para mostrar el resultado de hace uso del compilador de NetBeans y listamos el tipo de Activo Fijo, seguido del número de años de vida útil del mismo.

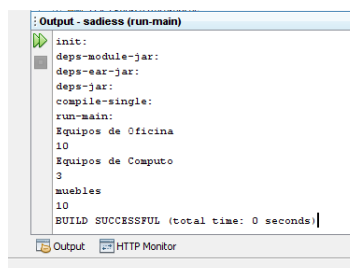


Figura III.27 Resultado de la consulta

VISTA

Para la construcción del submódulo Vista se va hacer uso de los componentes que proporciona NetBeans cuando se usa la tecnología JSF como son botones, etiquetas, y otros componentes más.

Los mismos que proporcionan una gran facilidad de trabajar integradamente con la tecnología. Los pasos para desarrollar este módulo en NetBeans usando la tecnología JSF se describe a continuación.

Primero creamos una nueva aplicación Web en NetBeans, seleccionamos la tecnología JSF, y empezamos a arrastrar los componentes para que sean parte de la página.

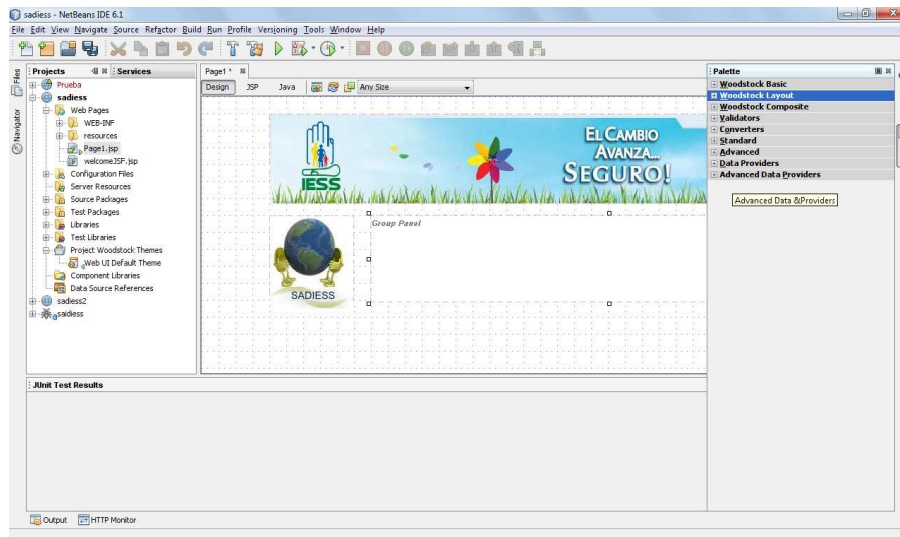


Figura III.28 Crear documento en NetBeans + JSF

NetBeans proporciona la opción de código y diseño lo cual nos permite ir verificando como va quedando nuestra página Web, haciendo uso de esto mientras se desarrollaba el modulo realizamos algunos cambios en adherir las imágenes que forman parte de las páginas de la aplicación.

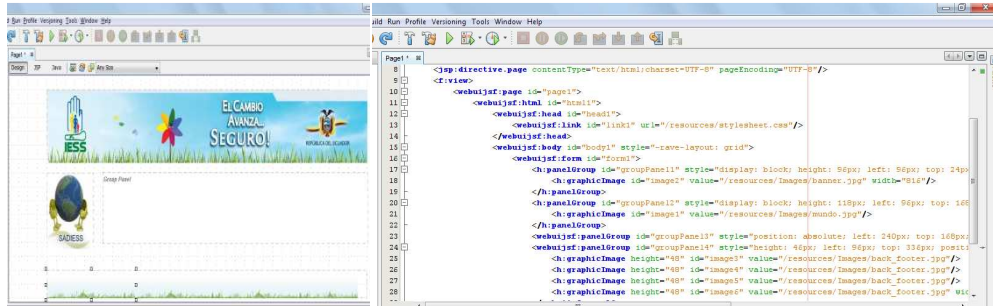


Figura III.29 Figura Diseño y código

CONTROLADOR

El submodulo controlador, será construido de manera que podamos notar las validaciones antes de la acción de una página. Para lo cual seguiremos los siguientes pasos:

Creamos una aplicación Web en NetBeans usando la tecnología JSF, usamos las capas anteriores Modelo, Vista para acceder a registrar un nuevo administrador.

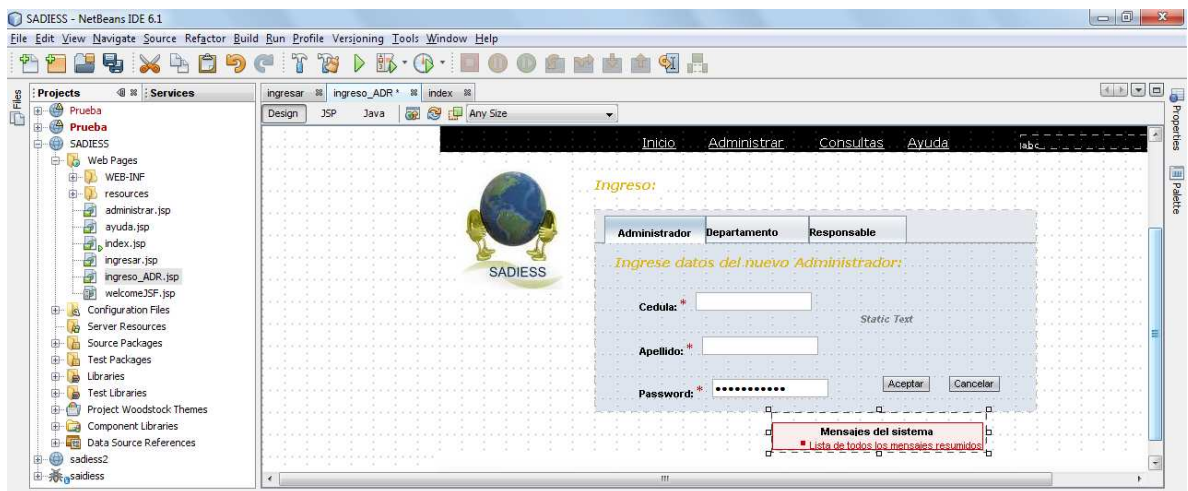


Figura III.30 Pantalla con los campos restringidos los espacios vacios

3.6.1.2 Modulo 2

Para el desarrollo de este módulo haremos uso de la autenticación de usuario.

Como ya contamos con la conexión a la base de datos, en la capa Modelo, procederemos a crear los campos en el formulario y aplicamos las validaciones.

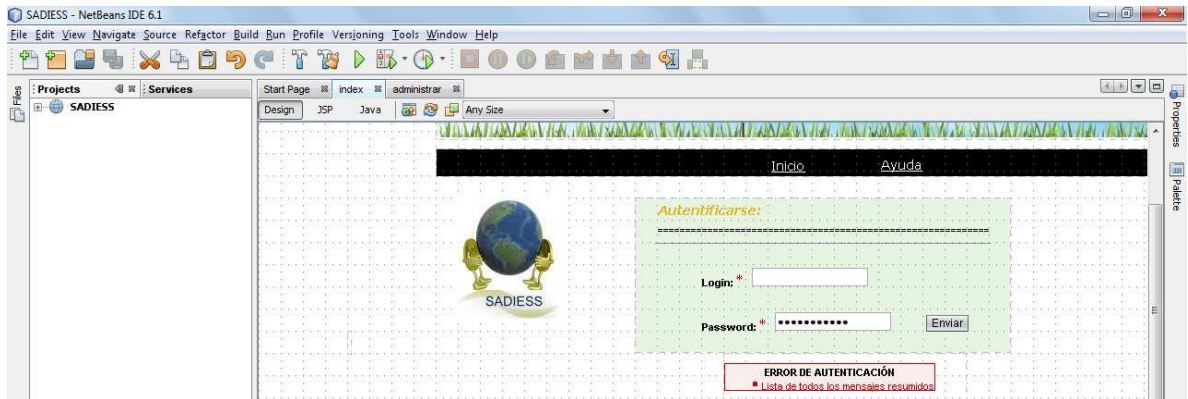


Figura III.31 Propiedades de la capa Controlador.

Y listo al ingresar a la página obtenemos ya nuestras variables de Sesión, las mismas que se crearan cada vez que el usuario inicie sesión.



Figura III.32 Pantalla Final modulo 2

Mediante el mismo menú de autenticar usuario podemos desconectar un usuario y restringir páginas.

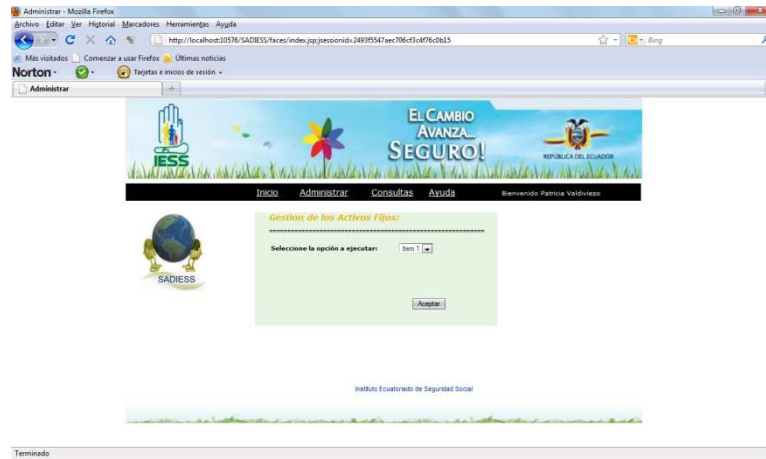


Figura III.33 Pantalla Verificando el inicio de sesión

3.6.1.3 Modulo 3

PRODUCTO

Madurez del Producto: Las versiones existentes de esta tecnología son: JSF 1.0, 1.1, 1.2 y 2.0

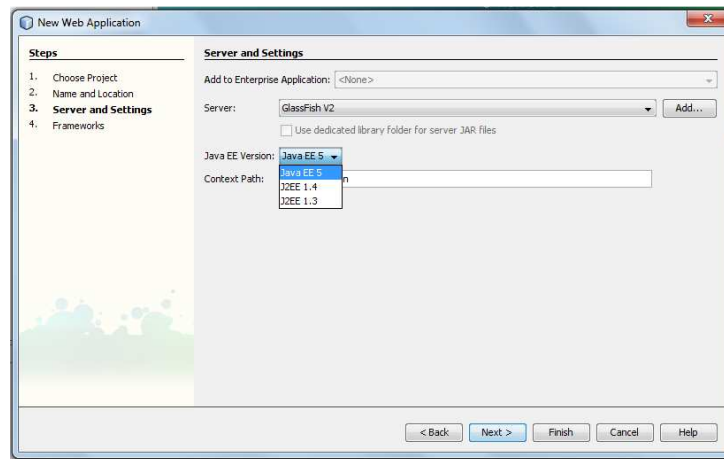


Figura III.34 Pantalla Verificando la utilización de la versión

Costos:Baja Inversión, para el momento de desarrollo y al ponerla en producción las herramientas y la tecnología misma, son sin costo para quienes la usan.

Licenciamiento:El costo se relaciona con el licenciamiento, hoy en día JSF tiene licencia libre.

INSTALACION

JDK (Java Development Kit):

Contenedor de Aplicaciones: Puerto 10576

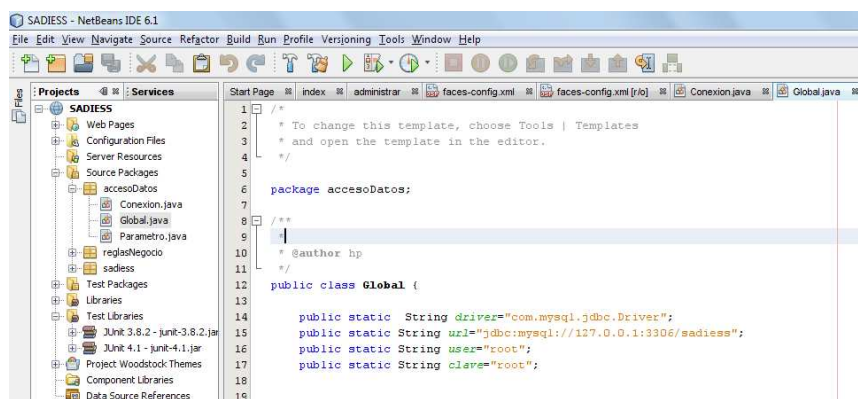


Figura III.35 Utilización del puerto en JSF

Entorno de Desarrollo: Hasta el momento su entorno de desarrollo más factible es NetBeans, sin embargo puede hacerlo en otros IDE's.



Figura III.36 Pantalla inicial del IDE NetBeans

La especificación o estándar: Entre los estándares que hace uso la tecnología JSF están: JSF (JSR-127): Trinidad, (JSR-252): JSF 1.2, (JSR-314): JSF 2.0;

3.6.2 DESARROLLO DE LOS MÓDULOS DE PRUEBA CON LA TECNOLOGÍA ADF

3.6.2.1 Modulo 1

Para la construcción del modulo se va hacer uso de varias herramientas adicionales como es JDeveloper y Mysql ya que estas nos permitirán trabajar en el desarrollo de una página web

de interfaz muy amigable para lo cual la tecnología ADF proporciona una gran facilidad de trabajar integradamente con estas herramientas.

Este modulo se divide en tres partes: capa de modelo que es el Acceso a Datos y la capa de Vista (Interfaz) y Controlador (reglas de negocio).

MODELO

En este modulo para la capa Modelo vamos a utilizar elementos que nos ofrece la tecnología ADF para el acceso y manipulación de la base de datos.

Creemos un nuevo proyecto de tipo Web Application ADF

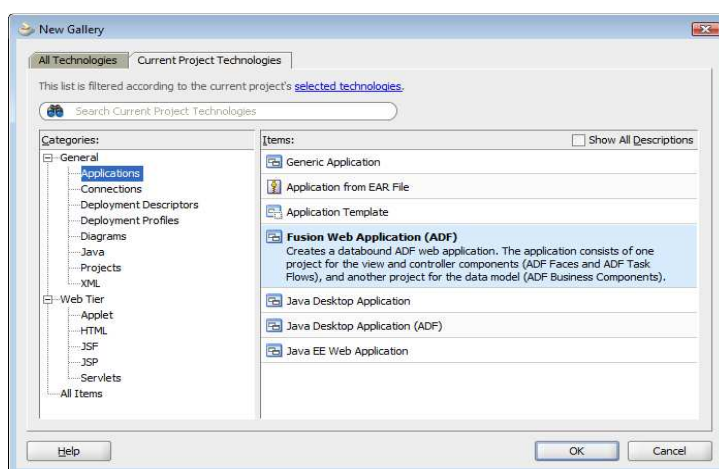


Figura III.34 Ventana para crear una nueva aplicación

Realizamos la conexión a la base de datos, el entorno de Desarrollo JDeveloper permite una conexión visual a la base de datos MySQL usando los elementos que se encuentran en la ventana de Database Navigator/IDE Connections.



Figura III.35 Panel para conexión Visual a una base de datos utilizando JDeveloper

Click derecho sobre IDE Connections/New connection

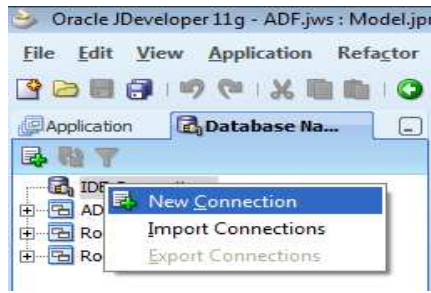


Figura III.36 Ventana para crear una nueva conexión.

Escogemos el servidor el usuario y a la base de datos a la cual nos queremos conectar y elegimos la opción aceptar.

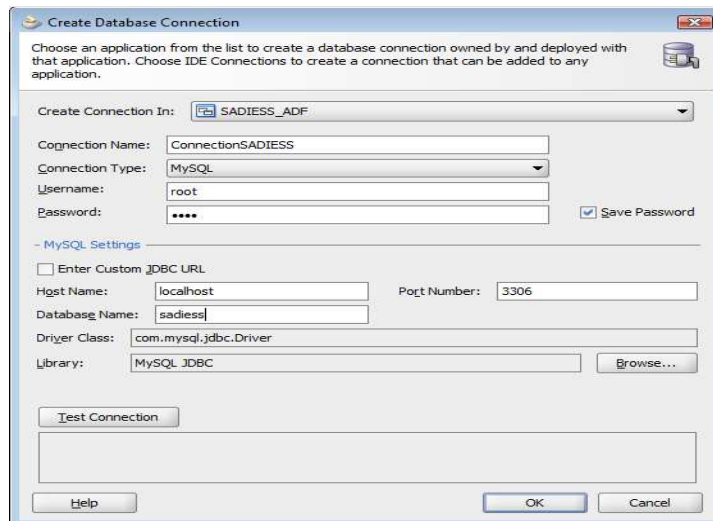


Figura III.37 Conexión Visual a una base de datos

Una vez realizado el ítem anterior podemos acceder directamente a la ventana Database Navigator/Seleccionar el nombre de la aplicación y elegir la base de datos.

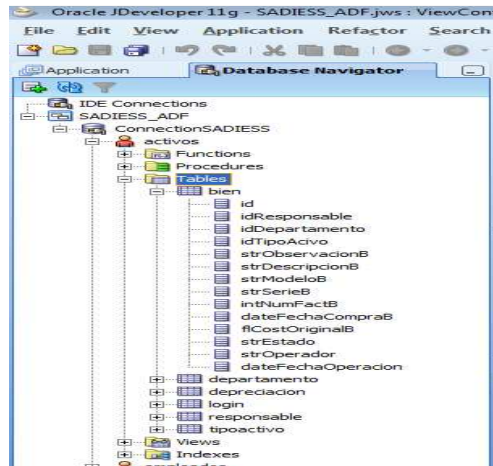


Figura III.38 Ventana de aplicación de base de datos

Para la creación de la página que desplegará el listado de los Tipos de Activos fijos, insertamos una nueva página arrastrando de la paleta de componentes ADF Task Flow/ View.

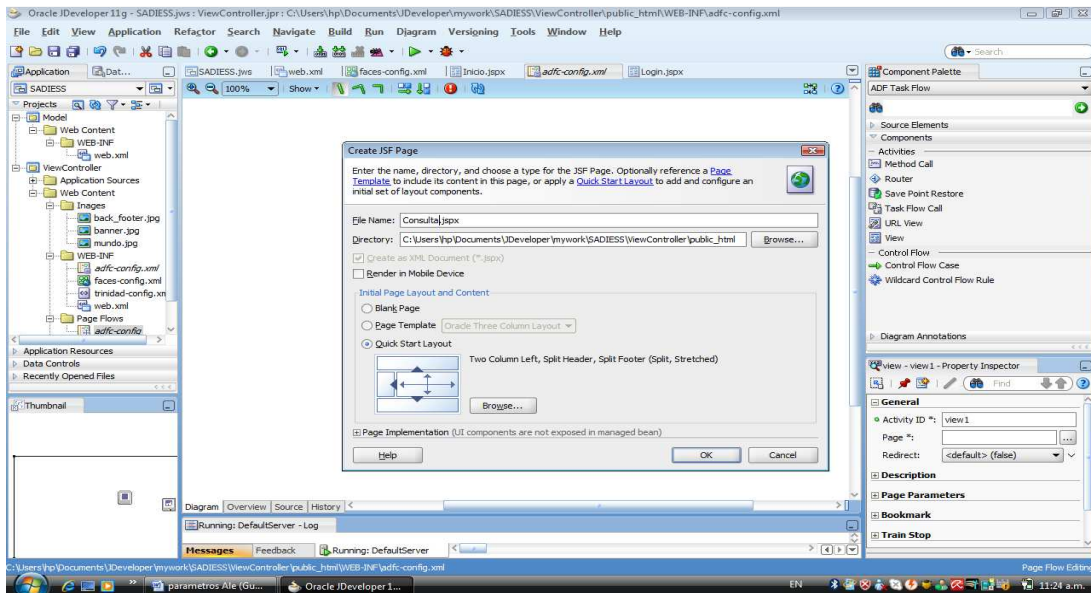


Figura III.39 Ventana de Insertar una nueva página View

La tecnología ADF al momento de crear una aplicación web automáticamente nos crea el patrón MVC por separado como es: model y ViewController.

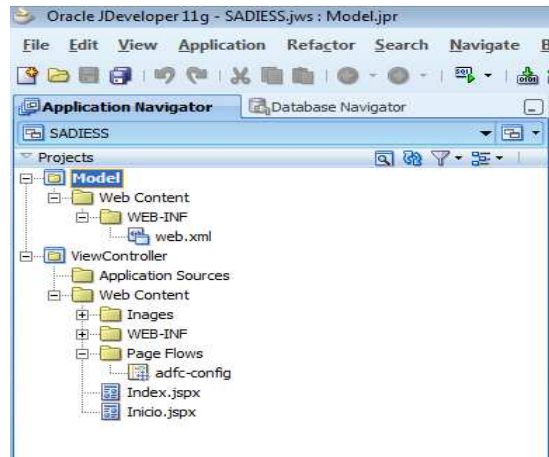


Figura III.40 Ventana para ver la división del patrón MVC

Para mostrar el resultado se hace uso de ciertos componentes para añadir datos a una tabla donde se mostrara el listado de los Tipos de Activos fijos.

Resultado final:

VISTA

Para la construcción del módulo/Vista se va hacer uso de la herramienta de desarrollo JDeveloper y de la paleta de componentes que ofrece como lo es ADF Face o ADF Task Flow ya que estas nos permitirán hacer una página web de interfaz muy amigable para lo cual la tecnología proporciona una gran facilidad de trabajar integradamente, puesto que viene con componentes que permiten acceder y editar elementos de este tipo desde el mismo entorno de desarrollo. Los pasos para que desarrollaren este módulo se describe a continuación. Primero creamos una nueva página que viene incorporada en la misma paleta llamada View, al momento de crear nos da la alternativa de elegir una categoría de la galería de componentes.

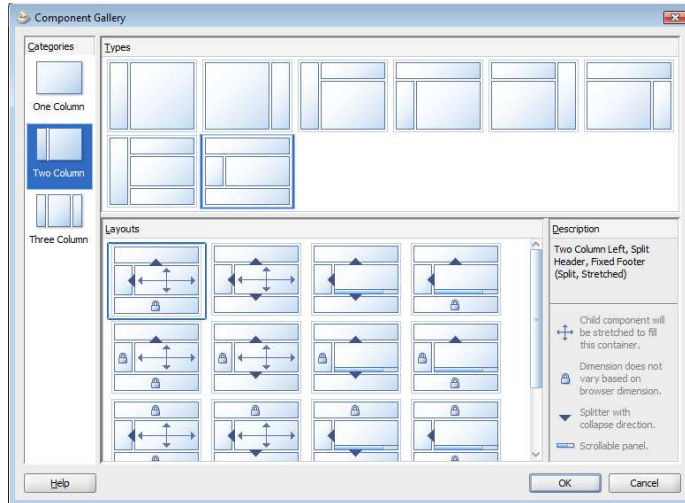


Figura III.41 Crear una nueva página View utilizando la paleta de componentes

A continuación comenzamos creando la cabecera insertando un elemento imagen, usando los componentes que se encuentra en la paleta de componentes.

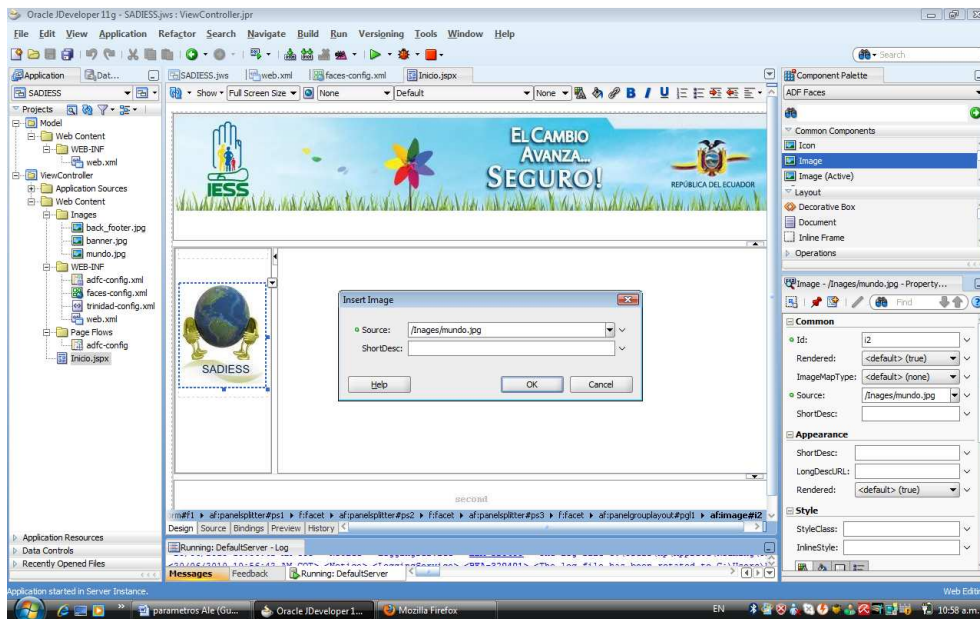


Figura III.42 Agregando elementos Imagen.

Para crear un menú agradable en la parte superior de la página podemos usar el elemento go Link, menú ADF y entre otras, Ahora solo es cuestión de ir añadiendo los demás elementos de la página principal como insertar texto, componentes como text input, button, calendar, panel entre otros.

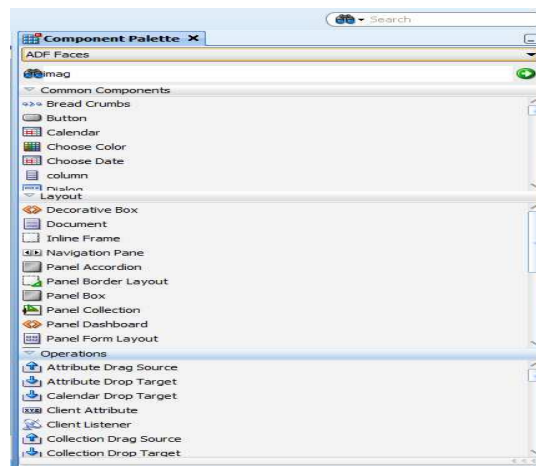
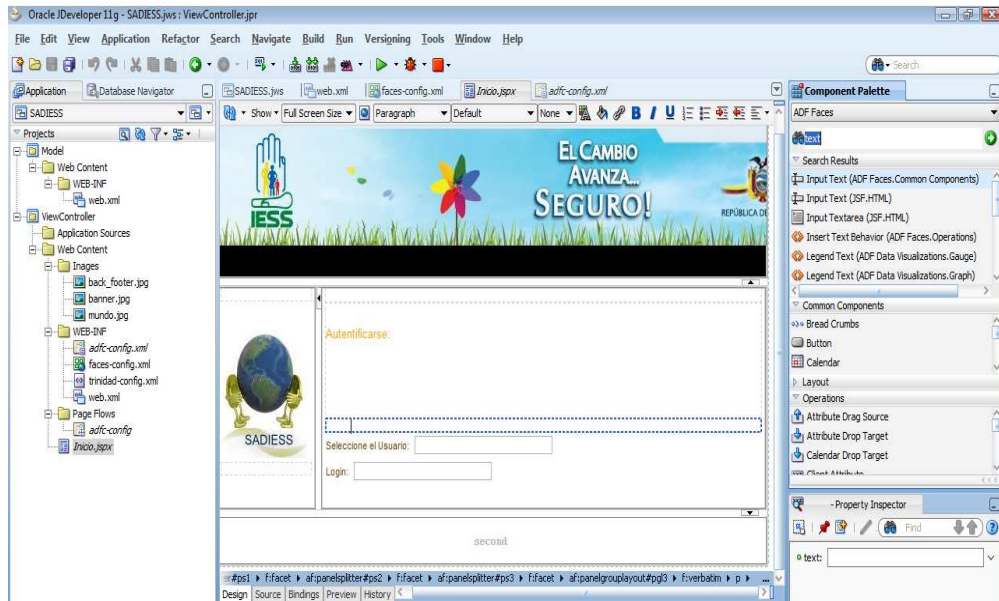


Figura III.43 Paleta de Componentes

A pesar de que vamos viendo de forma cómo va quedando la página al momento de desarrollarla tenemos que verificar nuestro producto final para lo cual vamos a usar un servidor web local en este caso usa puerto 7101, lo ejecutamos y la página final es la siguiente.

CONTROLADOR

3.6.2.2 Modulo 2

Para el desarrollo de este módulo sobre la autenticación de usuario se seguirá los siguientes pasos:

Primero creamos un New View que se encuentra en la paleta ADF Task Flow, pero antes nos ubicamos en la carpeta page Flow/ doble clic en adf-config y agregamos un View la llamaremos login.jspx.

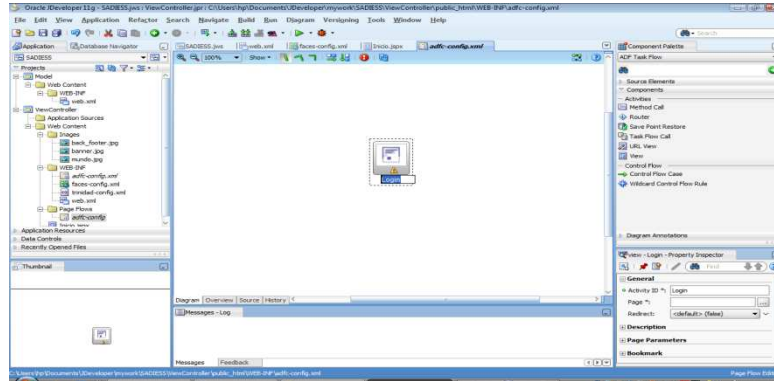


Figura III.44 Crear una nueva página View

Procedemos a diseñar con los componentes que nos ofrecen. Creamos 2 Input Text y 1 Password Field.

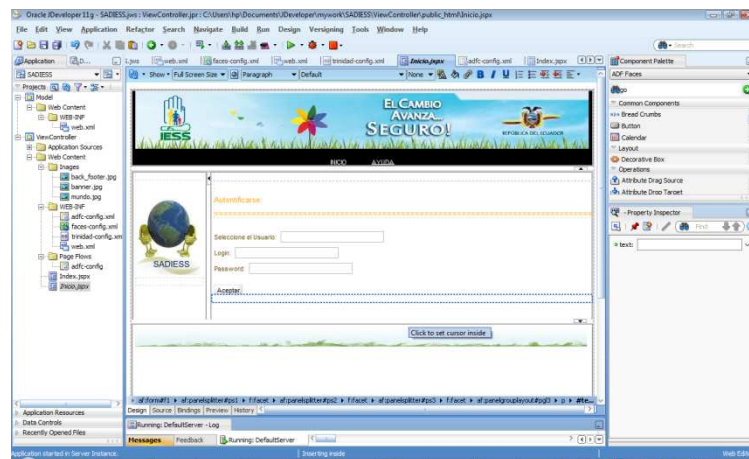


Figura III.45 Componentes más Validaciones

Realizamos las respectivas conexiones para la autenticación.

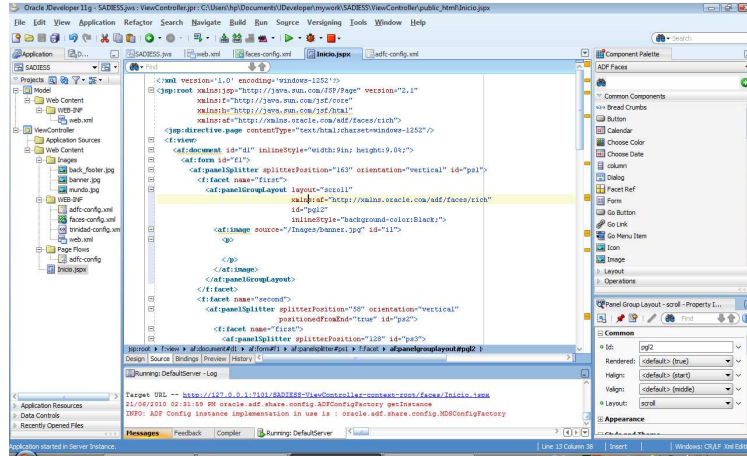


Figura III.46 Conexión mediante código

Y listo esta tecnología habrá creado automáticamente las sesiones que se crearan cada que un usuario se autentifique.

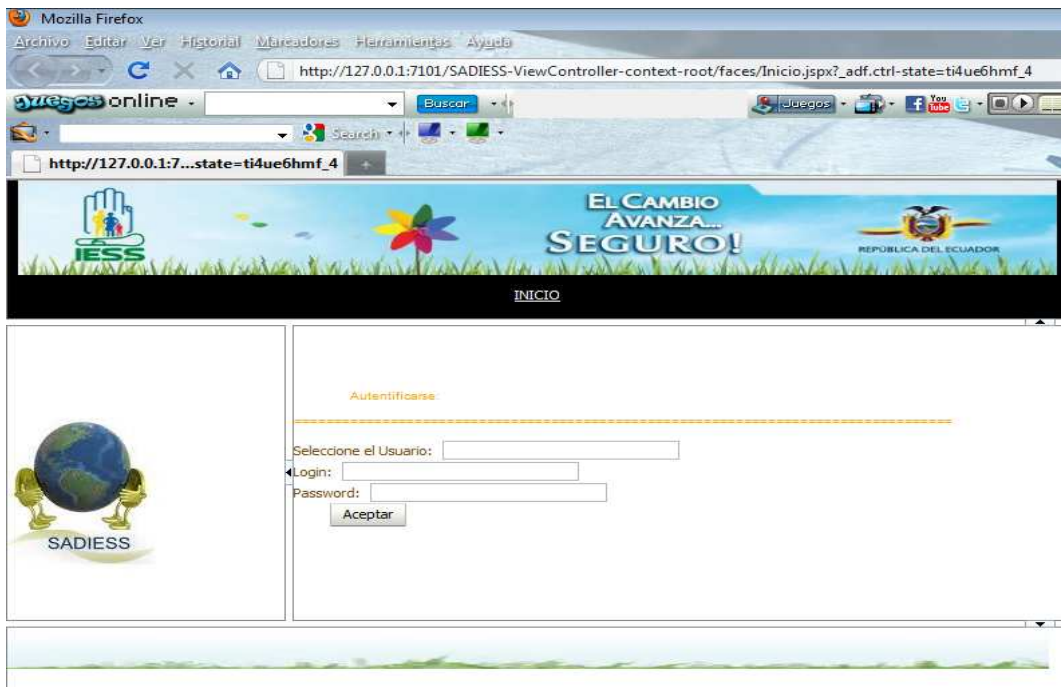


Figura III.47 Pantalla final del modulo 2/Autenticación.

Mediante la autenticación podemos restringir ciertas páginas.

3.6.2.3 Modulo 3

PRODUCTO

Madurez del Producto

Las versiones existentes de esta tecnología son: ADF 10 y ADF 11

Costos: Alta Inversión, para el momento de desarrollo prácticamente es gratuita pero al momento de la producción tiene su costo que depende del costo de la licencia de Oracle Application Server.

Licenciamiento: La licencia de Oracle ADF se incluye en la licencia de Oracle Application Server (OC4J). Esto significa que Oracle ADF es gratuita para todos los usuarios que poseen una licencia de Oracle Application Server. El desarrollo con ADF no tiene licenciamiento mientras dure el desarrollo, al momento de desplegar aplicaciones construidas con ADF en ambientes de producción la regla es:

Si se despliega en un servidor de aplicaciones Oracle el uso de ADF está incluido.

Si se despliega en otros servidores de aplicaciones, el uso de ADF tiene un licenciamiento.

Instalación

JDK (Java Development Kit):

Contenedor de Aplicaciones: Usa el servidor Oracle WebLogic 11gR1 Puerto 7101

Entorno de Desarrollo: Hasta el momento su entorno de desarrollo más factible es JDeveloper.

La especificación o estándar: Entre los estándares q hace uso la tecnología ADF esta: EJB 3 (JSR-220), JSF (JSR-127): Trinidad, Data Binding (JSR-227), Portlets (JSR-168).

3.7 ANÁLISIS CUALITATIVO Y CUANTITATIVO.

En este punto vamos a realizar el análisis cuantitativo y cualitativo realizado con las tecnologías de aplicaciones web para Java con JSF y ADF, mediante la utilización de cuadros comparativos que permitirán determinar de manera cuantitativa y cualitativa la mejor tecnología para el desarrollo web, mismos que serán interpretados y calificados, los cuadros comparativos serán realizados tomando en cuenta los parámetros definidos con anterioridad, con sus respectivas variables que ayudaran a realizar un análisis más profundo de cada parámetro.

La calificación de cada parámetro de comparación estará realizada en base a la escala que se mostrara continuación, lo cual nos permitirá obtener resultados para determinar la tecnología más adecuada para el desarrollo de aplicaciones web.

Tabla III.XII Análisis Cualitativo y Cuantitativo

Regular	Bueno	MuyBueno	Excelente
<70%	>=70%y<80%	>=80y<95%	>=95

A cada una de las variables que intervienen se les dio un valor de 1, 2, 3 o 4 puntos dependiendo de la importancia de dicha variable. Para determinar la calificación de cada una de las variables empleadas se uso la siguiente tabla de valoraciones.

3.7.1 Escala de valoración cualitativa.

Tabla III.XIII Escala de valoración cualitativa.

1	2	3	4
No se cumple	Se cumple insatisfactoriamente	Se cumple aceptablemente	Se cumple plenamente
Inadecuado	Mas o menos	Adecuado	Muy adecuado
Insuficiente	Parcial	Suficiente	Excelente
Ninguno	Parcialmente	En su mayor parte	Totalmente

Y para poder dar una calificación general de cada una de las variables según su valor se uso la siguiente tabla de equivalencias.

Tabla III.XIV Equivalencia entre el puntaje de una variable y su valoración cualitativa

V. Cualitativa \ V. Variables	1	2	3	4
1	0.25	0.5	0.75	1
2	0.5	1	1.5	2
3	0.75	1.5	2.25	3

Para realizar la comparación se utilizara la siguiente nomenclatura:

X: representa el puntaje obtenido por la Tecnología JSF en la variable.

Y: representa el puntaje obtenido por la Tecnología ADF en la variable.

W: representa el puntaje sobre el cual se está calificando la variable.

Pjsf: representa el puntaje alcanzado de JSF en el parámetro.

Padf: representa el puntaje alcanzado de ADF en el parámetro.

Pt: Representa el puntaje sobre cual es evaluado el parámetro.

Cjsf: calificación porcentual obtenida por JSF en un parámetro.

Cadf: calificación porcentual obtenida por ADF en un parámetro.

Las formulas que se utilizaran en el proceso de análisis comparativo son las siguientes.

$$Pt = \sum w;$$

$$Pjsf = \sum x;$$

$$Padf = \sum y;$$

$$Cjsf=(Pjsf/Pt)*100; Cadf=(Padf/Pt)*100;$$

3.7.2 Evaluación del parámetro: Manejo del Patrón MVC

3.7.2.1 Variable: Modelo/Acceso a Datos.

Como comentábamos el acceso a datos es de suma importancia al momento de crear una aplicación web dinámica, por lo cual se analizara los aspectos que brinda a cada herramientas para la manipulación de base de datos y acceso a información almacenada en dichas bases.

Determinación de Indicadores

- a) ConexiónVisual adatos
- b) Soporteparamúltiples BasesdeDatos.
- c) Manipulacióndebasede datos
- d) Desempeño conlabase dedatos

Valoraciones.

a) Conexión Visual a datos

La ventaja que nos proporciona una herramienta de desarrollo es importante para realizar las conexiones visualmente a la base de datos. Valoración: (3 puntos)

b) Manipulación de base de datos

La Manipulación de la base de datos que permite una herramienta de desarrollo es fundamental para tener un control de los diferentes elementos y componentes de la misma. Valoración: (3 puntos)

c) Desempeño con la base de datos

El manejo de la base de datos como consultas, inserciones, eliminaciones, modificaciones es imprescindible para lograr resultados e información correcta. Valoración: (3 puntos)

d) Soporte para múltiples Bases de Datos

La administración de bases de datos es muy importante para la manipulación de información, dichas bases de datos pueden ser de diferentes tipos MySQL, Oracle, SQL Server entre otros. Valoración: (2 puntos)

Tabla III.XV Valoraciones de la Variable 1 acceso a datos

Indicadores	Tecnología JSF	Tecnología ADF
Conexión Visual a datos	Muy adecuado	Adecuado
Manipulación de base de datos	Excelente	Parcial
Desempeño con la base de datos	Excelente	Parcial
Soporte de múltiples DBMS	Se cumple aceptablemente	Se cumple aceptablemente

Interpretación.

La conexión visual a datos en una herramienta de entorno visual es de gran soporte para realizar tareas más rápidas en cuanto a la importación de datos a la aplicación, en cuanto a

esto JSF obtuvo un puntaje de 3 ya que cuenta con una ventana donde se puede realizar la conexión a una base de datos de forma rápida y muy completa, por otro lado ADF obtuvo 2,25 ya que la conexión visual si se la puede hacer pero no es muy amigable.

La manipulación de base de datos que proporciona cada tecnología permitirá el correcto o inadecuado manejo de datos en el desarrollo de una aplicación web. JSF obtuvo un puntaje de 3 debido a que mediante la ventana de Services/databases permiten un fácil y adecuado manipulación de datos, ADF obtuvo un puntaje de 1,5 ya que en la ventana de Database Navigator no permite arrastrar y manipular con facilidad las tablas y esto dificulta su uso.

Desempeño con la base de datos esto permite realizar las principales acciones como inserción eliminación actualización de datos. JSF obtuvo un puntaje de 3, debido a que permite manipular y realizar estas acciones directamente usando la herramienta, ADF obtuvo 1.5 ya que para manipular la base de datos debe enlazar primero a otros componentes que se vinculan en la capa modelo por tanto su utilización es más tediosa.

El soporte de múltiples Bases de Datos de esta variable depende de que tan accesible es la creación de aplicaciones Web con distintas Bases de Datos, JSF y ADF permiten la conexión con cualquier base de datos, siempre y cuando se apunte al driver respectivo, por lo cual la puntuación para esta variable es de 2, 2 respectivamente.

De manera global JSF provee mejores elementos para el acceso a datos.

Calificación.

$$\text{Pt: } w=3+3+3+2=11$$

$$\text{Pjsf= } x= 3+3+3+2=11$$

$$\text{Padf= } y=2,25+1,5+1,5+2=7,25$$

$$\text{Cjsf}=(\text{Pjsf}/\text{Pt}) * 100= 11/11 * 100= 100\%$$

$$\text{Cadf}=(\text{Padf}/\text{Pt}) * 100= .25/11 * 100=65,90\%$$

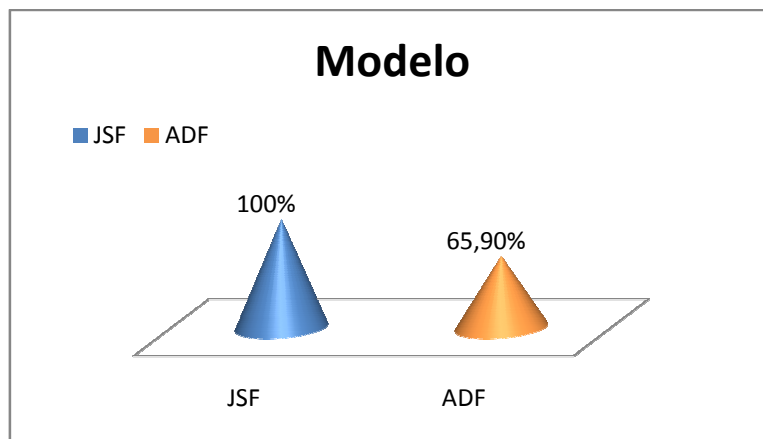


Figura III.48 Comparación Porcentual del Sub-Modulo Modelo

3.7.2.2 Variable: Vista

Una tecnología que usa una herramienta para el diseño en entorno visual se caracteriza por la facilidad de manipular componentes de forma grafica y sencilla con opciones de arrastrar y soltar. Podremosevaluar este parámetro tomando en cuenta los componentes gráficos y la forma de manejar dichoscomponentes queproporciona cadaunadelastecnologíasaevaluar.

Determinación de indicadores

- a) DiseñadorGUIVisual
- b) Facilidade aprendizaje yusodelos elementosgráficos
- c) Existenciadeherramientasdeayuda yconsulta
- d) Herramientasdenavegacióndeforma visualygrafica
- e) Fusióndecódigoydiseno.

Valoraciones.

a) Diseñador GUI Visual

Un diseñador GUI visual en una herramienta de desarrollo es fundamental para hacer más fácil el proceso de creación visual de interfaces.Valoración(3 puntos).

b) Facilidad de aprendizaje y uso de los elementos gráficos

Es sencilla y clara la enseñanza de la herramienta usando la tecnología y se puede acceder fácilmenteatodosloscomponentesgráficos.Valoración: (2puntos)

c) Existencia de herramientas de ayuda y consulta

Se tiene todos los componentes necesarios y documentación necesaria para poder ayudarse en el momento del desarrollo de la aplicación. Valoración: (2 puntos)

d) Herramientas de navegación de forma visual y grafica

Se tiene esta opción para poder navegar interactuar con los archivos y documentos que están involucrados en la aplicación. Valoración: (2 puntos)

e) Fusión de código y diseño

Se tiene la fusión de estos 2 componentes incluyendo la documentación de la declaración de los controles, la programación con esta documentación permitirá aclarar las ideas rápidamente al programador. (2 puntos)

Tabla III.XVI Tabla de valoraciones de la Variable Vista

Indicadores	JSF	ADF
Diseñador GUI Visual	Se cumple aceptablemente	Se cumple aceptablemente
Facilidad de aprendizaje y uso de los elementos gráficos	Suficiente	Excelente
Existencia de herramientas de ayuda y consulta	Totalmente	En su mayor parte
Herramientas de navegación de forma visual y grafica	Totalmente	Parcialmente
Fusión de código y diseño	Se cumple aceptablemente	Se cumple aceptablemente

Interpretación

- ✓ El parámetro de uso de componentes de interfaz nos permitió determinar cuál de las herramientas comparadas nos proporcionan mayor eficacia para el diseño de las GUI en el desarrollo Web y facilita la creación de las páginas Web. El diseñador visual GUI permitió realizar el diseño de una página Web que contenía una tarea sumamente sencilla pero fue así por la utilización de los componentes proporcionados, al evaluar este parámetro JSF con su herramienta y ADF igual, obtuvieron un puntaje de 3 por poseer excelentes componentes para esta tarea.

- ✓ La Facilidad de aprendizaje y uso de los componentes gráficos es algo sumamente importante ya que de esto dependerá el buen uso y manejo de los mismos, JSF proporciona elementos a los cuales hay que adherirles funcionalidad y simplemente arrastrarlos y soltarlos así constituyen parte de la página Web de la misma por tanto obtuvo un puntaje de 1.5, La tecnología ADF, obtuvo un puntaje de 2 debido a que utiliza componentes JSF, es decir usa nuevos componentes y los ya existentes.
- ✓ La Existencia de herramientas de ayuda y consulta facilitan al desarrollador el manejo de la tecnología y sus utilidades para esta variable JSF alcanzo la puntuación de 2 por las ayudas en línea y locales al igual que ADF obtuvo un puntaje de 2 por los elementos de ayuda que proporcionan.
- ✓ Las Herramientas de navegación de forma visual y grafica permiten la accesibilidad a todos los documento con los que se esté trabajando para una página Web en esta variable los puntajes alcanzados fueron 2, para JSF y 1 para ADF, tomando en cuenta lo observado en los módulos de prueba.
- ✓ La fusión de código y diseño es muy útil al momento del desarrollo de una aplicación para ir verificando en ambos entornos, JSF y ADF conjuntamente con la herramienta que fusionan obtuvieron 1,5 pues permiten verificar el diseño así como su código interno.

Calificación

$$\text{Pt: } w=3+2+2+2+2=11;$$

$$\text{Pjsf= } x= 3+1.5+2+2+1.5=10;$$

$$\text{Padf= } y=3+2+2+1+1.5=9,5;$$

$$\text{Cjsf}=(\text{Pjsf}/\text{Pt}) * 100= 10/11 * 100= 90.9\%$$

$$\text{Cadf}=(\text{Padf}/\text{Pt}) * 100= 9,5/11 * 100=86.36\%$$

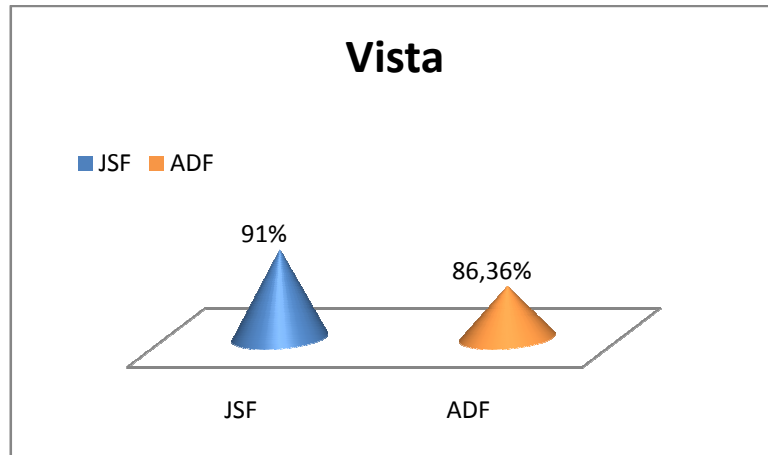


Figura III.49 Comparación Porcentual del Sub-Modulo Vista

- ✓ Basándonos en la información recopilada de las tecnologías y en el desarrollo de los módulos prácticos se pudo determinar mediante valores cuantitativos para cada una de las variables que intervienen en este parámetro, como podemos notar en el gráfico mostrado anteriormente la tecnología que brinda mayor eficiencia al momento de trabajar con los componentes de la GUI es JSF puesto que posee facilidad para trabajar con los mismos.

3.7.2.3 Variable: Controlador

El desarrollo de formularios será evaluado mediante la valoración de los componentes que brinda cada herramienta para la administración de cada uno de los elementos que intervienen en un formulario.

Determinación de Indicadores

- a) Generación automática de etiquetas de formulario.
- b) Validaciones de Formulario.
- c) Manejo visual de componentes de un formulario.
- d) Manejo de controles personalizados.

Valoraciones

- a) **Generación automática de etiquetas de formulario.**

Se tiene la creación de los formularios utilizando las respectivas opciones que posee cada herramienta. Valoración: (3 puntos)

b) Validaciones de Formulario.

Permite poder validar los datos al momento del ingreso de datos en el mismo. Valoración: (3 puntos)

c) Manejo visual de componentes de un formulario.

Se refiere a que se puede tener el acceso de observar los componentes de un formulario. Valoración: (2 puntos)

d) Manejo de controles personalizados.

Se tiene acceso al manejo de los controles de un formulario dependiendo de lo que se necesite en el momento de la construcción de la página web. Valoración(2 puntos)

Tabla III.XVII Valoración de la Variable Controlador

Indicadores	JSF	ADF
Generación automática de etiquetas de formulario	Muy adecuado	Muy Adecuado
Validaciones de Formulario	Se cumple plenamente	Se cumple aceptablemente
Manejo visual de elementos de un formulario.	Excelente	Excelente
Manejo de controles personalizados	Suficiente	Suficiente

Interpretación

- ✓ La generación automática de etiquetas de un formulario al insertar componentes permitemanipular las propiedades de un elemento de formulario ademásdefinir a donde serán enviados los datos una vez que se llene la información en el mismo, los puntajes alcanzados en la evaluación es la siguiente para JSF 3, puesto que esta herramienta inserta de forma automática las

etiquetas necesarias para hacer al formulario muy funcional, al igual que ADF, ya que contienen un documento .xml, el cual contiene estas etiquetas y su comportamiento.

- ✓ La validación de Formulario es algo fundamental para que el usuario ingrese correctamente los datos, siguiendo un formato bien definido, además de asegurar que no existan campos vacío en campos que sean obligatorios entre otras validaciones. la puntuación obtenida es para JSF 3 ,y para ADF 2.25 puesto que JSF provee componentes que permiten realizar la validación de formulario de forma sencilla y rápida, por ejemplo con el uso de la ventana Paleta/Validadores al contrario de ADF, que permite si realizar las validaciones, mas se deben adherir otros componentes para sustentarlo.
- ✓ El manejo visual de elementos de un formulario proporciona al desarrollador agilizar el proceso de desarrollo de formulario. Tanto JSF como ADF obtienen el mismo puntaje de 2, debido a que manejan fácilmente las características mediante el uso del panel de comportamientos en las propiedades.
- ✓ El manejo de controles personalizados seria un soporte para la validación de datos, para crear elementos de formularios con formatos predefinidas por el desarrollador. JSF y ADF obtuvieron un puntaje de 1,5 porque cuenta con las propiedades de los componentes para realizar estas acciones

Calificación.

Pt: $w=3+3+2+2=10$

Pjsf= $x= 3+3+2+1,5=9,5$

Padf= $y=3+2,25+2+1,5=8,75$

Cjsf= $(Pjsf/Pt)* 100= 9,5/10*100= 95\%$

Cadf= $(Padf/Pt)*100=8,75/10*100=87,5$

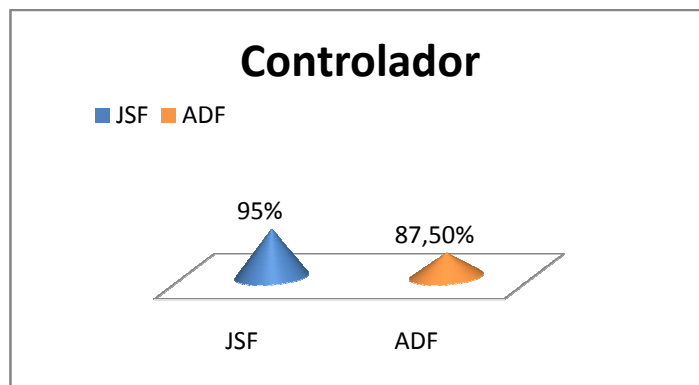


Figura III.50 Comparación Porcentual del Sub-Modulo Controlador

Analizando la información recopilada podemos notar que JSF está más centrado en la parte del Controlador, pues actúa directamente con los componentes.

3.7.3 Evaluación del parámetro Manejo de Seguridades.

El manejo de seguridades podrá ser evaluado mediante las opciones que proporciona la tecnología para generar los principales ítems de seguridades como sesiones y validación de datos, además de la facilidad que brinda la tecnología para el manejo de variables de sesiones.

3.7.3.2 Variable: A nivel de Aplicación

Determinación de Indicadores

- a) Variable Uso de sesiones.
- b) Manejo de Cookies.
- c) Encriptación de datos.
- d) Validación de datos.

Valoraciones.

a) Indicador Uso de sesiones.

El manejo de sesiones en una página Web es de gran importancia para mantener una administración del contenido de nuestra página por lo cual se ha tomado en cuenta esta variable y analizaremos la facilidad de uso y gestión de sesiones que proporciona la herramienta, valoración (3 puntos)

b) Manejo de Cookies.

El uso de cookies en una página Web es importante ya que estas proporcionan un mecanismo que sirve para almacenar datos en el navegador del usuario remoto,

para así poder identificar al usuario cuando vuelva a ingresar e identificar a este a través de diferentes páginas de un mismo sitio por lo cual nos parece importante conocer como maneja este ítem cada una de las tecnologías, valoración (2 puntos).

c) Encriptación de datos.

Un aspecto muy importante en cuanto a la seguridad de datos que atraviesa la gran red del internet es la capacidad de transportar datos cifrados, por lo cual nos parece importante analizar las opciones de manejo que proporciona cada herramienta para la encriptación de datos. (2 puntos).

d) Validación de datos.

La validación de datos es algo que siempre se debe de tomar en cuenta para mantener más segura nuestra página dando así privilegios de uso de información.

Es por esto que se analizara las opciones que nos permiten manipular la validación de datos en cada una de las tecnologías a ser evaluadas (2 puntos).

Tabla III.XVIII Tabla de valoración de la variable A nivel de aplicación

Indicadores	JSF	ADF
Uso de sesiones.	Muy adecuado	Muy adecuado
Manejo de Cookies.	Suficiente	Suficiente
Manejo de Encriptación de datos.	Parcialmente	Parcialmente
Validación de datos.	Excelente	Excelente

Interpretación.

- ✓ El uso de sesiones en una aplicación Web es de gran utilidad ya que permite controlar el acceso individual a una página o grupo de páginas determinado, mediante la asignación de un identificador único para

cada usuario. JSF provee una amplia gama de elementos que permite realizar estas tareas de manera ágil por ejemplo el uso de la ventana Servicios también cuenta con componentes gráficos que permiten insertar rápidamente variables de sesión por esto esta herramienta obtiene una puntuación de 3, ADF de igual manera facilita al usuario la creación de sesiones, por tanto estas tecnologías obtuvieron la puntuación de 3.

- ✓ Manejo de Cookies esta opción nos permite mantener un control sobre datos que se guardan en el ordenador del cliente y almacenan información referente al mismo. JSF ofrece la opción de inserción de variable cookies de manera visual al igual que ADF, por lo cual obtuvieron una puntuación de 1,5.
- ✓ Manejo Encryptación de datos es algo sumamente importante aplicar en el desarrollo de páginas Web para mantener una buena confidencialidad de la información, ninguna de las tecnologías proporcionan elementos que faciliten esta tarea por lo cual las dos obtienen un puntaje de 1.
- ✓ Validación de datos en cuanto a usuarios obtuvieron un puntaje de 2 las ambas tecnologías, debido a que cada usuario cumple un papel específico en la aplicación y pues se debe designar los roles.

Calificación.

Pt: $w=3+2+2+2=9$

Pjsf= $x= 3+1,5+1+2=7,5$

Padf= $y=3+1,5+1+2=7,5$

Cjsf= $(Pjsf/Pt)* 100= 7,5/9*100= 83\%$

Cadf= $(Padf/Pt)*100= 7,5/9*100= 83\%$

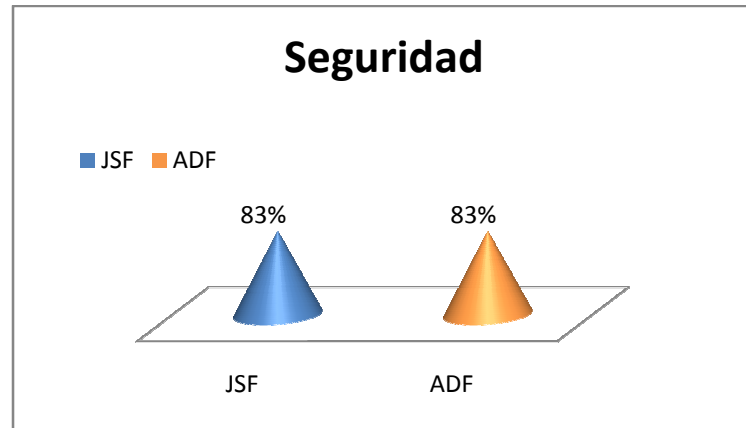


Figura III.51 Comparación porcentual del parámetro Seguridades

3.7.4 Evaluación del parámetro:Producto.

Antes de desarrollar una aplicación Web es recomendable realizar un análisis de las tecnologías o herramientas que se va a aplicar y determinar las ventajas y desventajas que se ofrecen.

3.7.4.1 Variable: Madurez del Producto

Determinación de Indicadores

- a) Especificaciones
- b) Licenciamiento
- c) Costos
- d) JDK
- e) Entorno de desarrollo.

Valoraciones.

a) Especificaciones

Establecer las versiones en la que se está desarrollando es muy importante para determinar su compatibilidad con las demás herramientas. valoración(2puntos).

b) Licenciamiento

El licenciamiento es algo que se debe tomar en cuenta ya que esto puede influir al momento de la producción de la aplicación e inconvenientes en la empresa que se va a implantar. Valoración(3puntos)

c) Costos

El costo es uno de los factores más indispensable ya que esto ayudara a determina la utilización de cierta tecnología que brinda mayor ventaja. valoración (3 puntos).

d) JDK

También se la conoce como versiones de Java. valoración(2puntos).

e) Entorno de Desarrollo

Es la herramienta que se utiliza para el desarrollo de una aplicación. valoración(3puntos).

Tabla III.XIX Valoraciones dela variable Madurez del producto

Indicadores	Tecnología JSF	Tecnología ADF
Especificaciones	Suficiente	Parcial
Licenciamiento	Suficiente	Parcial
Costos	Muy adecuado	Mas o menos
JDK	Adecuado	Adecuado
Entorno de desarrollo	Se cumple aceptablemente	Se cumple

Interpretación.

- ✓ El numero de versiones existentes es un factor que puede indicar la calidad de un producto, mientras más especificaciones existan esta demuestra que posee mejores prestaciones es por ello que la tecnología JSF obtuvo un puntaje de 1,5 y ADF con una calificación de 1.
- ✓ La utilización de ciertos tipos de licenciamiento influye de alguna manera en el entorno donde se vaya a dar la producción. Es por ello que JSF obtuvo un puntaje de 2,25 ya ADF con 1,5 ya que su licencia se incluye en la licencia de Oracle Application Server (OC4J).
- ✓ El costo influye en gran manera en el área financiero de la Empresa, de acuerdo a este análisis JSF obtuvo un puntaje de 3 y ADF con calificación de 1,5.
- ✓ Las versiones de Java se aplican tanto en las 2 tecnologías JSF y ADF y obtuvieron puntajes similares de 1,5
- ✓ La variedad de las IDE's y si es compatible con Tecnologías ayudará al

desarrollador a efectuar de manera más rápida y a elección de su gusto, dependiendo en el entorno de desarrollo que más se afine. Es por ello que JSF tiene más ventajas en cuanto a la selección de Ides y obtuvo un puntaje de 3 a diferencia de ADF que no ofrece alternativas de elegir Ides para el desarrollo y su calificación es de 1,5.

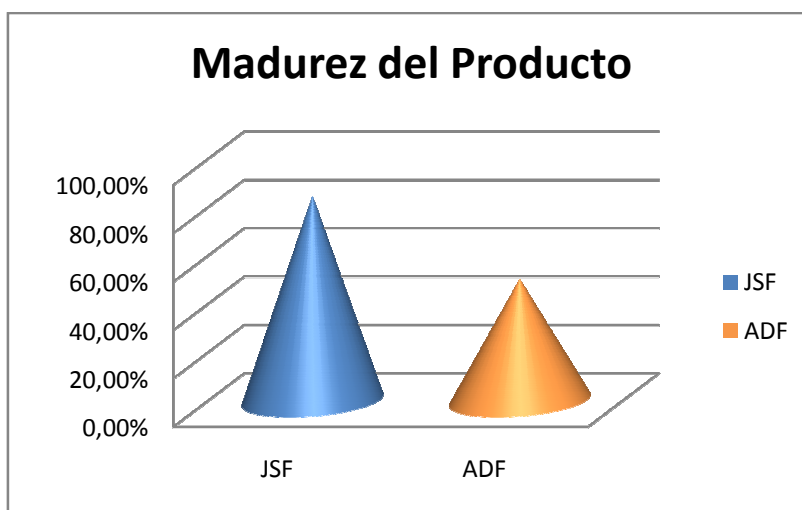
Calificación.

$$\text{Pt: } w=2+3+3+2+3=13$$

$$\text{Pjsf= } x= 1,5+2,25+3+1,5+2,25=10,5$$

$$\text{Padf= } y= 1+1,5+1,5+1,5+0,75=6,25$$

$$\text{Cjsf}=(\text{Pjsf}/\text{Pt})\cdot 100=10,5/13\cdot 100=80\% \quad \text{Cadf}=(\text{Padf}/\text{Pt})\cdot 100= 6,25/13\cdot 100= 48\%$$



Analizando la información recopilada podemos notar que JSF en cuanto a la madurez del producto es más accesible a su utilización.

3.7.4.2 Variable: Instalación

Una correcta Instalación con todos los requerimientos, evitara inconvenientes que se pueden dar a futuro en el momento del desarrollo, por tal razón es indispensable tener en cuenta todos los factores que puedan influenciar en la misma.

Determinación de Indicadores

- a) JDK

- b) Contenedor de aplicaciones
- c) Entorno de Desarrollo
- d) Versionamiento

Valoraciones.

a) JDK

También se la conoce como versiones de Java. valoración(2puntos).

b) Contenedor de aplicaciones

Servidores de Aplicaciones donde se va a ejecutar la aplicación. valoración(3puntos).

c) Entorno de Desarrollo

Es la herramienta que se utiliza para el desarrollo de una aplicación. valoración(3puntos).

d) Versionamiento:valoración(2puntos)

Tabla III.XX Tabla de valoraciones de la variable Instalación

Indicadores	Tecnología JSF	Tecnología ADF
JDK	Avance significativo	Avance significativo
Contenedor de aplicaciones	En su mayor parte	Parcialmente
Entorno de Desarrollo	Muy adecuado	Mas o menos
Versionamiento	En su mayor parte	En su mayor parte

Interpretación.

- ✓ Las versiones de Java se aplican tanto en las 2 tecnologías JSF y ADF y obtuvieron puntajes similares de 1,5
- ✓ El uso del contenedor de aplicaciones Web cumple un papel muy importante en el desempeño y funcionamiento. Y si es adaptable a varias tecnologías. JSF es compatible con varios contenedores y alcanzo un puntaje de 2,25 a diferencia de que ADF obtuvo 1,5.
- ✓ La variedad de las IDE's y si es compatible con Tecnologías ayudará al desarrollador a efectuar de manera más rápida y a elección de su gusto, dependiendo en el entorno de desarrollo que más se afine. Es por ello que JSF tiene más ventajas en cuanto a la selección de ldes y obtuvo un puntaje de 3 a diferencia de ADF que no ofrece

alternativas de elegir Ides para el desarrollo y su calificación es de 1,5.

- ✓ En cuanto al uso de los Versionamientos es la misma en las 2 tecnologías obteniendo un puntaje de 1,5.

Calificación.

Pt: $w=2+3+3+2=10$

Pjsf= $x= 1,5+2,25+3+1,5=8,25$

Padf= $y=1,5+1,5+1,5+1,5=6$

Cjsf= $(Pjsf/Pt)*100=8,25/10*100=82,5\%$ **Cadf=** $(Padf/Pt)*100= 6/10*100= 60\%$

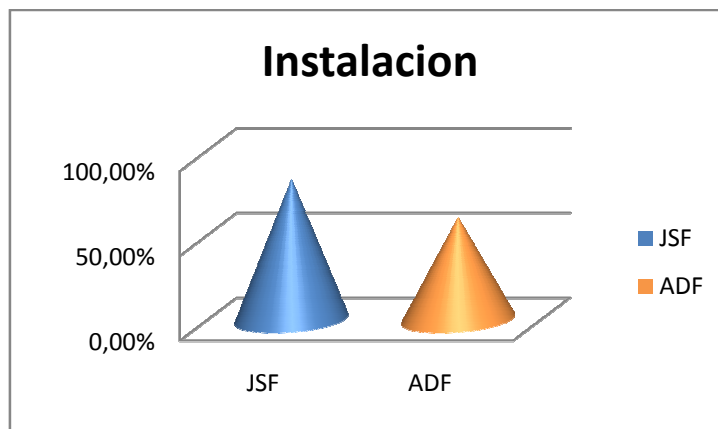


Figura III.52 Comparación Porcentual del Parámetro Instalación

Analizando la información recopilada podemos notar que JSF en cuanto a los requerimientos para su instalación es más factible.

3.8 Puntajes Alcanzados

El puntaje final y el porcentaje que ha obtenido cada herramienta de desarrollo de Entorno Visual se obtiene de la siguiente manera:

Puntaje Total del Análisis: (PA) = $\sum(Pt)$

Puntaje Total de JSF: (PTjsf) = $\sum Pjsf$

Puntaje Total de ADF: (PTadf) = $\sum Padf$

Porcentaje Total de JSF: (%Jsf) = $(PTjsf)/(PA)* 100\%$

Porcentaje Total de ADF: (%Adf) = $(PTadf)/(PA)* 100\%$

3.8.1 Tabla General de Comparación

Tabla III.XXI Tabla General de Comparación

Parámetro	Variables	Tecnología JSF	Tecnología ADF
MVC	1.1	3	2,25
	1.2	3	1,5
	1.3	3	1,5
	1.4	2	2
	2.1	3	3
	2.2	1,5	2
	2.3	2	2
	2.4	2	1
	2.5	1,5	1,5
	3.1	3	3
	3.2	3	2,25
	3.3	2	2
	3.4	1,5	1,5
	Seguridad	4.1	3
4.2		1,5	1,5
4.3		1	1
4.4		2	2
Producto	5.1	1,5	1
	5.2	2,25	1,5
	5.3	3	1,5
	6.1	1,5	1,5
	6.2	2,25	1,5
	6.3	3	1,5
	6.4	1,5	1,5
TOTALES:		53	43

3.8.2 Diagrama General De Resultados

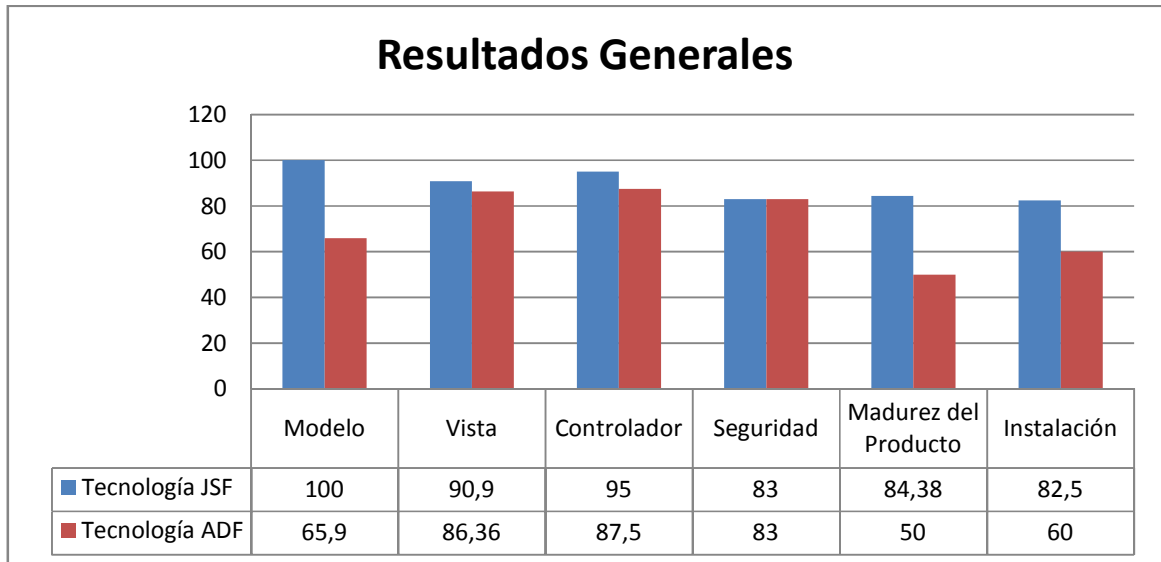


Figura III.53 Diagrama General de Resultados

P(A)=59

%Jsf = (53/59)*100=89,83% equivalente a Muy Bueno

%Adf = (43/59)*100=72,88% equivalente a Bueno

3.9 Interpretación

De acuerdo al análisis y a los puntajes obtenidos con cada una de las Tecnologías se ha obtenido como resultado que la tecnología Java Server Page ha obtenido el puntaje más alto con un porcentaje del 89,83% que es equivalente a Muy Bueno, Mientras que la Tecnología ADF a obtenido un valor de 72,88 equivalente a Bueno, por tanto se deduce que el puntaje obtenido en JSF es mayor que en ADF.

3.10 Resultado del Análisis

- ✓ En cuanto al análisis del de la variable Modelo -- acceso a datos, cada una de las tecnologías evaluadas poseen distintas formas de realizar una conexión visual a la base

de datos, sin embargo la tecnología que brinda mejores formas de realizar esta tarea es Java Server Face por su facilidad.

- ✓ La tecnología Java Server Face ofrece gran utilidad para el desarrollo de interfaces de usuario (GUI), logrando que el desarrollo de aplicaciones web se convierta en un trabajo sencillo, fácil y amigable para los desarrolladores.
- ✓ El desarrollo de páginas con JSF es una tarea muy sencilla puesto que vienen incorporados elementos que permiten manipular eficientemente todo componente de una página, entre ellos los converters, validators que permite realizar las respectivas validaciones y conversiones.
- ✓ En cuanto a la Seguridad podemos decir Ambas tecnologías proveen un buen soporte en cuanto a la administración de sesiones mediante la generación automática de código para la autenticación de usuarios con sus respectivas variables de sesión.
- ✓ Un punto débil que posee JSF ante ADF es que al momento de crear un proyecto automáticamente en su estructura ya viene representada el patrón MVC (Modelo, Vista, Controlador) y esta ayuda en gran manera para poder distinguirlas. A diferencia con la tecnología JSF el desarrollador se encarga de crear la estructura del patrón por ejemplo para el Modelo se lo realiza mediante los Source Packages con las reglas de negocio.
- ✓ Al momento de realizar el análisis comparativo de estas Tecnologías para el desarrollo de aplicaciones Web pudimos darnos cuenta que ambas tecnologías en cuanto al licenciamiento son gratuitos en el momento del desarrollo. Pero ADF al momento de realizar su producción ya tiene su costo ya que la licencia de Oracle ADF depende de la licencia de Oracle Application Server (OC4J).
- ✓ También Java Server Face trabaja de manera integrada con la herramienta de reportes como el iReport que ayuda a la realización de los reportes simplificando así el flujo de trabajo de diseño de sitios Web, lo que disminuye la tarea de la creación de una aplicación web.

3.11 Conclusiones

- ✓ JSF obtuvo un valor del 89,83% en contraste a ADF que obtuvo 72,88%.
- ✓ De acuerdo a los resultados mencionados anteriormente y según los puntajes alcanzados en cada uno de los parámetros de evaluación podemos concluir que la tecnología en entorno Java que brinda las mejores prestaciones y disminuye el tiempo en el desarrollo de aplicaciones web es Java Server Face.

CAPITULO IV

DESARROLLO DEL SISTEMA DEL CONTROL DE ACTIVOS FIJOS Y SU DEPRECIACIÓN EN EL IESS DE CHIMBORAZO

4.1 INGENIERIA DE LA INFORMACIÓN

4.1.1 Definición del ámbito

El Instituto Ecuatoriano de Seguridad Social, de Chimborazo, en el departamento financiero se maneja varios procesos manuales para realizar el ingreso y la gestión de los activos fijos de la institución, así como el cálculo de las depreciaciones de los mismos. Este proceso se lo realiza con el fin de mantener un inventario de los activos fijos y a la vez verificar su ubicación su responsable y su desgaste inclusive diario, calculado mediante la depreciación.

Esta fase financiera se la realiza actualmente, mediante las formulas que proporciona Excel, detallando cada uno de los atributos del bien que ingresa a la institución, así como la fecha de compra y así se guardan las características y se calculan las depreciaciones de los bienes.

Este proceso es muy complejo debido a que la información que se la tiene separada y dispersa, teniendo que buscar cada atributo del bien para unirlo y reportar la información.

4.1.2 Identificar Requerimientos

Para sistematizar el proceso anterior se han identificado los siguientes requerimientos, para dar solución a los inconvenientes citados anteriormente.

Req1: Ingresar al Sistema de Activos Fijos mediante Autenticación

Req2: Permitir el ingreso de las características y ubicación del Activo Fijo.

Req3: Permitir la gestión de los Activo Fijo, como: actualizar, eliminar.

Req4: Permitir solo la consulta de las fichas creadas, no acepta ningún tipo de modificación.

Req5: Permitir ingresar datos de reparación o mantenimiento de cada bien.

Req6: Generar la Depreciación de los activos mediante el método de la línea Recta.

Req7: Obtener un listado de Bienes por código o descripción.

Req8: Obtener un listado de Bienes por Departamentos.

Req9: Obtener listado por Tipos.

Req10: Obtener listado por estado.

Req11: Obtener las depreciaciones generadas de cada Bien.

4.1.3 Estudio de Factibilidad

4.1.3.1 Factibilidad Económica

El IESS de Chimborazo cuenta con una infraestructura informática necesaria para la implementación del sistema, es decir que poseen todos los equipos hardware y software por lo que la institución no tendrá que hacer inversión alguna en la adquisición de recursos.

4.1.3.2 Factibilidad Técnica

a. Recursos Humanos

El Recurso humano con el que se cuenta para el desarrollo de la aplicación es el siguiente:

Tabla IV.XXII Recursos Humanos

Recurso Humano	Cargo
Ing. Danilo Pastor	Director
Dr. Julio Santillán	Asesor
Ing. Santiago Pozo	Supervisor de sistemas del IESS.
Alejandra Guacho	Desarrolladora
Patricia Valdiviezo	Desarrolladora

b. Recursos Software

Tabla IV.XXIII Recursos Software

Recurso Software	Característica
Sistema Operativo	Microsoft Windows 7
Servidor de Base de datos	Mysql
Herramientas de desarrollo	Netbeans 6.1 y Jdeveloper
Tecnología	Java (JSF, ADF)

4.1.3.3 Factibilidad Operativa

Para el desarrollo de la aplicación se cuenta con el apoyo del supervisor de sistemas del IESS así como también de la persona encargada del manejo de los activos Fijos del departamento financiero del IESS, al igual que con todo el recurso humano mencionado anteriormente.

4.1.3.4 Factibilidad Legal

Se cuenta con el permiso de las autoridades respectivas por lo cual no existe ningún tipo de impedimento legal para el desarrollo de la aplicación.

4.1.4 Especificación de Requerimientos

Ver Anexo 1

4.2 ANALISIS DEL SISTEMA

4.2.1 Casos de Uso del Sistema

Para la obtención de los requisitos de la aplicación a elaborar, uno de los mejores métodos son los casos de uso. Los mismos que permiten representar de manera sencilla, clara, comprensiva y en general la forma de acciones y reacciones del comportamiento de un sistema.

Los objetivos de los casos de uso en nuestra aplicación son los siguientes:

- ✓ Determinar las acciones que desempeñan los actores del sistema y cómo interactúan con este.
- ✓ Aplicar las relaciones que existen entre los casos de uso en el sistema propuesto.
- ✓ Representar en un lenguaje natural y en un diagrama los casos de uso.

Actores que intervienen y sus funciones.

Tabla IV.XXIV Usuarios

Actor	Función
Contador	Ingresa, modifica y elimina la información de los activos fijos.
Usuarios	Accede a las consultas respectivas de los activos fijos, son los responsables de cada departamento de la institución.

Para la solución del problema se ha identificado los siguientes casos de uso, para el sistema de gestión de activos fijos y la depreciación de los mismos en el IESS de Chimborazo.

- ✓ Autenticación
- ✓ Gestionar Información de los Activos fijos
- ✓ Generar Depreciación de los Activos fijos
- ✓ Consultas

4.2.2 Detalle de los Casos de Uso Identificados

4.2.2.1 Funcionalidad de los Casos de Uso

Los Casos de Uso demuestran la funcionalidad de nuestra aplicación.

Tabla IV.XXV Caso de Uso Autenticación

IDENTIFICACIÓN DE CASO DE USO:	<i>Autenticación</i>	
ACTORES:	Contador (Iniciador), Sesión.	
PROPOSITO:	Acceder al sistema	
VISIÓN GENERAL:	El Contador debe estar registrado en el sistema para que pueda modificar la información.	
TIPO:	Primario esencial.	
Curso Típico De Eventos		
	Actores	Sistema
	1. El Contador Ingresa al sistema	
		2. Solicita Autenticación
	3. Ingresa su nombre y contraseña	
		4. Verifica los datos ingresados.
Cursos Alternativos		
Línea 4: Si los datos ingresados son incorrectos solicita que se ingrese nuevamente la información, caso contrario ingresa a manipular la información		

Tabla IV.XXVI Caso de Uso Gestionar Información de los Activos Fijos

IDENTIFICACIÓN DE CASO DE USO:	Gestionar Información de los Activos Fijos	
ACTORES:	Contador.	
PROPOSITO:	Realizar el ingreso de datos y la manipulación de los mismos.	
VISIÓN GENERAL:	El contador debe recopilar la información de cada bien e ingresarlo en el sistema con todos sus atributos, los mismos datos que puede modificarlos o eliminarlos.	
TIPO:	Primario esencial.	
Curso Típico De Eventos		
	Actores	Sistema
	1. El Contador se autentifica.	
		2. Presenta las opciones para gestionar la información.
	3. Decide que acción realizar.	
		4. Actualiza la BDD.
CURSOS ALTERNATIVOS		
Línea 3: Si los parámetros a modificar o ingresar son incorrectos el sistema arroja un mensaje de error.		

Tabla IV.XXVII Caso de Uso Generar Depreciación de los Activos fijos

IDENTIFICACIÓN DE CASO DE USO:	Generar Depreciación de los Activos fijos	
ACTORES:	Contador.	
PROPOSITO:	Calcular la depreciación de los activos fijos	
VISIÓN GENERAL:	El contador debe elegir la depreciación que va a mandar a calcular ya sea anual, mensual o diaria.	
TIPO:	Primario esencial.	
Curso Típico De Eventos		
	Actores	Sistema
	1. El Contador se autentifica.	
		2. Presenta las opciones para gestionar la información.
	3. Selecciona Depreciación de Activos Fijos	
		4. Presenta Opciones de depreciación.
	5. Escoge las opciones	
		6. Realiza los cálculos Internos 7. Actualiza la BDD. 8. Presenta Depreciaciones calculadas.
CURSOS ALTERNATIVOS		
Línea 6: Si algún dato no corresponde al cálculo el sistema arroja un mensaje de error.		

Tabla IV.XXVIII Caso de Uso Generar Consultas

IDENTIFICACIÓN DE CASO DE USO:	Consultas	
ACTORES:	Usuarios	
PROPOSITO:	Obtener la información recopilada de los activos fijos.	
VISIÓN GENERAL:	Los usuarios del sistema pueden acceder a obtener información de los activos fijos.	
TIPO:	Primario esencial.	
Curso Típico De Eventos		
	Actores	Sistema
	1. El Usuario se autentifica.	
		2. Presenta las opciones de consulta.
	3. Selecciona lo requerido	
		4. Presenta reporte de la consulta solicitada.
Cursos Alternativos		
Línea 1: Si el usuario no está registrado no tendrá acceso a las consultas, y el sistema emitirá un mensaje de error.		

4.2.2.2 Diagrama de los Casos de Uso

De manera general se presentan los casos de uso que intervienen en el funcionamiento del sistema.

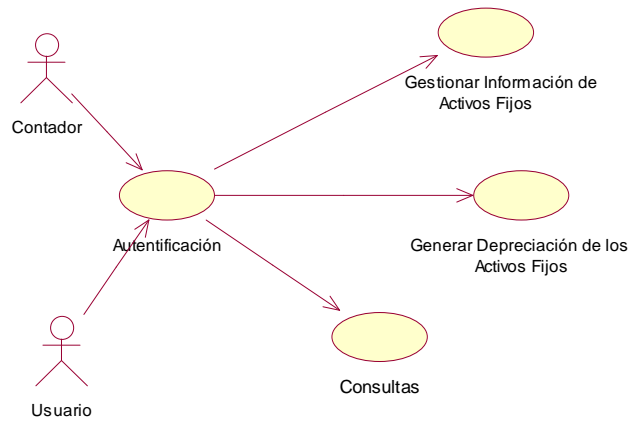


Figura IV.54 Diagrama de Casos de Uso General

4.2.3 Diagramas de Secuencia

Los diagramas de secuencias permiten modelar el comportamiento del sistema tal y como los actores lo necesitan, vamos a partir de los casos de uso. Para cada caso de uso se va a construir una serie de Diagramas de Secuencia que muestren los eventos que llegan al sistema para cada escenario del caso de uso.

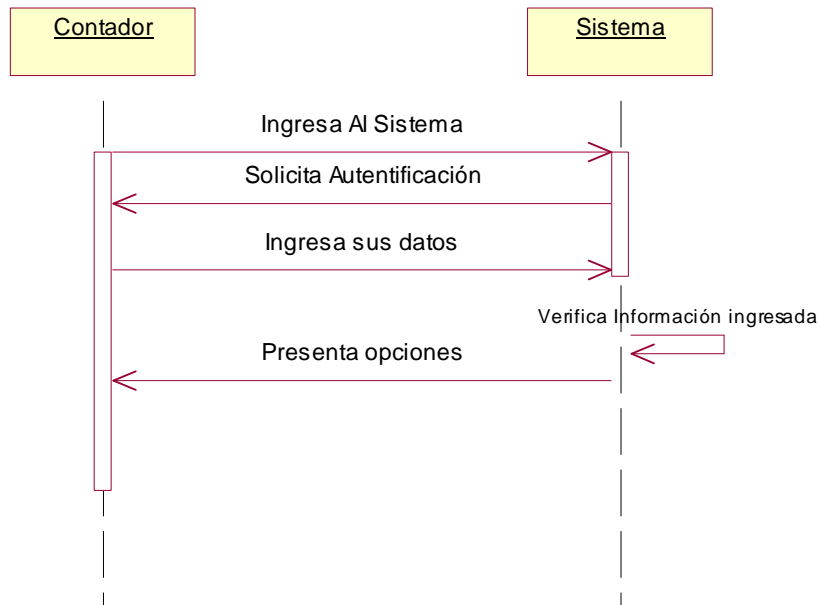


Figura IV.55 Diagrama de Secuencia de Autenticación

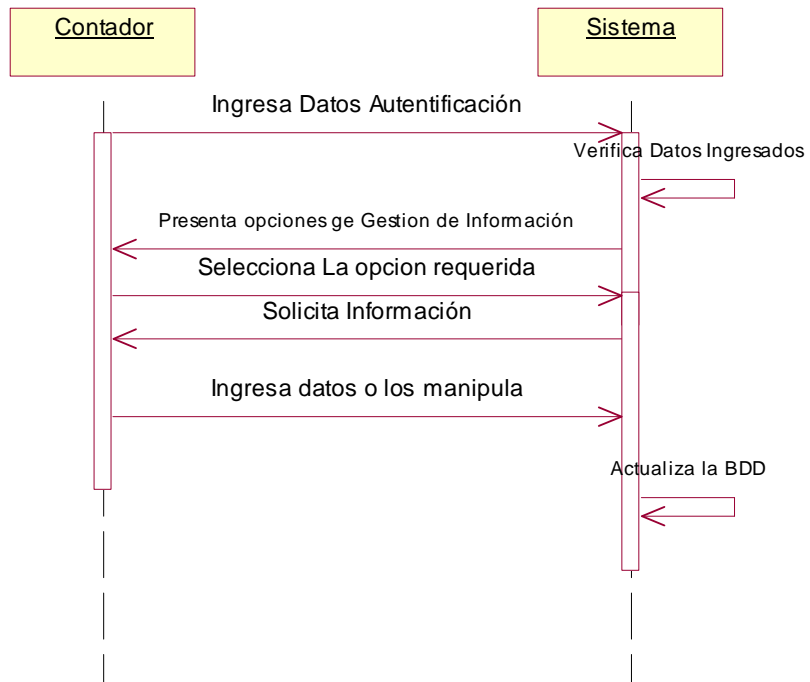


Figura IV.56 Diagrama de Secuencia de Gestión de Información de Activos Fijos

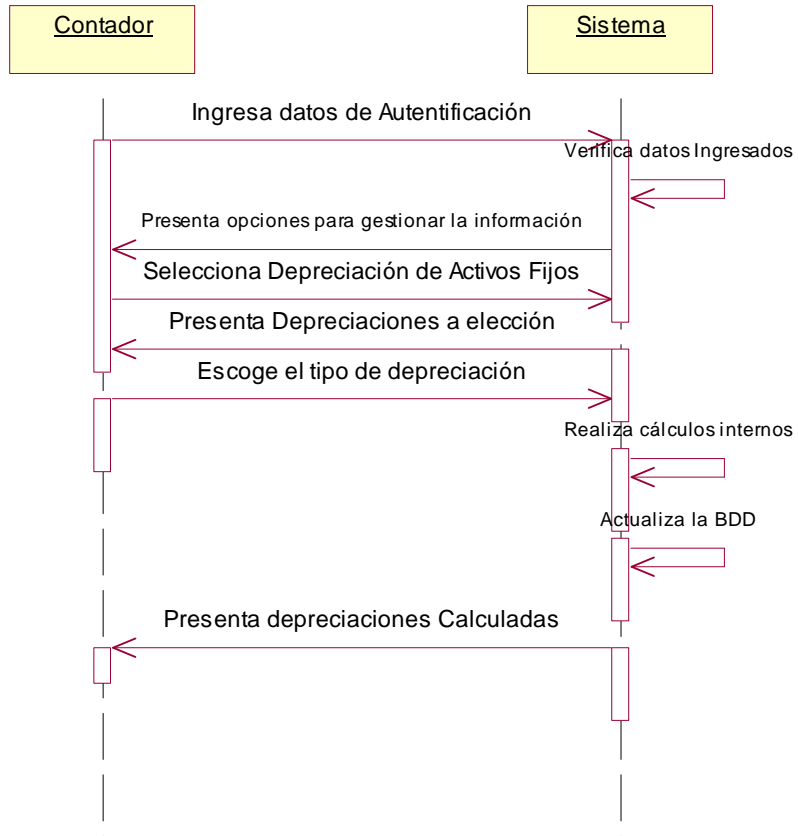


Figura IV.57 Diagrama de Secuencia de Generar depreciación de Activos Fijos

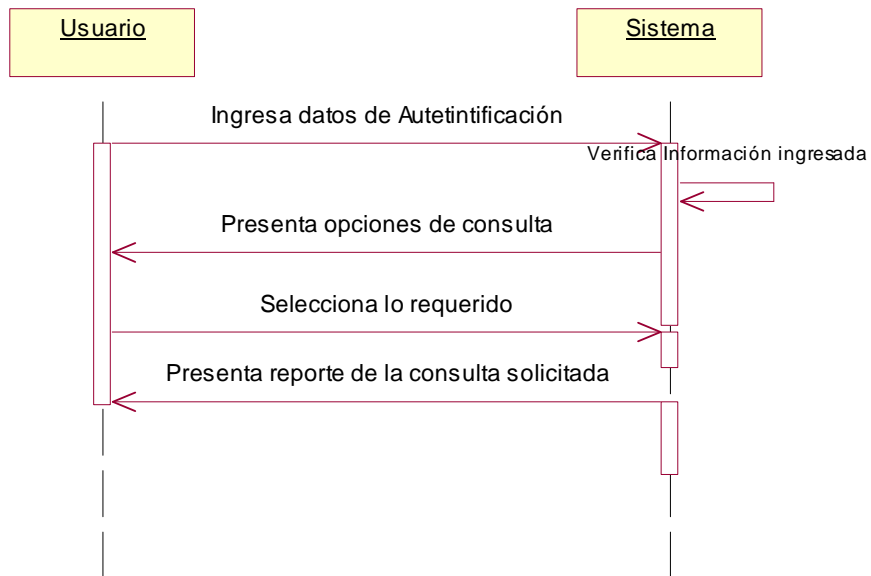


Figura IV.58 Diagrama de Secuencia de Consultas

4.2.4 Diagramas de Colaboración

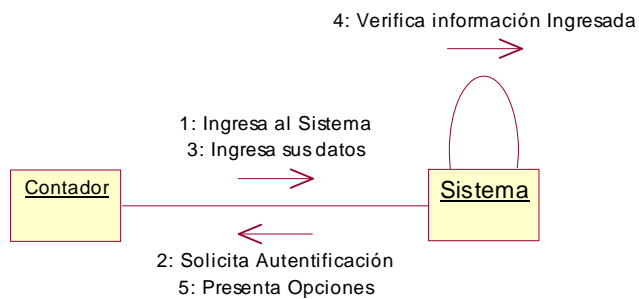


Figura IV.59 Diagrama de Colaboración de Autenticación

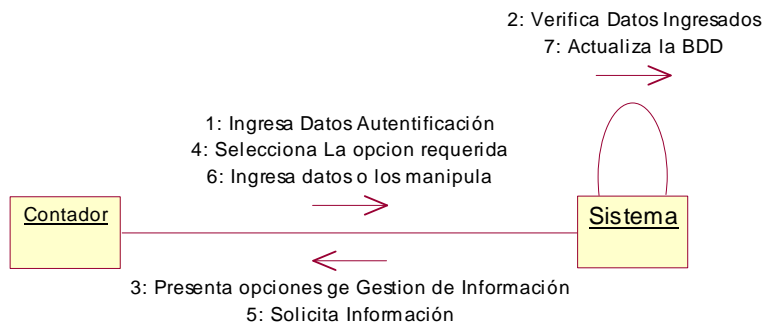


Figura IV.60 Diagrama de Colaboración de Gestión de Información de Activos Fijos

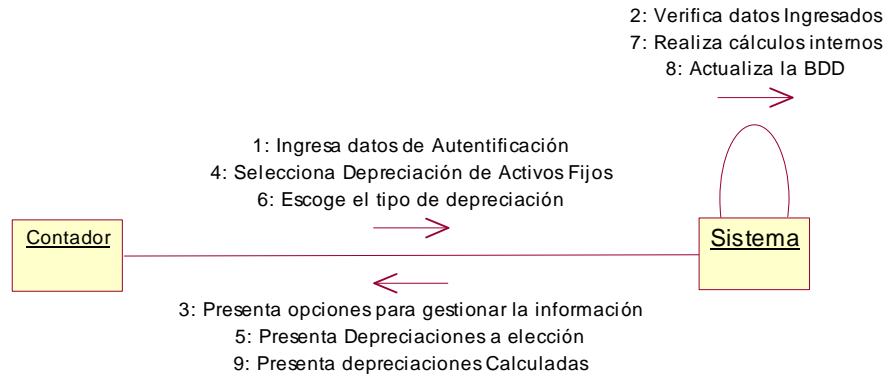


Figura IV.61 Diagrama de Colaboración de Generar depreciación de Activos Fijos

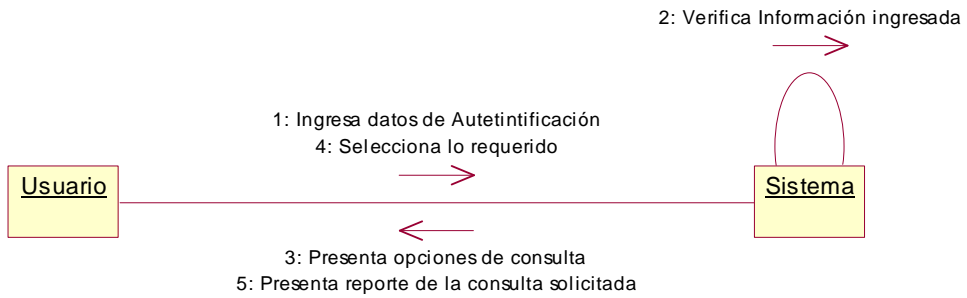


Figura IV.62 Diagrama de Colaboración de Consultas

4.3 DISEÑO

4.3.1 Casos de Uso Reales

Tabla IV.XXIX Caso de Uso Autenticación

Actores	Sistema
1. El Contador Ingresa al sistema	
	2. Solicita Autenticación
3. Ingresa su nombre y contraseña	
	4. Verifica los datos ingresados.

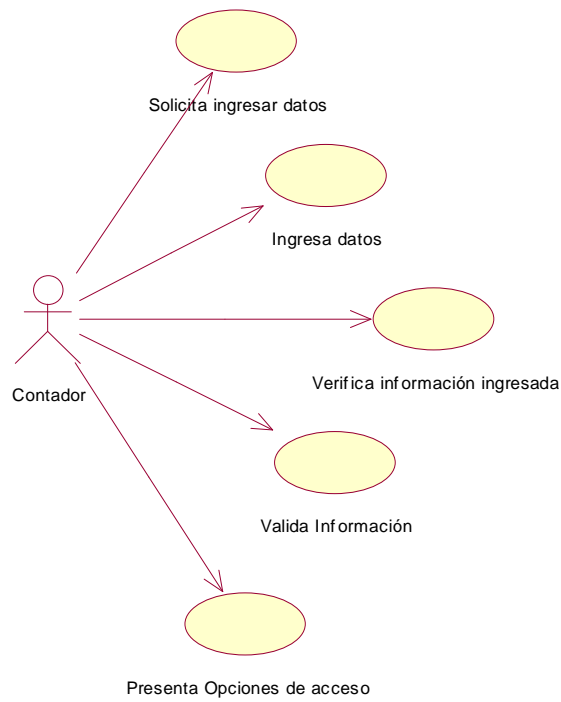


Figura IV.63 Diagrama de caso de Uso de Autenticación

Tabla IV.XXX Caso de Uso Gestionar Información de los Activos Fijos

Actores	Sistema
1. El Contador se autentifica.	
	2. Presenta las opciones para gestionar la información.
3. Decide que acción realizar.	
	4. Actualiza la BDD.

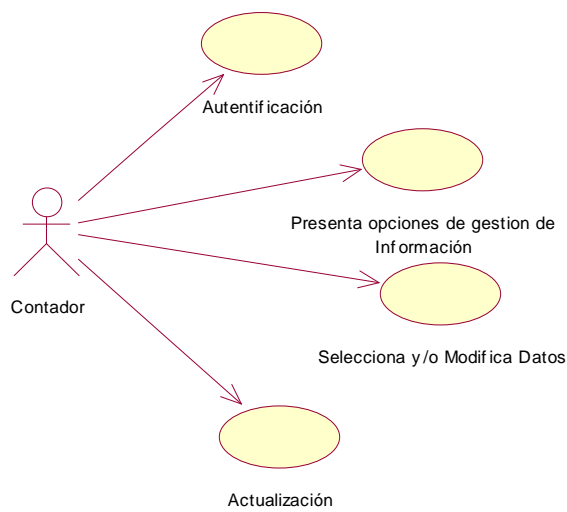


Figura IV.64 Diagrama de Caso de Uso de Gestionar Información de los Activos Fijos

Tabla IV.XXXI Caso de Uso Generar Depreciación de los Activos fijos

Actores	Sistema
1. El Contador se autentifica.	
	2. Presenta las opciones para gestionar la información.
3. Selecciona Depreciación de Activos Fijos	
	4. Presenta Opciones de depreciación.
5. Escoge las opciones	
	6. Realiza los cálculos Internos 7. Actualiza la BDD. 8. Presenta Depreciaciones calculadas.

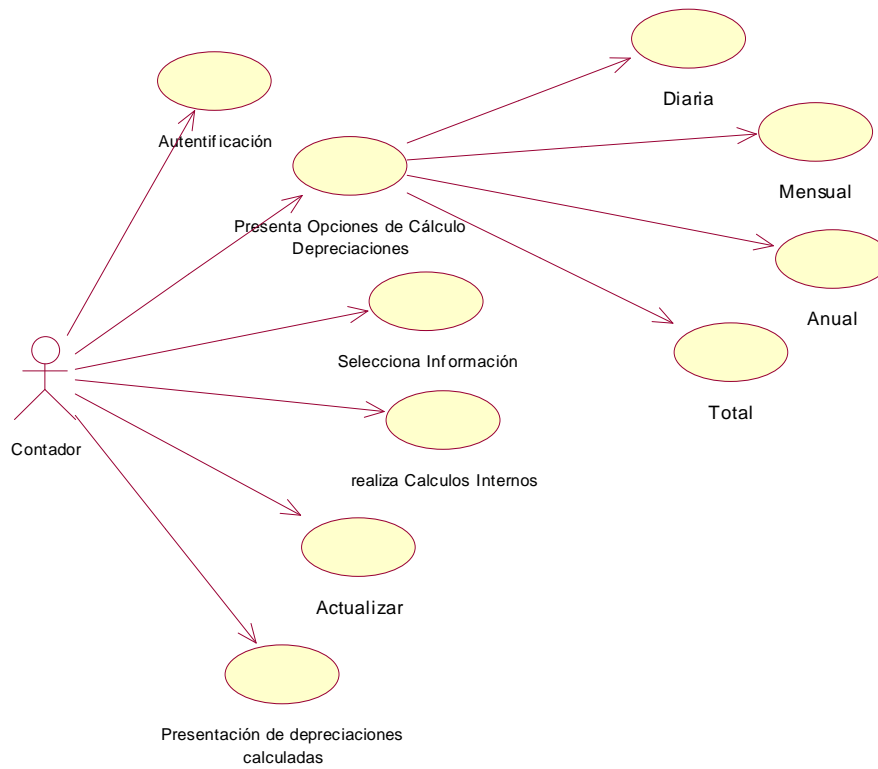


Figura IV.65 Diagrama de Caso de Uso de Generar Depreciación de los Activos fijos

Tabla IV.XXXII Caso de Uso Generar Consultas

Actores	Sistema
1. El Usuario se autentifica.	
	2. Presenta las opciones de consulta.
3. Selecciona lo requerido	
	4. Presenta reporte de la consulta solicitada.

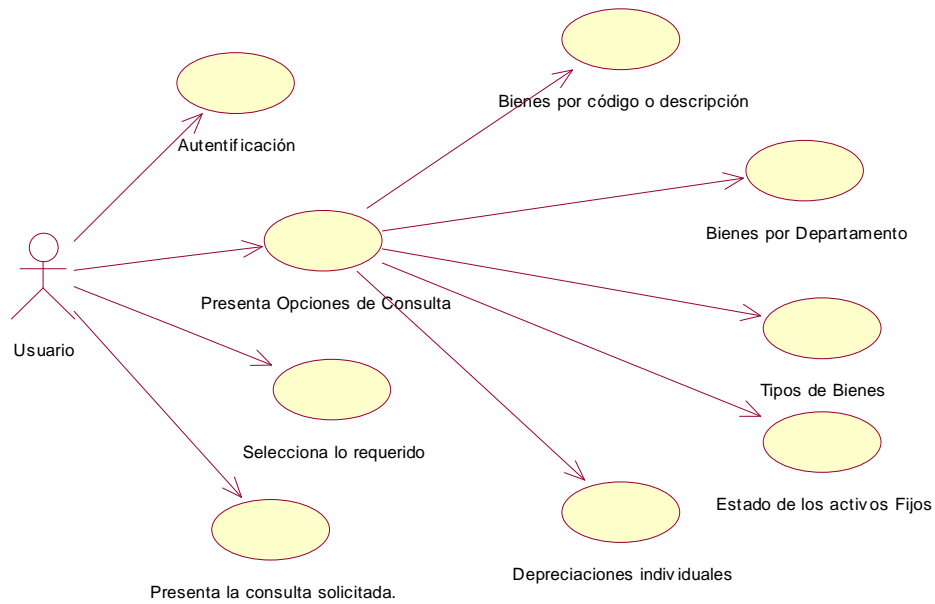


Figura IV.66 Diagrama de Caso de Uso de Generar Consultas

4.3.2 Definición de Informes e Interfaces de Usuario

4.3.2.1 Definición de la información de la Interfaz de Usuario

Para la interfaz de usuario hemos partido del flujo de los casos de usos reales, que actúan como prototipos para definir la interfaz del usuario tanto para las pantallas como para los documentos.

Es por tanto que la interacción entre el usuario y el sistema es mediante una página Web, por lo cual necesitaremos un Browser o navegador.

4.3.2.2 Lenguaje de Comunicación

a. Comunicación con el Usuario.

La comunicación aplicación/usuario es de fácil entendimiento para el usuario debido a que se maneja como una aplicación Web.

b. Nivel Arquitectónico

El entorno propuesto posee seguridad a nivel de arquitectura de aplicación como podemos observar en la figura a continuación.

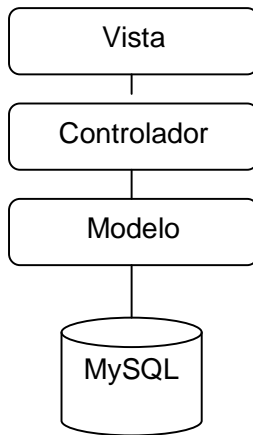


Figura IV.67 Nivel Arquitectónico

4.3.3 Diagramas de Interacción

4.3.3.1 Diagramas de Secuencia

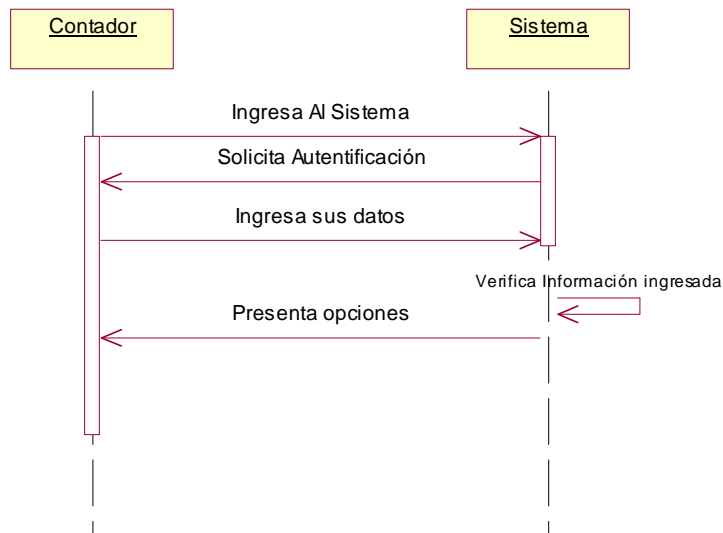


Figura IV.68 Diagrama de Secuencia de Autenticación

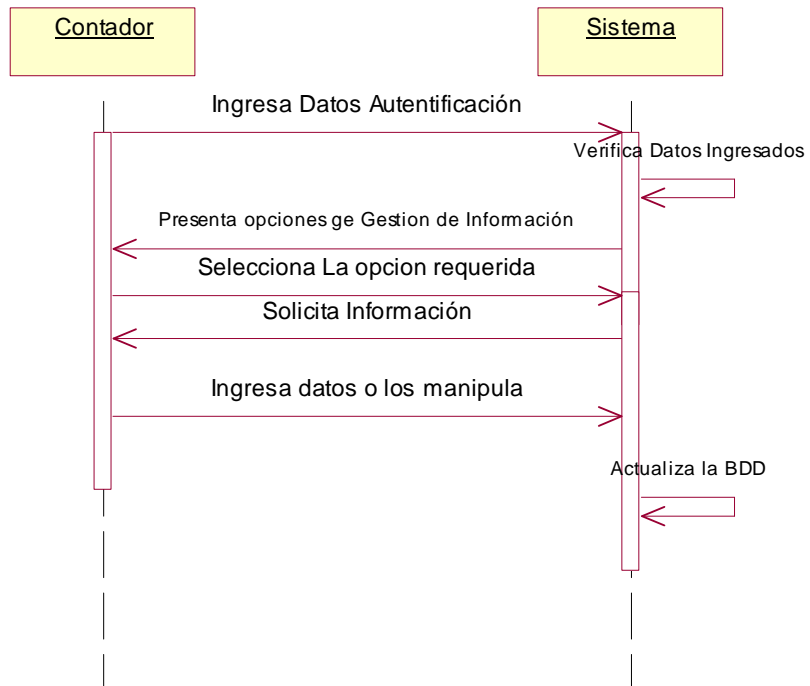


Figura IV.69 Diagrama de Secuencia de Gestión de Información de Activos Fijos

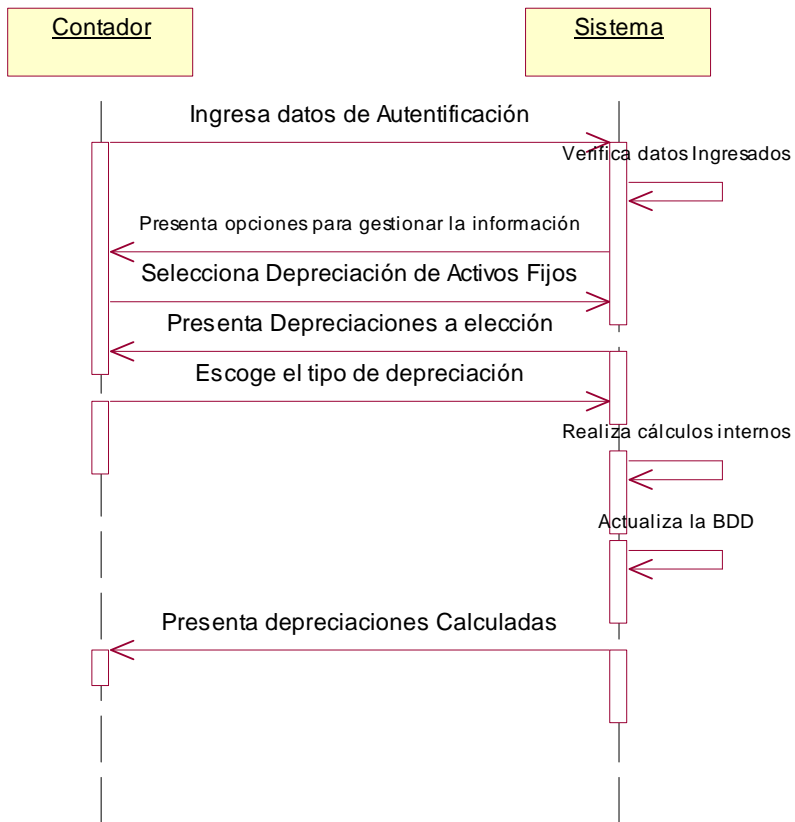


Figura IV.70 Diagrama de Secuencia de Generar depreciación de Activos Fijos

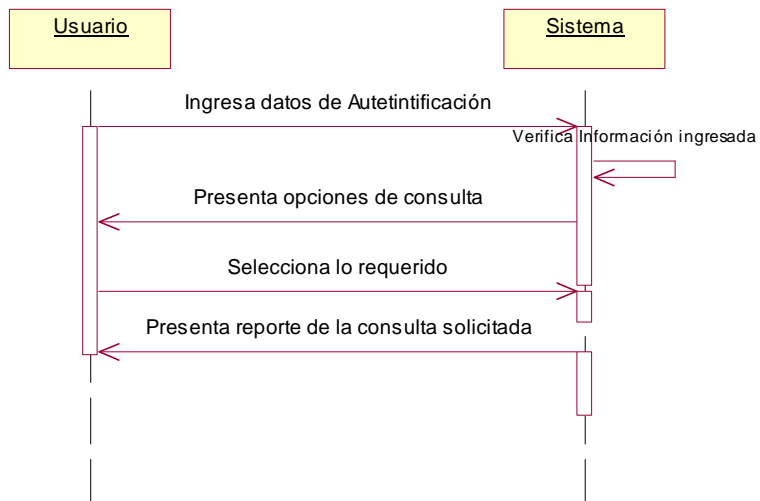


Figura IV.71 Diagrama de Secuencia de Consultas

4.3.3.1 Diagramas de Colaboración

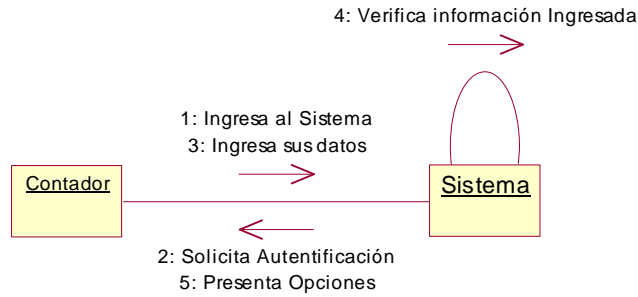


Figura IV.72 Diagrama de Colaboración de Autenticación

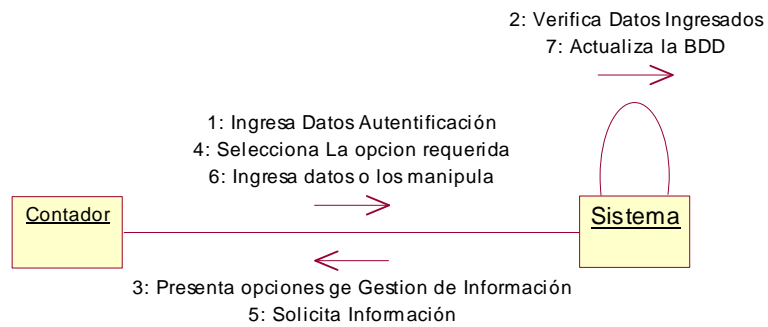


Figura IV.73 Diagrama de Colaboración de Gestión de Información de Activos Fijos

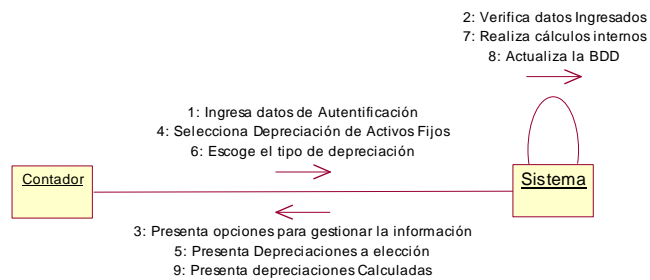


Figura IV.74 Diagrama de Colaboración de Generar depreciación de Activos Fijos

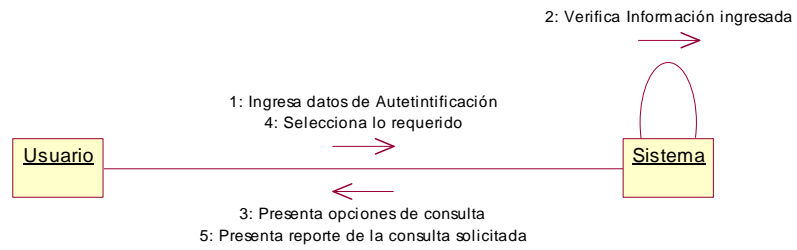


Figura IV.75 Diagrama de Colaboración de Consultas

4.3.3.3 Diagrama de Calles

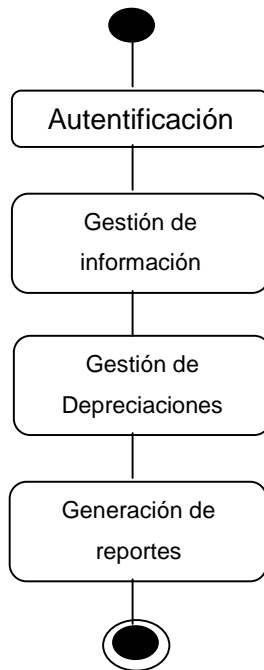


Figura IV.76 Diagrama de Calles

4.3.4 Diagrama de Clases

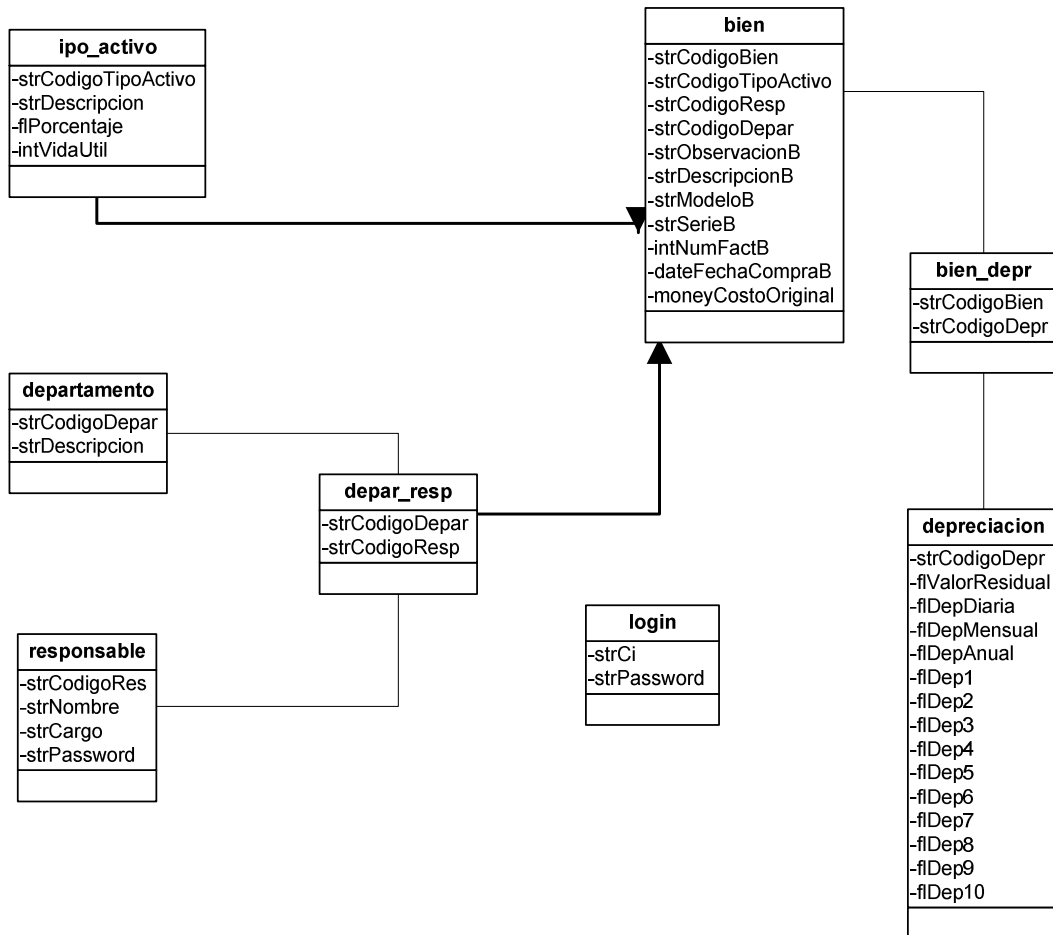


Figura IV.77 Diagrama de Clases

4.3.5 Diagrama de Base de Datos

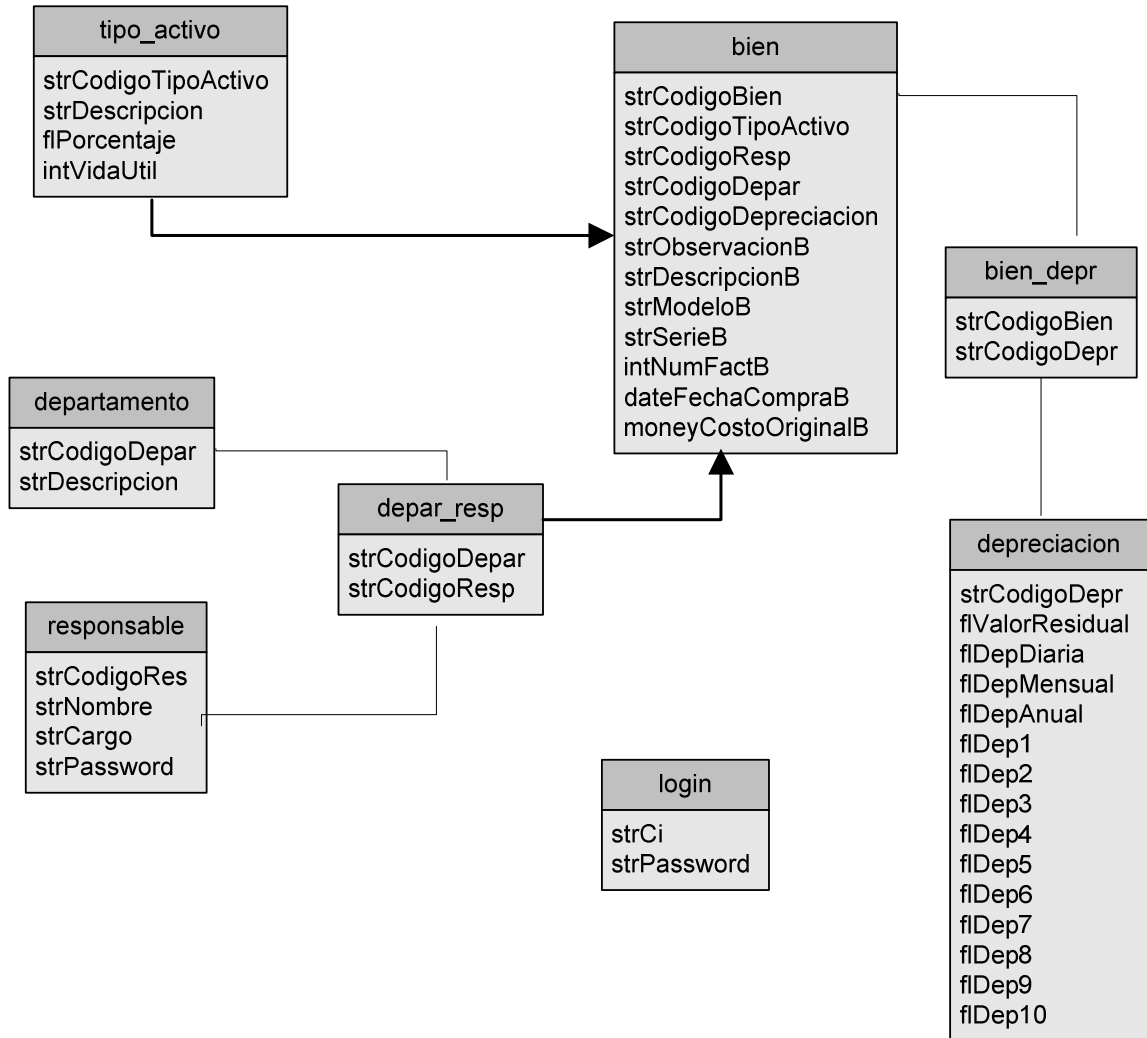


Figura IV.78

Diagrama de Base de Datos

4.3.6 Diagrama de Despliegue

4.3.6.1 Diagrama de componentes

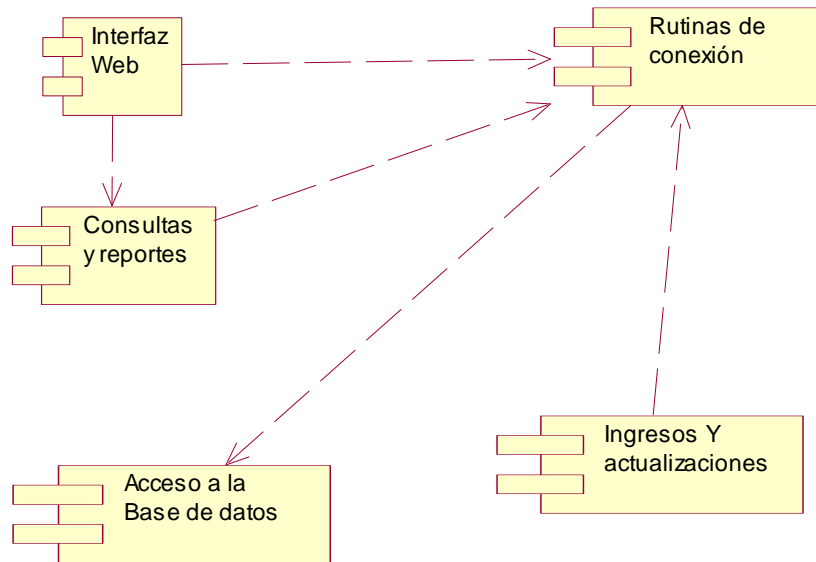


Figura IV.79 Diagrama de Componentes

4.3.6.2 Diagrama de Nodos

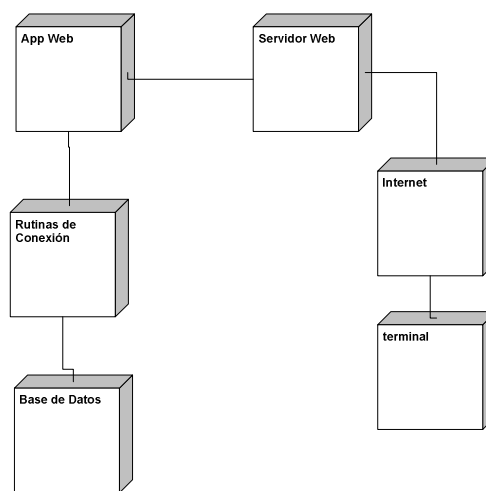


Figura IV.80 Diagrama de Nodos

4.4 Implementación y Pruebas

4.4.1 Definición de estándares de Programación

Para realizar la codificación del sistema se han definido los siguientes estándares:

- ✓ El Nombre de las tablas comienza con letra minúscula, seguido de () en caso de que tenga más de una palabra.
- ✓ El Nombre de los campos empieza con una letra minúscula, seguido de una letra mayúscula en el caso de más de una palabra.
- ✓ El Nombre de los atributos de las clases comienza con letra minúscula, seguido de una letra mayúscula en el caso de más de una palabra.
- ✓ Los métodos que interactúan con la base de datos reciben como parámetros de entrada:
 - Connection: representa la cadena de conexión hacia la base de datos.
 - Lista de parámetros en el caso de ser necesaria.
- ✓ Los métodos que interactúan con la base de datos en el caso de funciones de inserción, deben devolver como resultado el código del campo que ha sido insertado.

4.4.2 Pruebas Unitarias

Para asegurar el correcto funcionamiento del sistema se han probado las clases y sus métodos de forma independiente, enviando datos de entrada desde el código para luego obtenerlos a través de los diferentes métodos para realizar consultas.

Se han probado especialmente todas las funciones para validaciones de datos.

4.4.3 Prueba de módulos y del Sistema

Las pruebas finales consistieron en verificar que la información ingresada se vea inmediatamente reflejada en las consultas del sistema, esto sirve para comprobar que la información se está registrando correctamente en la base de datos.

Se provocaron errores intencionales para verificar el correcto funcionamiento del sistema, así como de las funciones de validación de datos, como por ejemplo:

- ✓ Realizar consultas a tablas sin valores
- ✓ Ingresar campos vacíos
- ✓ Buscar información que no existe.

CAPITULO V

ANALISIS DE RESULTADOS

Con el fin de realizar la validación de la hipótesis: “El análisis comparativo entre las tecnologías ADF y JSF, para el desarrollo Web permitirá desarrollar el sistema Web del departamento de contabilidad del IESS con mejor rapidez y seguridad en la aplicación. Se determina que la estadística inferencial y específicamente, Chi cuadrado es el método a aplicar cuando se desea realizar la demostración de una hipótesis, teniendo como limitante la falta de un sistema anterior con el cual se pueda comparar, para la correcta ejecución de este método se debe realizar un cuestionario con preguntas relacionadas a la hipótesis, posteriormente se realizará la evaluación respectiva con el fin de afirmar o negar la validez de la hipótesis.

5.1 Estadística Inferencial

La Estadística Inferencial es una parte de la Estadística que comprende los métodos y procedimientos para deducir propiedades de una población a partir de una pequeña parte de la misma. La bondad de estas deducciones se mide en términos probabilísticas, es decir, toda inferencia se acompaña de su probabilidad de acierto, a continuación se desarrolla el estudio estadístico, tomando en cuenta las variables dependientes e independiente.

TablaV.XXXIII Variables a ser analizadas

Variable Independiente	El análisis comparativo de las tecnologías ADF y JSF, para el desarrollo Web.
Variabes dependientes	Desarrollar el sistema Web del departamento de contabilidad del IESS con mejor <i>rapidez y seguridad</i> en la aplicación

5.1.1 Selección Y Determinación De La Muestra

Con la finalidad de recolectar información, se elaboro una encuesta y se realizara a 20 personas involucradas en el desarrollo web con java; las personas que respondan esta encuesta deberán tener una mediana experiencia en la utilización de las herramientas visuales para el desarrollo web; un aspecto muy importante es la rapidez y claridad de la encuesta, esto evita malestar y confusión en el encuestado.

La encuesta realizada se encuentra en el anexo 2

5.1.2 Obtención De Los Datos

La encuesta fue realizada a 20 personas relacionadas con el desarrollo de web con java, mismas que han desarrollado sus aplicaciones con herramientas de entorno visual y sin su uso, constó de 4 preguntas relacionadas al uso de las tecnologías de entorno visual, las mismas que son fundamentales en la demostración de la hipótesis:

- ✓ ¿Usando la tecnología JSF para el desarrollo de aplicaciones Web se obtiene rapidez en el mismo?
- ✓ ¿Usando la tecnología JSF para el desarrollo de aplicaciones Web no se obtiene rapidez en el mismo?
- ✓ ¿JSF no proporciona mayor seguridad para las aplicaciones Web?
- ✓ ¿JSF proporciona mayor seguridad para las aplicaciones Web?

La calificación que se ha dado a las respuestas se encuentra de la escala de 1 a 5, sin embargo, el encuestado no está en la obligación de calificar todos los enunciados, debido a que se debe respetar el criterio de cada persona y no exigir una respuesta, de este modo tenemos una respuesta más cercana a la realidad.

5.1.3 Clasificación Y Organización De Los Datos

Las respuestas que se han obtenido son las siguientes:

TablaV.XXXIV Respuestasdelosencuestados

ENCUESTADOS	ENUNCIADOS			
	1º	2º	3º	4º
1	5	0	0	2
2	5	1	0	3
3	5	1	0	0
4	4	1	0	3
5	3	4	0	3
6	1	1	0	4
7	2	2	0	5
8	4	0	0	3
9	5	0	0	0
10	4	1	0	2
11	5	0	0	3
12	3	0	0	3
13	5	0	0	3
14	4	0	0	4
15	5	0	0	3
16	4	1	0	4
17	5	1	0	0
18	4	1	0	0
19	5	0	0	0
20	4	0	0	0
TOTAL	82	14	0	45

5.1.4 Análisis De Los Datos

Para realizar el cálculo del Chi-cuadrado es preciso construir una tabla de contingencia, o tabla de valores observados, esta es una estructura de filas y columnas que sirven para mostrar el resultado de clasificar el total de casos (datos).

Tabla V. XXXV Valores observados

	Valores Obtenidos en las preguntas Realizadas		Total
Variable Dependiente			
Rapidez	82	14	96
Variable Dependiente			
Seguridad	0	45	45
Total	82	59	141

El siguiente paso es determinar los valores esperados, a partir de los datos observados, y para esto debemos multiplicar los respectivos marginales y dividir por el gran total.

$$V.E.= (\sum \text{fila} * \sum \text{columna})/\text{Total}$$

$$V.E._{11} = (96 * 82) / 141 = 55,83$$

$$V.E._{12} = (96 * 59) / 141 = 40,17$$

$$V.E._{21} = (45 * 82) / 141 = 26,17$$

$$V.E._{22} = (45 * 59) / 141 = 18,83$$

Tabla V. XXXVI Valores Esperados

	Valores Esperados en base a los Observados		Total
Variable Dependiente Rapidez	55,83	40,17	96
Variable Dependiente Seguridad	26,17	18,83	45
Total	82	59	137

A través de Chi-cuadrado se probará de forma afirmativa o negativa que la distribución de las frecuencias observadas difiere significativamente en relación a la distribución de las frecuencias que deberíamos esperar.

$$X^2 = \frac{\sum(O - E)^2}{E}$$

O= Frecuencia o valores observados, E= Frecuencia o valores Esperados

TablaV.XXXVII Chi-cuadrado

O	E	O-E	(O-E) ²	(O-E) ² /E
82	55,83	26,17	684,87	12,27
14	40,17	-26,17	684,87	17,04
0	26,17	-26,17	684,87	26,17
45	18,83	26,17	684,87	36,37
				X2=91.85

Para afirmar o negar la hipótesis debemos comparar el valor obtenido (91.85) con el Chi-cuadrado crítico de la tabla de valores críticos, los parámetros que debemos tomar en cuenta son los grados de libertad y el nivel de significación; el primero se define como el (número de columnas-1) x (número de filas -1), en este caso es (2-1) x (2-1) = 1; el nivel de significación también conocido como nivel de confianza se refiere a la probabilidad de que los resultados observados se deban

al azar, este valor es fijado por el investigador usualmente es de 5% o 10 % (es decir 0.005 o 0.01)
)Considerando $P=0.05$ y $GL=1$, se tiene que X^2 crítico es igual a 3.84. Se observa que X^2 calculado (91.85) es mayor a X^2 crítico (3.84).

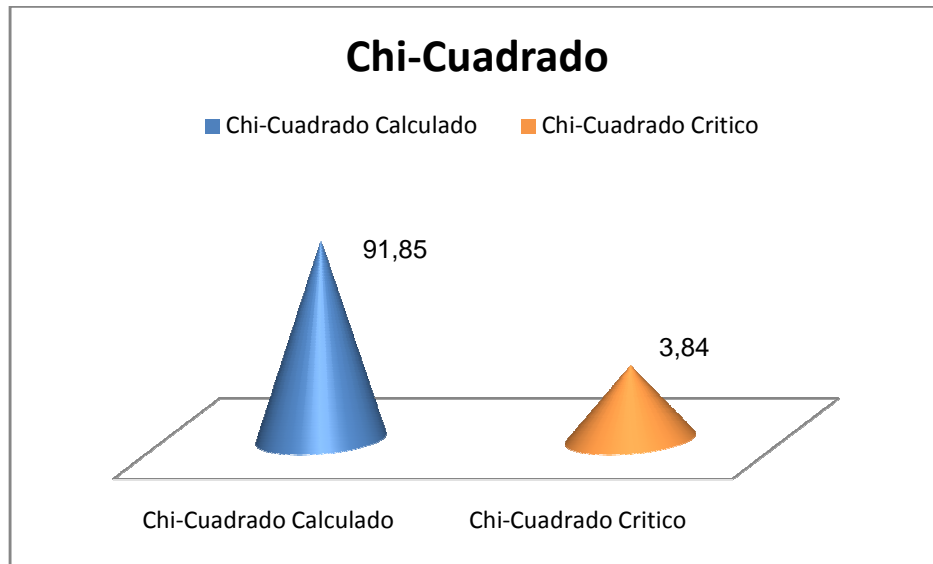


Figura V. 81 Gráfico que representa los valores del Chi-Cuadrado

Ahora podemos afirmar que el desarrollo web usando la tecnología JSF proporciona rapidez y seguridad en el desarrollo, debido a que el valor calculado de chi-cuadrado crítico es menor al calculado observado y esperado se puede afirmar la hipótesis.

5.1.5 Representación Gráfica De Datos

El siguiente gráfico indica los puntajes obtenidos en la encuesta, el primer enunciado es el que tiene un valor mayor (82 puntos), luego se puede ver que el cuarto enunciado presenta también un número considerable de puntos (45), muy por debajo de estos valores se encuentra el segundo y tercer enunciado (14 y 0 respectivamente).

Preguntas

- ✓ ¿Usando la tecnología JSF para el desarrollo de aplicaciones Web se obtiene rapidez en el mismo?

- ✓ ¿Usando la tecnología JSF para el desarrollo de aplicaciones Web no se obtiene rapidez en el mismo?
- ✓ ¿JSF no proporciona mayor seguridad para las aplicaciones Web?
- ✓ ¿JSF proporciona mayor seguridad para las aplicaciones Web?

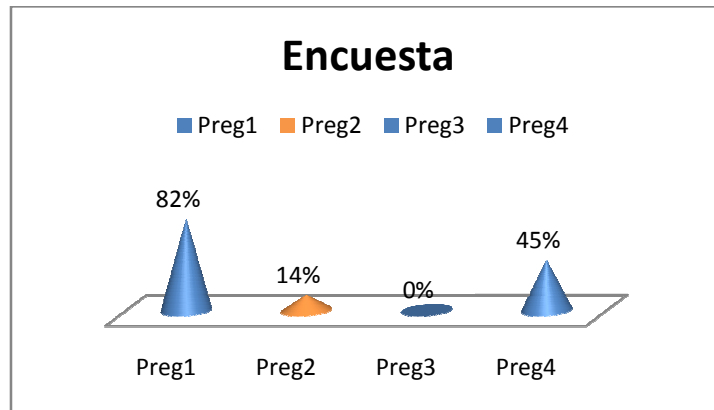


Figura V.82 Puntajes obtenidos de los enunciados

Debido a que cada encuestado podía seleccionar el enunciado que creía conveniente para votar, es importante conocer el número de personas que votaron por cada enunciado, y lo podemos ver en la siguiente tabla:

Tabla V. XXXVIII Número de votantes por enunciado

ENUNCIADOS

	1	2	3	4
VOTANTES	20	12	5	1

El primer enunciado capta el número máximo de votantes, por el contrario, el tercer enunciado es el menos votado, de esta forma podemos observar que el tercer enunciado no solo es el que tiene menor puntaje, sino menor número de votantes.

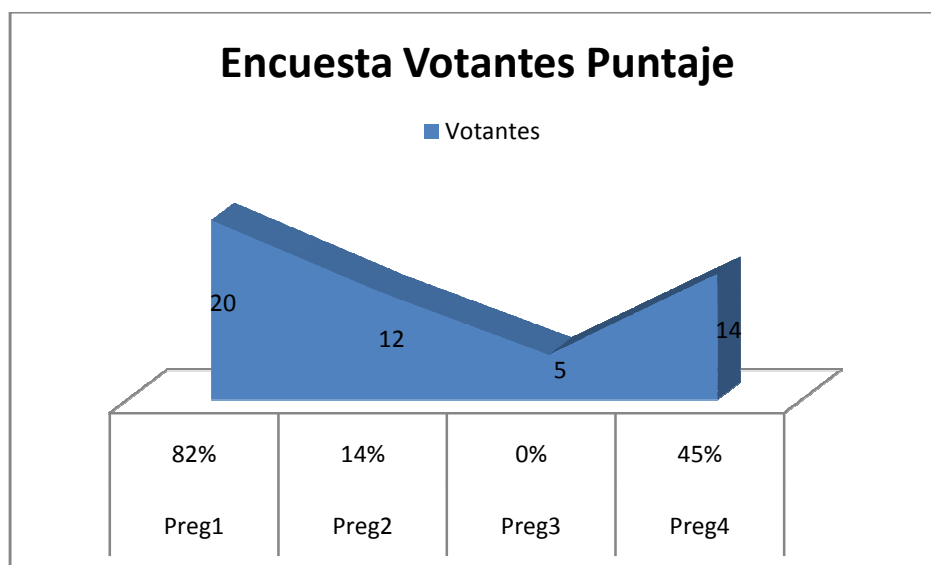


Figura V.83 Relación entre número de votantes y puntaje

Una apreciación importante es que, la curva formada por los puntajes obtenidos, es semejante a la curva del número de votantes.

5.1.6 Validación De La Hipótesis

La implementación es en sí el núcleo del desarrollo de un sistema, así como, la ingeniería de software realiza investigaciones para poder llegar a presentar la solución deseada, existen grupos que se encuentran trabajando con la finalidad de disminuir el trabajo que realiza el programador, disminuyendo de esta manera, el tiempo empleado en realizar una tarea obteniendo rapidez, sin descuidar características como la calidad, la seguridad, la eficiencia, etc.

Si podemos contar con Tecnologías Java que nos permiten agilizar nuestro trabajo, estaremos en la capacidad de presentar una solución más productiva, de esta forma se demuestra que el *“Análisis Comparativo De Tecnologías De Aplicaciones Web En El Entorno Jsfp permitirá obtener rapidez y seguridad en las aplicaciones Web”* principalmente por permitir identificar la Tecnología más productiva respecto a las necesidades que deseamos cubrir.

5.1.7 CONCLUSIONES

- ✓ El resultado de Chi-cuadrado calculado 91.85 es mucho mayor que Chi-cuadrado crítico 3.84, por lo que podemos confirmar la validez de la hipótesis.
- ✓ La rapidez y seguridad en el desarrollo de páginas Web usando la tecnología JSF queda confirmado a más de la utilización del método del chi-cuadrado por las preguntas y la obtención de los resultados, ya que en la pregunta N°1 que se refiere a la rapidez hay una sumatoria de 82 que difiere considerablemente de su negación que es 14, así también la negación de la pregunta referente a la seguridad nos hace notar con su valor cero que proporciona seguridad en la creación de páginas Web, usando JSF con un valor de 45.
- ✓ La distribución de una encuesta debe estar orientada al público que tenga conocimiento y dominio del tema, de esta manera aseguramos que la muestra sea más real.

CONCLUSIONES

1. El análisis comparativo de las Tecnologías JSF y ADF permitió entender claramente que la tecnología JSF simplifica la construcción de interfaces en aplicaciones Java. Los desarrolladores pueden rápidamente construir aplicaciones Web incorporando componentes de interfaz reutilizables en una página y conectando éstos componentes a los datos orígenes. Estos componentes soportan eventos generados por el cliente los cuales pueden ser gestionados desde el lado del servidor.
2. Los valores obtenidos en el análisis comparativo nos permite deducir que JSF es mejor que ADF en los puntos analizados y lo refleja en un 89,83%, a diferencia de ADF que obtuvo un valor de 72,88%.
3. Las variables, indicadores e índices utilizados para el análisis comparativo nos han servido de manera óptima, para obtener el resultado en el análisis.
4. Mediante el estudio de las tecnologías con los IDE's se trabajó favorablemente con NetBeans y JDeveloper para JSF y ADF respectivamente.
5. Se realizó el desarrollo de la página para el departamento de contabilidad del IESS de manera óptima usando la tecnología JSF que nos brinda mejores prestaciones en rapidez y seguridad, Variables claves de nuestra hipótesis.
6. La hipótesis resultó afirmativa, mediante el uso del método chi-cuadrado, que arrojó el valor chi-cuadrado calculado con los datos observados y esperados de 91,85, que es mucho mayor que el valor de chi-cuadrado crítico como es de 3,84. En referencia a estos datos damos como aceptada la hipótesis.
7. La aplicación SADIESS cumple con los requerimientos planteados ofreciendo a los usuarios organización en la información de los bienes.

RECOMENDACIONES

1. Se recomienda el uso de la tecnología JSF para la creación de páginas Web para obtener rapidez y seguridad en el desarrollo, debido al conjunto de componentes que proporciona.
2. Es importante para el desarrollo de una aplicación Web basarse en el análisis comparativo, para no cometer errores al elegir la tecnología.
3. Estudiar detenidamente las tecnologías para de esta manera determinar adecuadamente los parámetros, variables e indicadores que nos ayudaran a realizar adecuadamente el análisis comparativo.
4. Es importante desarrollar con el Ide que preste las mejores opciones basándose en las tecnologías a utilizar, para de esta manera facilitar el uso y explotación de las tecnologías.
5. Implementar aplicaciones Web, utilizando la tecnología JSF, siempre que esto sea posible, para sacar provecho de los beneficios que brinda la misma.
6. Es importante determinar correctamente las variables dependientes e independientes para realizar la comprobación de la hipótesis usando el método adecuado para lo mismo, en este caso el método Chi-Cuadrado.
7. Continuar incrementando la funcionalidad del sistema SADIESS, con el fin de poder brindar nuevos y mejores servicios, en cuanto a la organización de los bienes que son asignados a los responsables de los departamentos del IESS Riobamba-Chimborazo.

RESUMEN

Se realizó un análisis comparativo de tecnologías Java en el entorno Web, mediante el desarrollo de un sistema Web para la gestión de los activos fijos y depreciación de los mismos en el departamento de contabilidad del Instituto Ecuatoriano de Seguridad Social Regional de Riobamba Provincia de Chimborazo (IESS-Regional).

Se escogió dos tecnologías: JSF Y ADF, seleccionadas mediante encuestas aplicadas a usuarios que han trabajado con estas tecnologías y foros que reflejaban el interés de los desarrolladores. Mediante módulos de prueba se fijaron parámetros de: manejo del Patrón MVC, seguridad, madurez del producto e instalación, y para la comprobación, se utilizó el método del Chi cuadrado. Observándose que Java Server Faces obtuvo 89.83% y ADF 72,88%, lo que determinó a la tecnología JSF con mejores prestaciones para el desarrollo del sistema Web.

La tecnología JSF se muestra superior brindando mejores prestaciones en acceso a datos alcanzando un 100% de calificación, eficiencia al momento de trabajar con los componentes de la GUI con un 90.09%, para el modelo controlador con 95%, en seguridad con 83%, en cuanto a la madurez del producto con 84,38% y en la instalación con 82,5%, por lo que se ha seleccionado a JSF para la realización de la aplicación web.

El análisis comparativo realizado ha permitido determinar que la tecnología JSF permite agregar rapidez y seguridad en el desarrollo de aplicaciones Web. Se recomienda a JSF para el desarrollo del sistema Web SADIESS en el IESS-Regional.

SUMMARY

Java-technologies comparative analysis in the web environment was carried out, through the web system development for the fixed assets management and depreciation in the accounting department of the Ecuadorian Institute of Regional Social Security from Riobamba Chimborazo province (IESS-Regional).

Two technologies have been chosen: JSF and ADF, selected through surveys applied to users who have worked with these technologies and forums that showed the interest of the developers.

Through testing modules parameters were set: MVC, management pattern, security, product maturity and installation, for verification, chi-square method was used.

Observing Java Server Face got 89,83% and ADF 72,88% determining the technology JSF with better benefits for the Web system development.

The JSF technology is superior providing a better access to data reaching a grade 100% efficiency at the moment to work with the GUI components 90,09%, for the controller pattern 95%, in security 83%, as far as maturity of the product is concerned 84,38% and the installation 82,5%, selecting JSF for the realization of web applic

The comparative analysis carried out has enabled to determine that JSF technology allows adding speed and security in the web applications development. The JSF use for development in SADIESS Web system in the IESS-Regional is recommended.

ANEXOS

BIBLIOGRAFIA

- [1] Barrios J, Java 2, Standard Edition, [En Línea],
<http://www.dcc.uchile.cl/~jbarrios/J2EE/node12.html>
(2009-09-21)
- [2] Barrios J, Arquitectura J2EE, [En Línea],
<http://www.dcc.uchile.cl/~jbarrios/J2EE/node14.html>
(2009-09-21)
- [3] Bench Mark, Tecnologías Faces, [En línea],
http://www.postgradoinformatica.edu.bo/enlaces/investigacion/pdf/INGSW3_50.pdf?PHPSESSID=3ab62098bf2d8aae88430bb18791f9ac
(2009-10-21)
- [4] Caraballo Alejandro, Arquitectura J2EE - Patrón MVC, [En línea],
<http://caraballomaestre.blogspot.com/2009/02/arquitectura-j2ee-patron-mvc.html>
(2009-09-21)
- [5] Comunidad de desarrolladores Mexicanos, Encuesta sobre framework MVC, [En Línea],
http://www.javamexico.org/poll/que_framework_mvc_prefieres_para_desarrollar
(2010-03-20)
- [6] Gallardo Pedro, ADF: El Framework de desarrollo de Fusión, [En línea],
http://www.bbr.cat/presentaciones/PDF/Noticias_EventosBbr/ADF_Enterprise_2_0/ADF.%20El%20framework%20de%20desarrollo%20de%20Fusion%20v3.0.pdf
(2009-09-22)
- [7] Google, Arquitectura en tres capas, [En línea],
http://www.google.com.ec/imgres?imgurl=http://oness.sourceforge.net/proyecto/html/images/three_layers_web.gif&imgrefurl=http://oness.sourceforge.net/proyecto/html/ch03s02.html&h=444&w=755&sz=19&tbnid=WymsG3FAZwigeM:&tbnh=84&tbnw=142&prev=/images%3Fq%3Darquitectura%2Bde%2Btres%2Bcapas&hl=es&usq=_pO073ekQfD3H8QAhD_RAbMcBUUY=&ei

[=yCrSS9ToC4nW9A...&ved=0CBMQ9QEwBQ](http://www.javahispano.org/forum/persistencia/es/base_de_datos_para_jsf_de_java/)

(2010-03-03)

[8] Java Hispano, Foro persistencia, [En línea],

http://www.javahispano.org/forum/persistencia/es/base_de_datos_para_jsf_de_java/

(2010-04-12)

[9] LINTI – Facultad de Informática, Struts y JavaServer Faces, cara a cara, [En línea],

<http://www.ing.unp.edu.ar/wicc2007/trabajos/ISBD/109.pdf>

(2009-09-22)

[10] Mateu Carles, Desarrollo de Aplicaciones Web, [En Línea],

<http://www.scribd.com/doc/4605108/Desarrollo-de-Aplicaciones-Web>

(2010-03-25)

[11] Oracle, ADF y Mysql con jdeveloper, [En Línea],

<http://www.zonaoracle.com/foro/index.php?topic=973.0>

(2010-04-13)

[12] Oracle, Desarrollo de una aplicación Web con JSF, [En Línea],

<http://www.scribd.com/doc/20367380/Java-Server-Faces>

(2010-04-19)

[13] Pecos Daniel, PostGreSQL vs. MySQL, [En línea],

http://danielpecos.com/docs/mysql_postgres/index.html

(2010-04-12)

[14] Programacion en Castellano, Desarrollo Rápido de Aplicaciones J2EE con Oracle ADF y MySQL en jDeveloper, [En Línea],

http://www.programacion.com/articulo/jap_adf_mysql/

(2010-04-13)

[15] Rojas Pedro, Conectividad JDBC, [En línea],

http://pedrorojas.over-blog.es/pages/Conectividad_JDBC-1356910.html

(2010-06-15)

- [16] Torres Lozano Gerardo, Tecnologías Oracle, [En línea],
<http://comunidadoraclehispana.ning.com/forum/topics/jdeveloper-es-gratis>
(2009-06-13)
- [17] Yang Shen Derek, Integración de JSF, Spring e Hibernate para crear una Aplicación Web del Mundo Real, [En Línea],
http://www.programacion.com/tutorial/jap_jsfwork/3/
(2009-09-23)
- [18] James Gosling, Plataforma Java, [En línea],
http://es.wikipedia.org/wiki/Plataforma_Java#Tecnolog.C3.ADas_Java