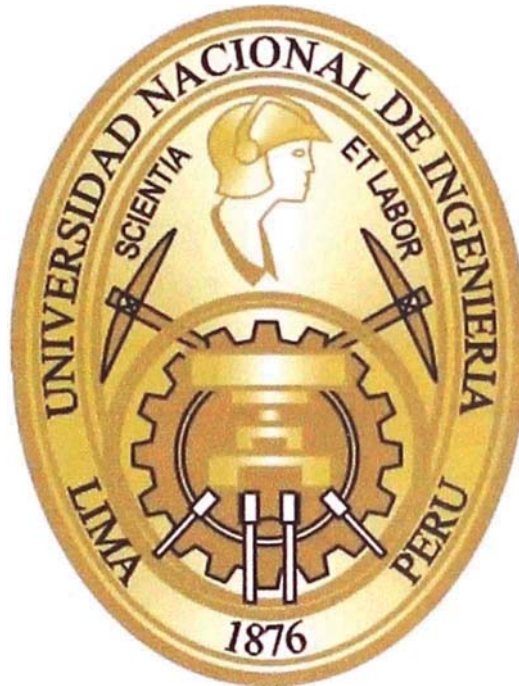


UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS



**RE DISEÑO TÉCNICO DE UN SISTEMA DE INFORMACIÓN
COMERCIAL**

INFORME DE SUFICIENCIA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS

CESAR ALBERTO GUTIÉRREZ GARAY

LIMA – PERÚ

2013

A Dios. Por darnos la oportunidad de hacerme participe de su creación.

A mis Padres y mis Hermanos. Por darme la vida y enseñarme a vivir con valores.

AGRADECIMIENTOS

Agradecimientos a los maestros que forjaron mi futuro, a las personas que sin conocerme dieron su aporte durante mi etapa de crecimiento profesional.

En especial a mis hijos y mi esposa quienes son el motor de mi vida.

INTRODUCCIÓN

Con la finalidad de incrementar los clientes anunciantes, el volumen de ventas de anuncios y optimizar los cobros de clientes morosos. La Gerencia general llevo a cabo la licitación de la implementar un Sistema Comercial que le permita gestionar las ventas y cobros de anuncios vendidos como también la configuración de la diagramación de los anuncios y bocetos, donde se busca la reducción de los tiempos de cada proceso por persona o tele operadora como la generación de las ventas hasta la impresión de los contratos.

El ingreso de la competencia como Telmex y otras empresas proveedoras de servicios publicitarios origina la reducción en las tarifas obligando en reducir los costos operativos de tele operadoras en las ventas y cobros recurrentes a los clientes morosos.

El proveedor considero para la solución, el modelo de la base de datos del sistema de anuncios y la nueva arquitectura java EE2.

RESUMEN EJECUTIVO

Este informe tiene como finalidad determinar los pasos y las soluciones más óptimas en la implementación de un sistema comercial de una empresa que vende anuncios.

La empresa cuenta con aplicaciones cliente servidor con base de Datos Oracle. En 2010 inició un proceso de migración a una arquitectura de n-capas, conservando el modelo inicial de base de datos y solo se procedió con la solución web con arquitectura java.

Redujo significativamente los tiempos de respuestas en las transacciones en las áreas de Facturación y Cobranza.

Se redujo los procesos repetitivos y también los costos de soporte y administración de las aplicaciones.

Se redujeron significativamente los tiempos de codificación y de integración con las demás interfaces planteadas en la solución inicial.

Se consiguió un alto nivel de aprobación de la calidad y de los tiempos de respuestas de las interfaces de los módulos de facturación y cobranza.

DESCRIPTORES TEMÁTICOS

Sistema Comercial

Framework

Componente

Arquitectura

Modelo MVC

Servidor de Aplicaciones

Información Comercial

Interfaz Web

4.2.1 BENEFICIOS TANGIBLES	74
CONCLUSIONES Y RECOMENDACIONES	76
CONCLUSIONES;	76
RECOMENDACIONES;	77
BIBLIOGRAFÍA	78
GLOSARIO	79
ANEXOS	80

1.4 Análisis FODA.....	16
CAPÍTULO II.....	18
MARCO TEÓRICO Y METODOLÓGICO.....	18
2.1. METODOLOGÍA Y DATOS DE LA APLICACIÓN	18
2.2. LENGUAJE Y SOFTWARE.....	19
2.3 HARDWARE Y AMBIENTES	19
2.4.2. STRUTS 2.....	24
2.4.3. SPRING FRAMEWORK.....	24
2.4.3.1. BENEFICIOS	25
2.4.3.2. ARQUITECTURA.....	27
2.4.3.3. EXTENSIONES.....	28
CAPÍTULO III.....	29
PROCESO DE TOMA DE DECISIONES.....	29
3.1. LA PROBLEMÁTICA.....	29
3.1.1 PROCESOS.....	30
3.2. PLANTEAMIENTO DEL PROBLEMA	32
3.2.1. ARQUITECTURA ACTUAL DE APLICACIONES	32
3.2.2. INTERFACES IMPLEMENTADAS EN SISTEMA COMERCIAL.....	33
3.3. ALCANCE DEL PROBLEMA	38
3.4. REDEFINICIÓN DEL PROBLEMA.....	38
3.5. OBJETIVOS DEL PROYECTO.....	42
3.6. ALTERNATIVAS DE SOLUCIÓN.....	42
a. JSF 2.0.....	47
b. Primefaces	49
c. Utilities	67
d. La integración con Java EE.....	67
e. Interfaces y Pantallas Obtenidas	68
CAPÍTULO IV	73
ANÁLISIS COSTO BENEFICIO	73
4.1 COSTO DE LA IMPLEMENTACIÓN.....	73
4.2 BENEFICIOS	73

ÍNDICE GENERAL

AGRADECIMIENTOS	3
ÍNDICE GENERAL	4
DESCRIPTORES TEMÁTICOS	7
RESUMEN EJECUTIVO	8
INTRODUCCIÓN	9
CAPÍTULO I	10
PENSAMIENTO ESTRATÉGICO	10
1.1 DIAGNOSTICO FUNCIONAL	10
1.1.1 ORGANIZACIÓN	10
1.1.2 ORGANIGRAMA	11
1.1.3. PRODUCTOS Y SERVICIOS	11
1.1.4. CLIENTES	11
1.1.5. PROVEEDORES	12
1.1.6. PROCESOS	12
MACRO PROCESOS	14
1.2. DIAGNOSTICO ESTRATÉGICO	14
1.2.1. VISIÓN Y MISIÓN DE LA EMPRESA	14
1.2.2. ANÁLISIS FODA	15
FORTALEZAS	15
DEBILIDADES	15
1.2.3. OPORTUNIDADES Y AMENAZAS	15
OPORTUNIDADES	15
AMENAZAS	15

CAPÍTULO I

PENSAMIENTO ESTRATÉGICO

1.1 DIAGNOSTICO FUNCIONAL

1.1.1 ORGANIZACIÓN

Empresa Transnacional con presencia en Inglaterra, España, Estados Unidos, Argentina, Chile y Perú, dedicada a la publicación de Directorios Comerciales impresos y electrónicos.

Con más de 50 años en el Perú, actualmente forma parte del grupo Yell, líder mundial en el mercado de directorios independientes. Esta experiencia permite brindar siempre la información que el cliente necesita.

Desde el 2000 cuenta en nuestro país con la certificación de calidad ISO y actualmente tenemos la certificación ISO 9001:2008.

Páginas Amarillas Perú genera más de 70 millones de consultas en hogares cada año y cuenta con una de las webs mas consultadas del país: la cual tiene más de 30 millones visitas al año.

1.1.2 ORGANIGRAMA

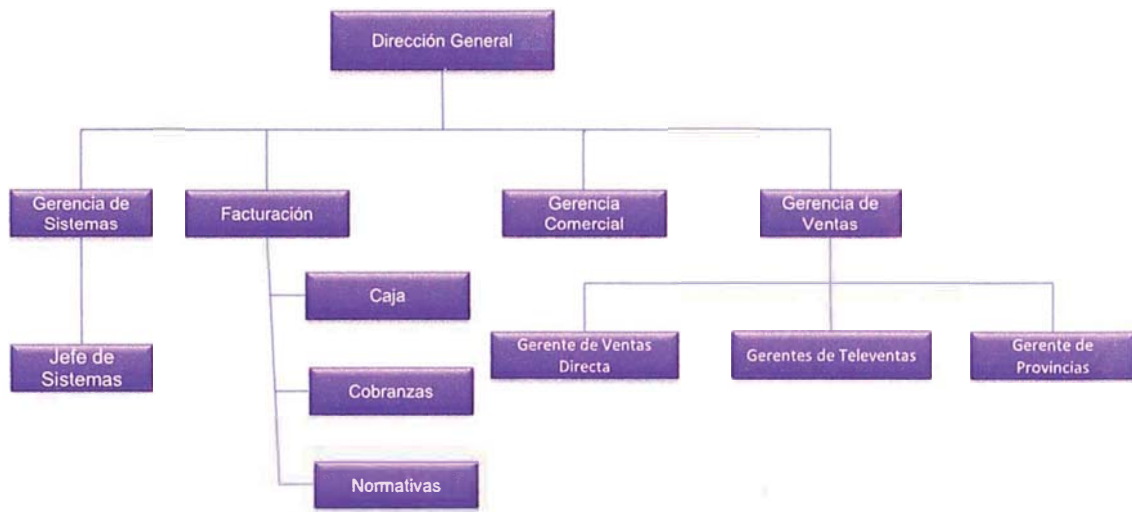


Figura 1: Diagrama Organizacional

1.1.3. PRODUCTOS Y SERVICIOS

La Empresa de anuncios comprenden los siguientes Productos:

Múltiples alternativas para atraer clientes al negocio.

- Páginas amarillas
- Amarillas Provincias
- Amarillas Online
- Páginas Blancas
- Páginas Blancas Online
- Amarillas Auto
- Negocios

1.1.4. CLIENTES

Los clientes se dividen en:

- Persona Natural:

Ciudadano con DNI

- Persona Jurídica:

Es la empresa que compra un aviso o anuncio publicitario.

1.1.5. PROVEEDORES

Podemos identificar a los siguientes Proveedores principales:

- IMPRENTA DE GUÍAS:

Son los responsables en la impresión de las guías siguiendo plantillas de diseños así como los bocetos que generan la aplicación durante la generación de la propuesta comercial.

1.1.6. PROCESOS

El siguiente modelo operacional representa el ciclo de implementación y que se puede esquematizar de la siguiente forma.

FLUJO ACTUAL

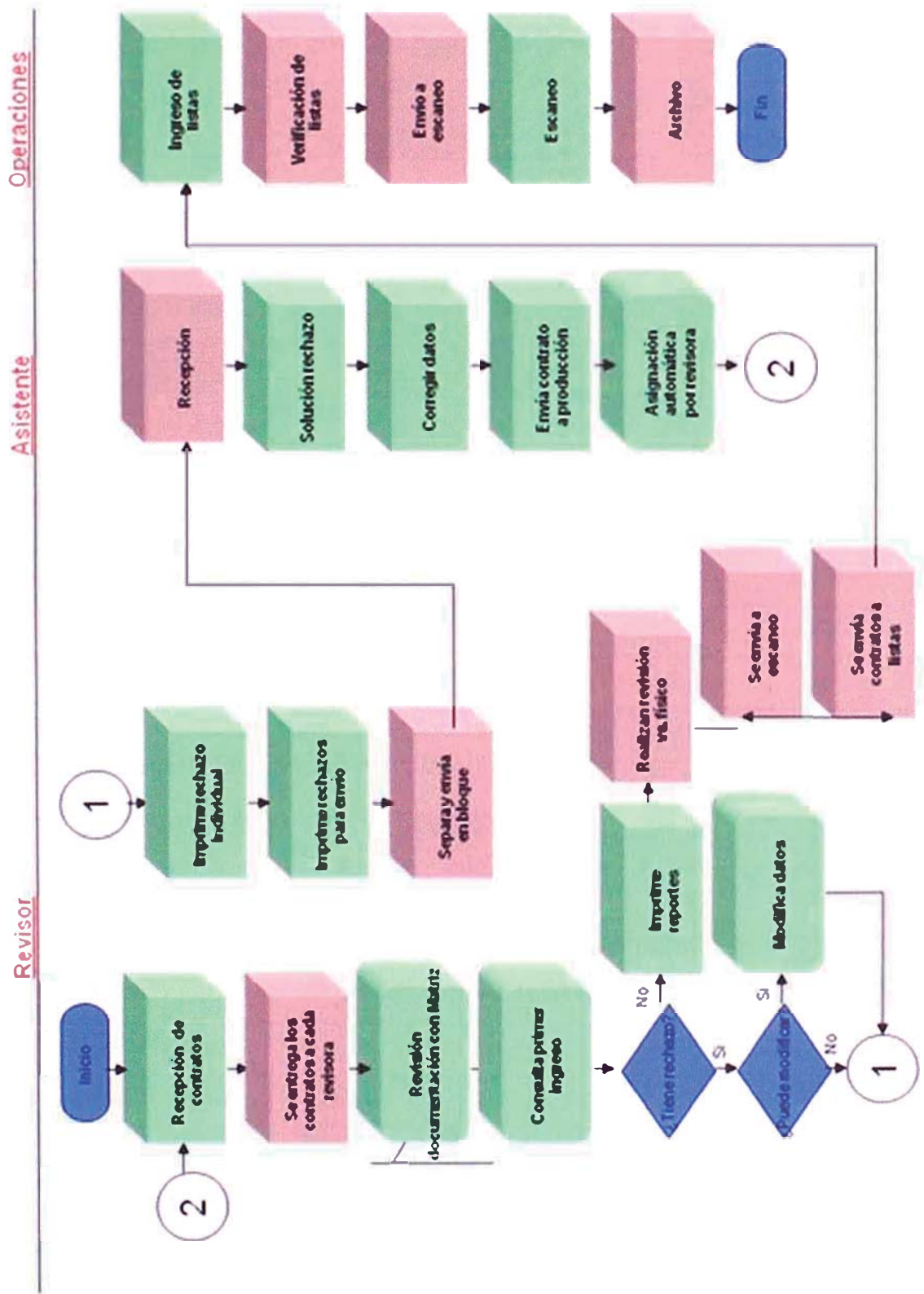


Figura 2: Diagrama Flujo de Proceso

MACRO PROCESOS

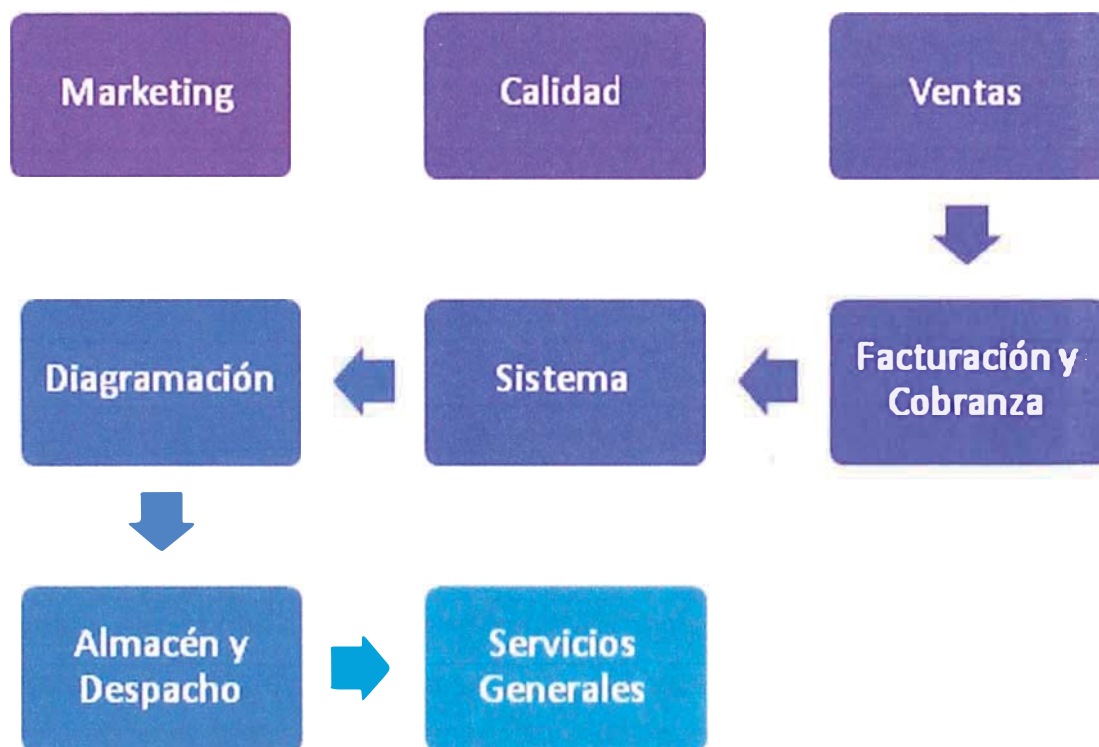


Figura 3: Diagrama Flujo de Macro Proceso

1.2. DIAGNOSTICO ESTRATÉGICO.

1.2.1. VISIÓN Y MISIÓN DE LA EMPRESA

VISIÓN

Ser el mejor puente de información comercial de compradores y vendedores independientemente del canal, del tiempo, o de la ubicación en los mercados donde operamos.

MISIÓN

Establecer los contactos entre la oferta y la demanda a través de una cartera integrada de productos publicitarios rentables y fáciles de usar, disponibles en formato impreso y online.

1.2.2. ANÁLISIS FODA

FORTALEZAS

Crecimiento anual de pesquisas de clientes para los servicios de guías impresas y publicaciones online.

Nuestro servicio y nuestros anuncios son de mayor aceptación por la cantidad de usuarios y crecimiento de más de 10 millones de visitas anual.

DEBILIDADES

Las moras en los pagos y la constante refinanciamiento por los montos facturados parciales.

Formas de pagos únicos de diferentes campañas, con ello se pierde el control de la liquidez total de las ventas por campaña.

1.2.3. OPORTUNIDADES Y AMENAZAS

OPORTUNIDADES

La creciente demanda de nuestros productos y el crecimiento de las PYME hacen que nuestro mercado sea más rentable.

AMENAZAS

El crecimiento de las empresas tiene necesidades de ubicarse en el mercado buscando medios atractivos donde se puedan promocionarse sus productos y servicios.

La creciente demanda y el aumento del número de usuarios de Internet, servicios móviles y el aumento de operadores de servicios de comunicación, hace que pueda crear nuevos servicios de páginas, orientado a su servicio.

1.4 Análisis FODA

		Oportunidades						Amenazas				
		1. Posicionar nuestros servicios en el mercado nacional	2. Diversificar nuestros servicios	3. Capturar otros mercados con nuevos servicios	4. Crecimiento de las empresas.	5. Mejorar servicio al cliente	6. mejoramientos en la calidad de los procesos	7. Buscar el cobro efectivo de los clientes en cada campaña	8. Usar TI para mejorar la productividad en la ventas	9. Fortalecer liderazgo con calidad de productos	10. Mejorar la calidad de las publicaciones online	11. Clientes con publicaciones que no cumplen con las especificaciones del anuncio.
Calidad de Productos	Fortalezas	1	2	3					8	9	10	11
Cartera de Clientes		1	2	3		5	6	7				
Procesos Integrado								7				
Precios Competitivos		1	2	3							10	11
Innovaciones tecnológicas					4			7				
Mejoras en la cobranza	Debilidades					5						
Competencias de productos similares												
Reducción de precio en las ventas								7			10	11
Planeamiento de las ventas a largo plazo											10	

Cuadro. 1 Análisis FODA (Los Números. están descritos en el siguiente cuadro).

		FORTALEZAS	
		OPORTUNIDADES	AMENAZAS
FORTALEZAS	1. Posicionar nuestros servicios en el mercado nacional.	9. Fortalecer liderazgo con calidad de productos	
	2. Diversificar nuestros servicios	10. Mejorar la calidad de las publicaciones on lines y los costos de avisos.	
	3. Capturar otros mercados con nuevos servicios		
	4. Crecimiento de las empresas.		
	5. Mejorar servicio al cliente	11. Clientes con publicaciones que no cumplen con las especificaciones del anuncio.	
	6. Mejoramientos en la calidad de los procesos para la Atención del cliente.		
	7. Buscar el cobro efectivo de los clientes en cada Campaña	12. Cliente sin retorno.	
	8. Usar TI para mejorar la productividad en la ventas y en la Reducción de gastos papelerías	13. Implementar políticas de crecimiento e innovaciones de productos on lines.	

Cuadro. 2 Matriz FODA

Conclusiones sobre el análisis FODA:

La empresa debe:

- Aprovechar el crecimiento del mercado nacional e internacional.
- Fortalecer y difundir la calidad de los productos.
- Usar las TI en innovaciones como herramientas de competitividad, para mejorar la atención al cliente, generar nuevos producto para los clientes.
- Mejorar la capacidad de recaudación de cobranzas.

CAPÍTULO II

MARCO TEÓRICO Y METODOLÓGICO

La finalidad de redefinir el diseño técnico para el sistema comercial es de lograr un producto de calidad que cumpla con los objetivos funcionales y financieros de la empresa, tanto a corto plazo como a futuro.

Se van aplicar los pasos necesarios para la selección de una buena arquitectura para su implementación que permita cubrir con las exigencias del negocio.

2.1. METODOLOGÍA Y DATOS DE LA APLICACIÓN

Detalla todos los lineamientos técnicos a considerar para el desarrollo del nuevo sistema de anuncios

Tabla 1: Capas y Licencias

Tipo de Aplicación	Web y procedimiento de Web Services para la grabación de los bocetos y anuncios diagramados
Número de Capas Arquitectura	n-capas
Licencias de software requeridas	software Open Source

2.2. LENGUAJE Y SOFTWARE

Tabla 2. Software y Extensiones

Motor de la base de datos	Oracle 11G
Servidor de aplicaciones JEE	Oracle Glassfish Server Edition v3.1.1
Plataforma de desarrollo	JEE v6.0, JSE v7.0, Eclipse 3.7, Netbean 7.1
Framework	Primefaces 2.1, 3.1 y JSF 2.0
Sistema operativo	Unix

2.3 HARDWARE Y AMBIENTES

Se requiere 3 PCs con las características y software instalado que se detallan a continuación.

La PC será utilizada por el Analista-Programador asignado al proyecto del rediseño del Sistema Comercial.

Para reducir los costos se ha definido que el software de desarrollo así como los de apoyo serán gratuitos.

2.4 FRAMEWORK Y EXTENSIONES

2.4.1. JAVA SERVER FACES (JSF)¹

¹ Antes de comenzar el desarrollo con JSF debemos conocer aquellos puntos que lo pueden hacer más largo de lo realmente necesario. Entre ellos la experiencia nos muestra los siguientes:

- Utilizar el alicate para clavar. JSF es una herramienta y como tal tiene una forma de uso. Si nos empeñamos en seguir desarrollando las páginas como siempre, intentando adaptar JSF al modo al que habitualmente desarrollamos en vez de adaptarnos a JSF complicaremos el desarrollo
- Abuso del javascript. JSF permite utilizar javascript para hacer más rápida una página html, evitando peticiones al servidor. Sin embargo la introducción de javascript en la página complica y alarga los desarrollos con JSF, y en general con jsp. La capa javascript añade etapas adicionales a la aplicación, que hace más

Framework que implementa el patrón MVC (Modelo-Vista-Controlador), es una tecnología y framework 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 como XUL.

a. ELEMENTOS DEL FRAMEWORK

Vista: Conjunto de ficheros JSP con las tag libraries de JSF Facelets (ficheros xhtml)

Otros PDLs (Page Declaration Languages)

- Describen la jerarquía de componentes JSF que conforman cada una de las páginas (pantallas) del interfaz de usuario de la aplicación.
- Vinculan los componentes JSF con los Managed Beans (objetos de respaldo)
- Se hace uso de la sintaxis del Unified Expression Language para referenciar los Managed Beans y sus atributos (`#{objeto.atributo}`)

difícil su depurado. Un consejo: la página debería poderse ejecutar sin pérdida de funcionalidad (sólo de rendimiento si se desactiva el javascript).

- La maquetación compleja también complica el desarrollo ya que obliga a utilizar muchas etiquetas y atributos, especialmente en los datatables. Si la maquetación de nuestras páginas es compleja deberíamos pensar en crear componentes JSF a medida que simplifiquen dicho trabajo.

Tomado de [Cristóbal González Almirón](#) Consultor de desarrollo de proyectos informáticos

url : <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=IntroduccionJSFJava>

Modelo: Managed Beans (objetos de respaldo gestionados por el framework JSF)

- Objetos Java (Java Beans) responsables de la lógica de la aplicación • implementada directamente en los propios Managed Beans • delegando en componentes de negocio (EJBs, Beans SPRING, Servicios Web, etc.,...)
- Responden a los eventos generados por los componentes JSF
- Controlan la navegación entre páginas (métodos de acción)

Controlador: Faces Servlet (configurado en faces-config.xml [opc. en JSF 2.0]) métodos de acción de los Managed Beans.

- Todas las peticiones HTTP del usuario pasan por el Faces Servlet
- Faces Servlet examina las peticiones recibidas, actualiza la representación del interfaz del cliente y los datos de los Managed Beans e invoca los manejadores de eventos y las acciones sobre el modelo a través de los métodos de los Managed Beans

b. ARQUITECTURA DE ASF 2.0

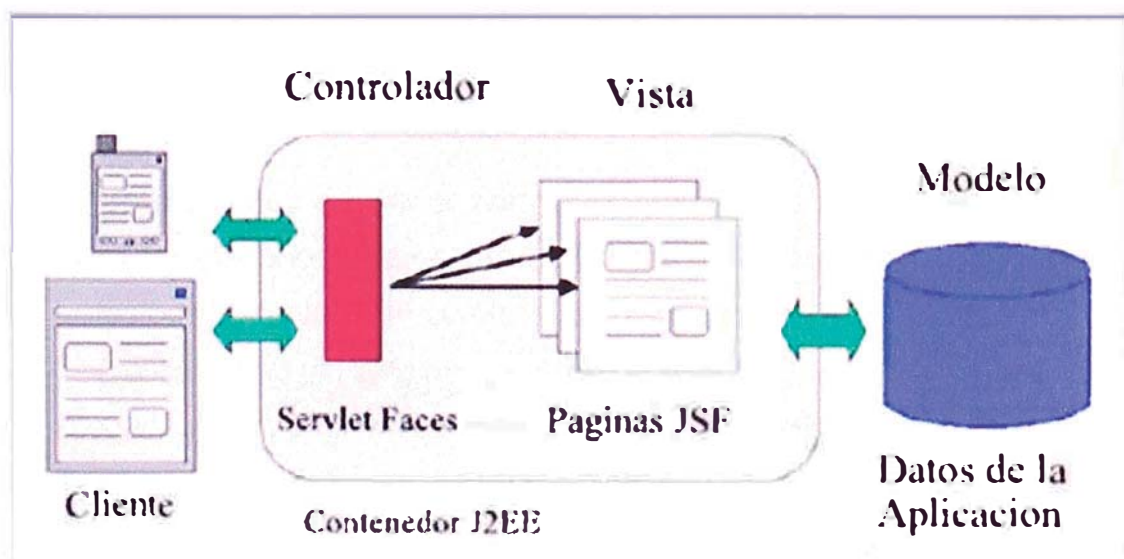


Figura 4: Diagrama de Arquitectura JSF 2.0

c. OTROS ELEMENTOS JSF

Renderizadores (render): objeto responsable de generar la representación de los componentes JSF a mostrar en los clientes y de recuperar las entradas de usuario recibidas (Strings en los parámetros HTTP) para actualizar los valores vinculados a los componentes por defecto, JSF incluye un render kit para HTML 4.0

Convertidores y validadores (converter, validator)

Convertidores:

- Al generar la presentación] transforman de forma conveniente los valores de los objetos Java vinculados a los componentes JSF en Strings (texto HTML)
- Al recuperar las entradas de usuario] transforman los Strings enviados por el cliente (parámetros HTTP) en los objetos Java vinculados a los componentes JSF según el tipo que corresponda (Integer, Double, Date, etc.)
- JSF incluye conversores para los tipos básicos
 - Paquete javax.faces.convert
 - Etiquetas <f:convertDateTime>, <f:convertNumber>, ...

Validadores:

- Objetos responsables de comprobar (antes de actualizar los atributos de los Managed Beans) que los valores recibidos y almacenados en los componentes JSF cumplen las restricciones especificadas. (validaciones del lado del servidor)
 - Cada componente JSF que reciba datos de entrada puede tener asociado 1 o más validadores
 - JSF incluye una serie de validadores estándar: <f:validateLength>, <f:validateDoubleRange>, <f:validateRegExp>, ...

d. JSF INCLUYE:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- Un conjunto por defecto de componentes para la interfaz de usuario.
- Dos bibliotecas de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- Beans administrados.

e. EXTENSIONES DE JSF

Rich Faces.- Agrega componentes visuales y soporte para AJAX.

ICEfaces.- Contiene diversos componentes para interfaces de usuarios más enriquecidas, tales como editores de texto enriquecidos, reproductores de multimedia, entre otros.

jQuery4jsf.- Contiene diversos componentes sobre la base de uno de los más populares framework javascript jQuery.

PrimeFaces.- Es una librería muy liviana, todas las decisiones hechas son basadas en mantener a PrimeFaces lo más liviano posible.

PrimeFaces es una librería muy simple que no necesita dependencias y configuraciones.

OpenFaces.- Librería open source que contiene diferentes componentes JSF, un Framework Ajax y un Framework de validación por parte del cliente.

ADFFaces, Es una interfaz propietario de Oracle que utilizan compontes y controles propios de Oracle se integra con el jdeveloper como herramienta de desarrollo.

2.4.2. STRUTS 2

Es un framework para el desarrollo de aplicaciones web, el cual hace que la implementación de las mismas sea más sencilla, más rápido, y con menos complicaciones. Además hace que estas sean más robustas y flexibles. Implementa el controlador del patrón de diseño MVC (Modelo Vista Controlador), y que podemos configurar de varias maneras; además proporciona algunos componentes para la capa de vista. Por si fuera poco, proporciona una integración perfecta con otros frameworks para implementar la capa del modelo (como Hibernate y Spring).

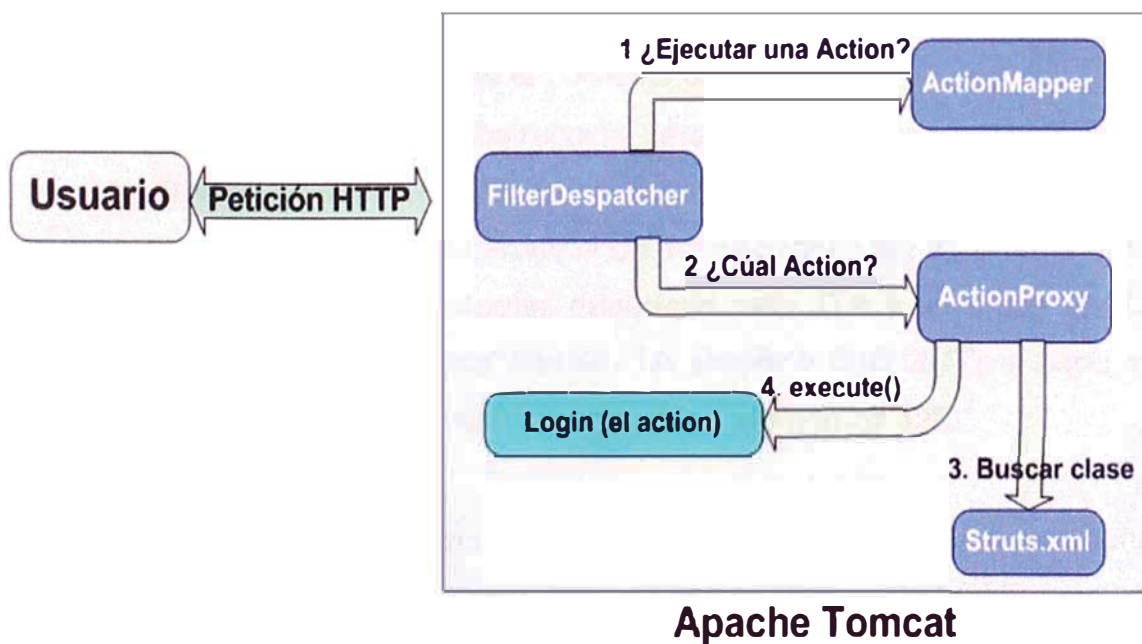


Figura 5: Diagrama Arquitectura Strut 2.0

2.4.3. SPRING FRAMEWORK

Spring es un framework de aplicaciones Java/J2EE desarrollado usando licencia de OpenSource. Se basa en una configuración a base de JavaBeans bastante simple. Es potente en cuanto a la gestión del ciclo de vida de los componentes y fácilmente ampliable. Es interesante el uso de

programación orientada a aspectos (IoC). Tiene plantillas que permiten un más fácil uso de Hibernate, iBatis, JDBC..., se integra "de fábrica" con Quartz, Velocity, Freemarker, Struts, Webwork2 y tienen un plugin para eclipse. Ofrece un ligero contenedor de bean para los objetos de la capa de negocio, DAOs y repositorio de Datasources JDBC y sesiones Hibernate. Mediante un XML definimos el contexto de la aplicación siendo una potente herramienta para manejar objetos Singleton o "factorías" que necesitan su propia configuración

2.4.3.1. BENEFICIOS

- Una potente gestión de configuración basada en JavaBeans, aplicando los principios de Inversión de Control (IoC). Esto hace que la configuración de aplicaciones sea rápida y sencilla. Ya no es necesario tener singletons ni ficheros de configuración, una aproximación consistente y elegante. Estas definiciones de beans se realizan en lo que se llama el contexto de aplicación.
- Una capa genérica de abstracción para la gestión de transacciones, permitiendo gestores de transacción añadibles (pluggables), y haciendo sencilla la demarcación de transacciones sin tratarlas a bajo nivel. Se incluyen estrategias genéricas para JTA y un único JDBC DataSource. En contraste con el JTA simple o EJB CMT, el soporte de transacciones de Spring no está atado a entornos J2EE.
- Una capa de abstracción JDBC que ofrece una significativa jerarquía de excepciones (evitando la necesidad de obtener de SQLException los códigos. Que cada gestor de base de datos asigna a los errores), simplifica el manejo de errores, y reduce considerablemente la cantidad de código necesario.
- Integración con Hibernate, JDO e iBatis SQL Maps en términos de soporte a implementaciones DAO y estrategias con transacciones.

Especial soporte a Hibernate añadiendo convenientes características de IoC, y solucionando muchos de los comunes problemas de integración de Hibernate. Todo ello cumpliendo con las transacciones genéricas de Spring y la jerarquía de excepciones DAO.

- Por su diseño el framework ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar.
- Una ventaja de Spring es su **modularidad**, pudiendo usar algunos de los módulos sin comprometerse con el uso del resto.

2 4.3 2 ARQUITECTURA

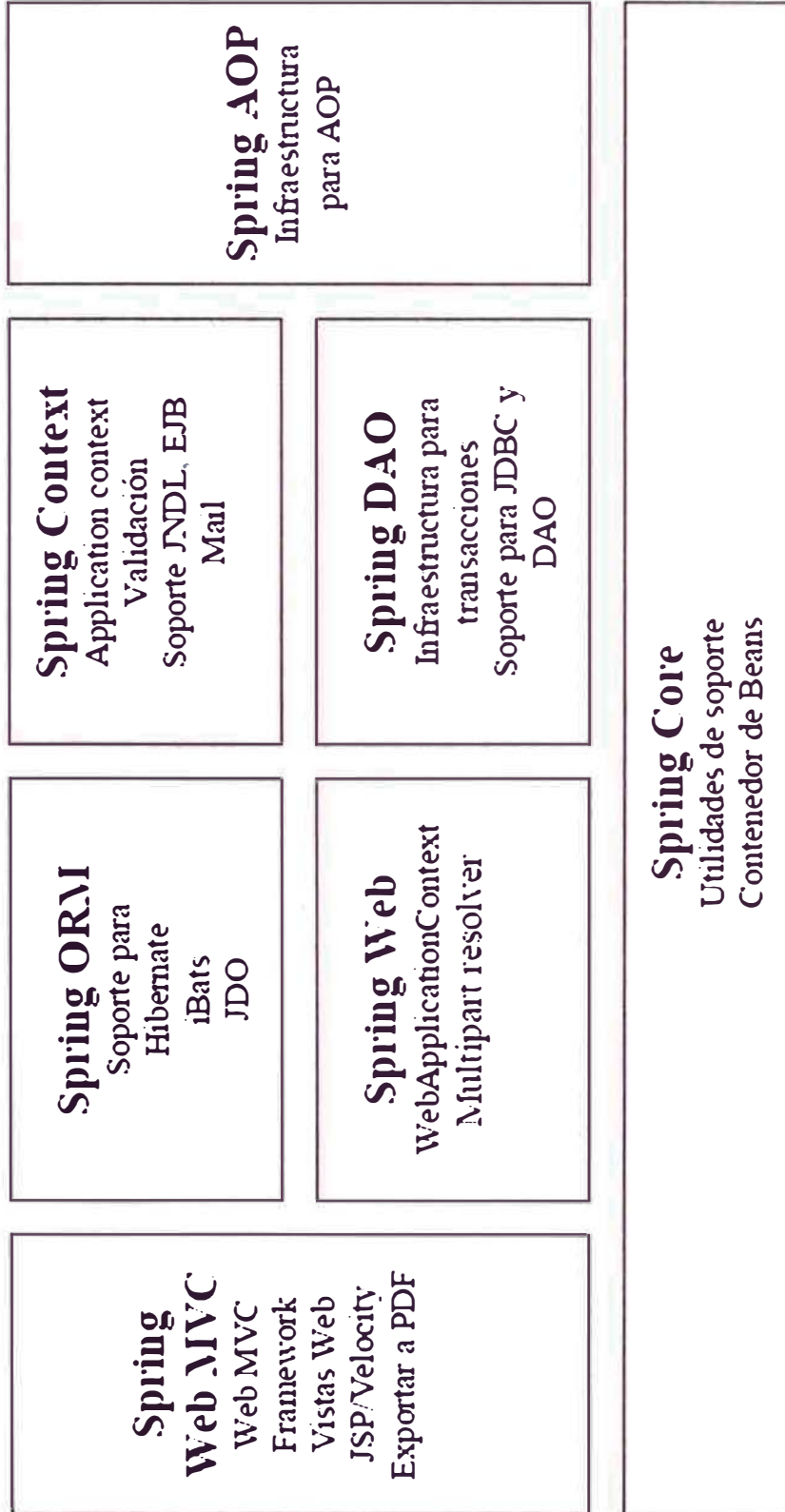


Figura 6: Diagrama Arquitectura de Spring

2.4.3.3. EXTENSIONES

Existen muchas extensiones y mejoras para construir aplicaciones basadas en web por encima de la plataforma empresarial de Java.

2.4.3.3.1. SPRING WEB

Provee el contexto para aplicaciones web. Este modulo también provee el soporte necesario para la integración de framework Struct de Yakarta.

Este modulo también se encarga de diversas operaciones web como por ejemplo: las peticiones multi-parte que puedan ocurrir al realizar cargas de archivos y la relación de los parámetros de las peticiones con los objetos correspondientes.

2.4.3.3.2. SPRING WEB MVC

Spring brinda un MVC (Model View Controller) para web bastante flexible y altamente configurable, pero esta flexibilidad no le quita sencillez, ya que se puedan desarrollar aplicaciones sencillas sin tener que configurar muchas opciones. Para esto se pueden usar muchas tecnologías ya que Spring brinda soporte para JSP, Struts, Velocity, entre otros.

CAPÍTULO III

PROCESO DE TOMA DE DECISIONES

3.1. LA PROBLEMÁTICA

El sistema de comercial se ha implementado bajo la siguiente arquitectura:

Tabla 3: Software y Extensiones Actual de la antigua implementación

Motor de la base de datos	Oracle 11G
Servidor de aplicaciones JEE	JBoos 4.2.ga
Plataforma de desarrollo	Netbean 6.7
Framework	ICEfaces 1.8 y JSF 1.2
Sistema operativo	Unix

El proyecto tuvo 3 intentos a pase de producción, después de muchas correcciones y cambios en las funcionalidades se realizo el pase a producción, pero con las quejas del usuario por la demoras en los tiempos de generación de las propuestas comerciales.

Inicialmente el proyecto estaba planteado para 100 usuarios de ventas y de cobranzas.

Durante el proceso de implementación no se realizaron pruebas de esfuerzos con la cantidad de usuario real, mostrándose claramente el interés de la reducción de costos del proyecto.

3.1.1 PROCESOS

El cliente solicita la publicación de su anuncio, se registra la propuesta comercial, pasa por el proceso de aprobación generándose el primer ingreso y contrato.

El la generación del contrato, para el solicitante se le agrupa los teléfonos como también se define la forma de pagos.

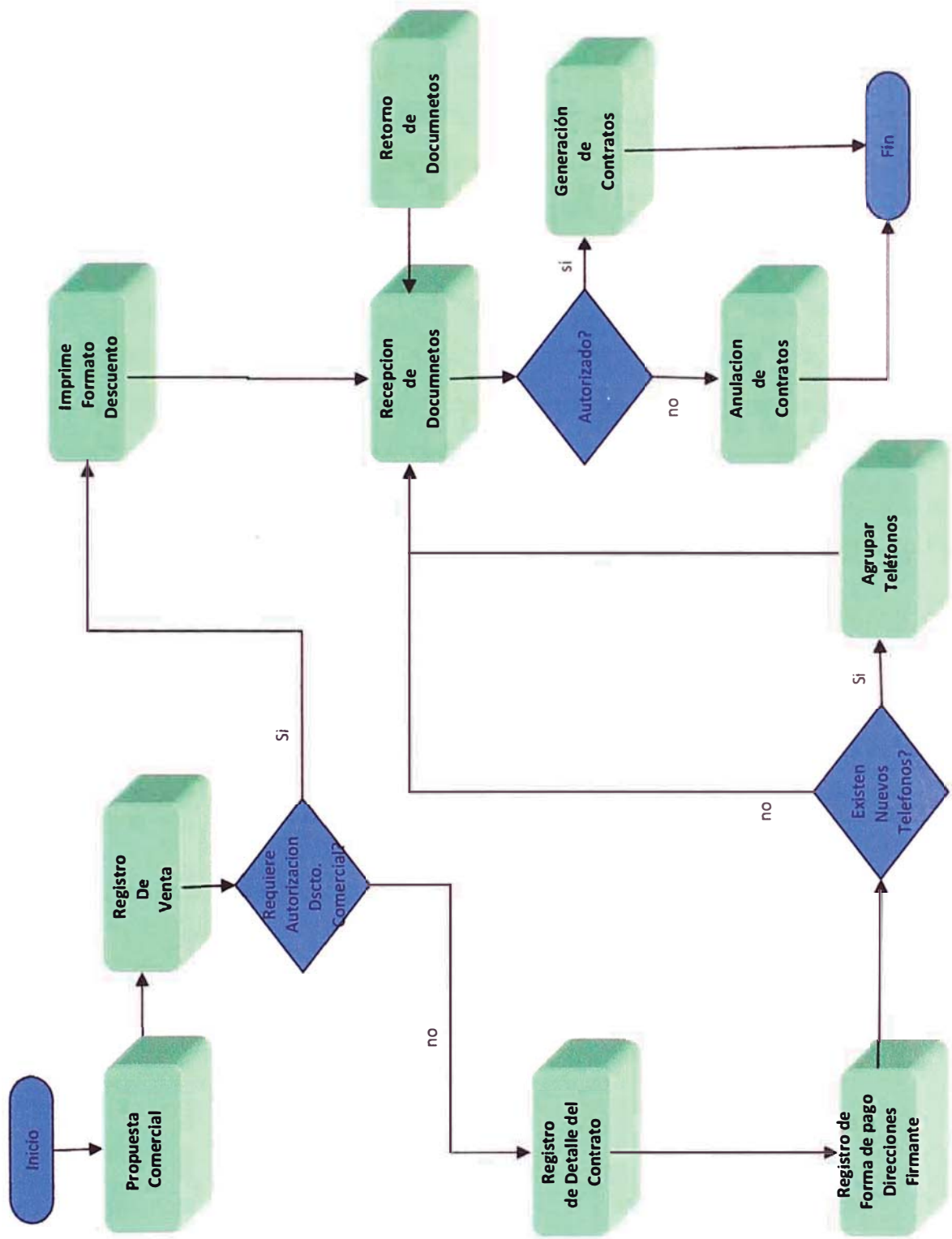


Figura 7: Diagrama de Proceso de Contrato

3.2. PLANTEAMIENTO DEL PROBLEMA

La empresa de anuncio se encuentra rediseñando su sistema de información comercial, quiere mejorar su servicio de ventas descentralizando por la aplicación web, para el cual se implementó una solución omitiendo los costos de especialistas.

El costo inicial en la solución y el mal planteamiento técnico llevaron a incrementar los costos para mejorar el performance de la solución, en consecuencia se estimó su construcción como fase inicial 7 meses, que al final se extendió a 2 años para su establecimiento.

Al final se implementó un sistema comercial con problemas de rendimientos y con alto nivel de desaprobación de los usuarios.

3.2.1. ARQUITECTURA ACTUAL DE APLICACIONES

Se planteó una nueva arquitectura y con la factorización algunos procesos e interacción con la base de datos.

Se muestra el diagrama de aplicaciones el antes (figura 8) y después (figura 9).

a). Situación anterior de la implementación.

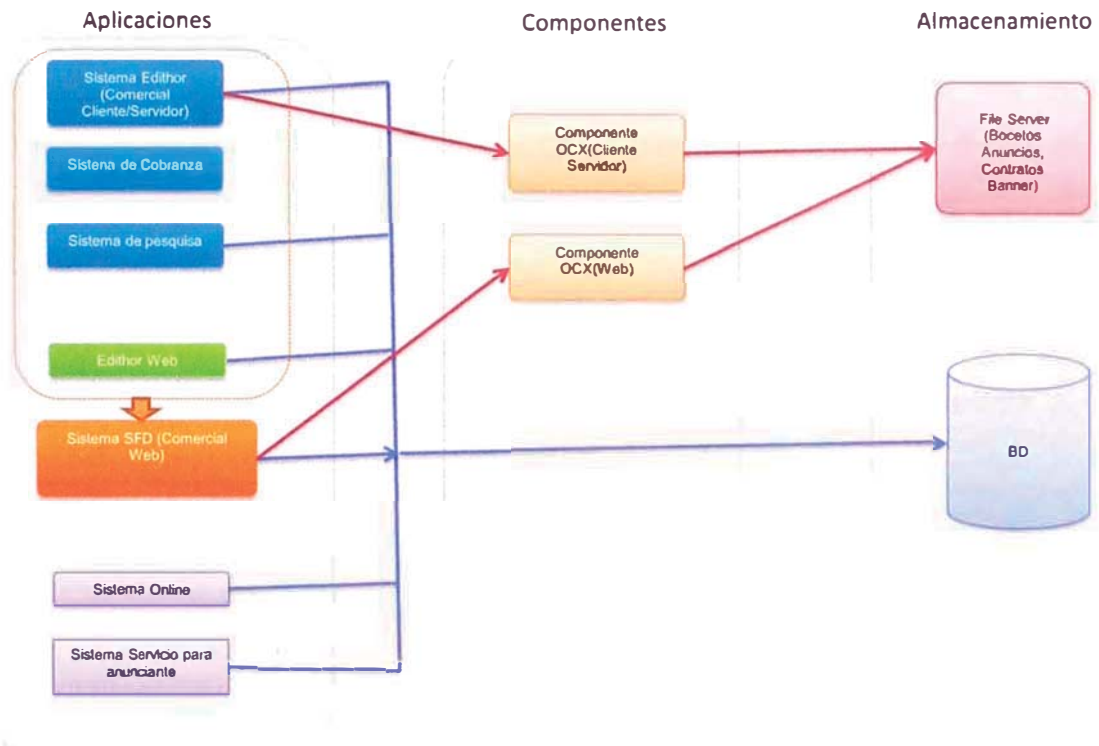


Figura 8: Diagrama Aplicaciones Actual

b). Situación de la solución con la aplicación actual.

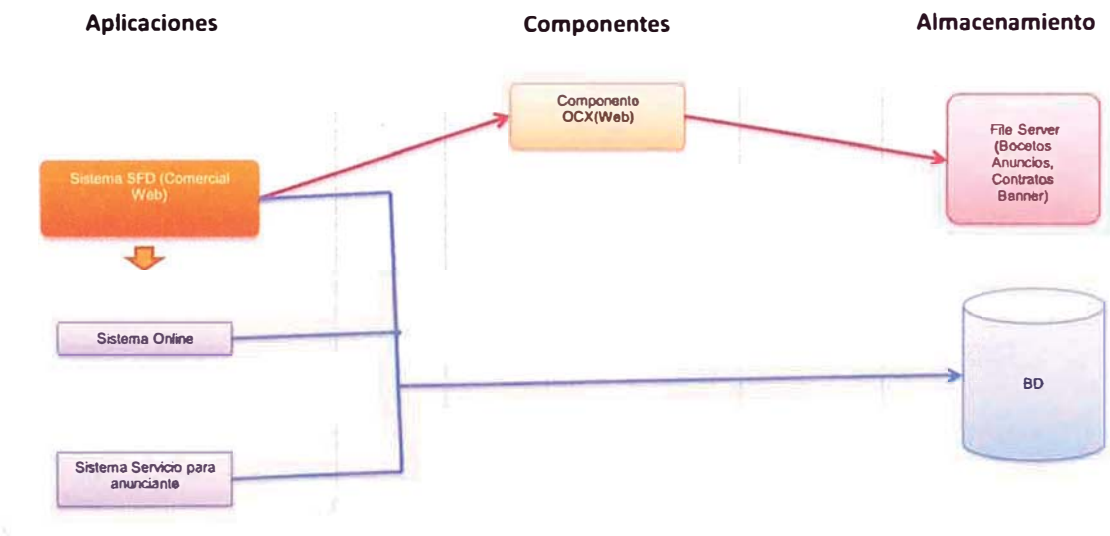


Figura 9: Diagrama Aplicaciones con la nueva implementación

3.2.2. INTERFACES IMPLEMENTADAS EN SISTEMA COMERCIAL

A continuación se muestran las pantallas principales donde muestran mucha carga de transacciones por la forma como se planteo la solución.

Propuesta Comercial

En esta pantalla se registra la propuesta al cliente, se hace una venta simulada de los productos que necesita para la promoción de su aviso, cada propuesta tiene un ítem donde se detalla las campañas, guías y promociones de cada paquete.

Si el usuario esta conforme se graba la transacción pasando por un proceso de aprobación donde se realizan los contratos con la información de la propuesta y de los ítem que se generaron.

Propuesta Comercial

Ficha: 8488696 Loc Contrat.: 00001 - LIMA Mto Bruto Real: 93,610 Mto IAP Relq: 89,553 % Dcto IAP Prop.: 4.58
 Estado: VENTA CONSOL Cliente: 50298 - FABRICA DE FIDEOS EL TRIUNFO S. Mto Bruto Relq: 93,851 % Inc Precios: 0.26 Mto Dcto IAP Prop.: 4,298
 Origen: CARTERA Firmante: Mto Descuento: 4298 Mto Inc Precios: 241 Calificación Financiera: **A**
 Campaña: LTM2012 Cargo: Mto Neto Real: 89,312 % Dcto IAP: 4.58
 Fecha Pago: 19/04/2012 Fecha Venc: 31/01/2012 Nro Cuotas: 12 Ítem con Precio: Tlo - Anual Buscar Foto

Propuestas Detalle Contacto Encuesta Histórico Online Rubros Incidencias Avisos Bocetos Escaneo Cobranzas Marketing Abonados Comprobantes

Nro	Topo	Total Pago	Dcto.Mkt	%Dcto.Mkt	Dcto.Com.	%Dcto.Com	Dcto.Tot	%Dcto.Tot	%Incr AM	%Incr BL	%Incr OL	%Incr BO	%Incr NO	Folio	Mt. Cuota
Topo		77,799.00	0.00	0.00	3,008.00	3.72	3,008.00	3.72	-17.27	-1.09	1.57	4.10	-1.09	12	6,483.25
Prop		80,807.00	0.00	0.00	0.00	0.00	0.00	0.00	-14.07	2.74	5.50	6.12	2.74	12	6,733.92

Ítem	Prod	Guía	Edic	Paquete	Publicación	Rubro	Localidad	M.Bruto	Prom.	Dcto	M.Net
1	AM	LIMA	2013		1.AP F	00808 FIDEOS	00001 LIMA	35,627.00		0.00	35,627.00
2	AM	LIMA	2013		4D F	02113 PASTAS	00001 LIMA	14,485.00		0.00	14,485.00
3	AM	LIMA	2013		INDAL R			204.00		0.00	204.00
4	AM	LIMA	2013		APR O	00808 FIDEOS	00001 LIMA	2,905.00		0.00	2,905.00
5	AM	LIMA	2013		LLORA O		00001 LIMA	350.00		0.00	350.00
6	AM	LIMA	2013		LLORA O		00001 LIMA	350.00		0.00	350.00
7	AM	LIMA	2013		LLORA O		00001 LIMA	350.00		0.00	350.00
8	AM	LIMA	2013		LLORA O		00001 LIMA	350.00		0.00	350.00

 Dcto General: 1.00 Cuota P Lista: 5,734.00 Total: 80,807.00 0.00 80,807.00

Figura 10: Pantalla Propuesta Comercial

Si la Transacción es exitosa el sistema responde mostrando un nuevo número de folio generado.

Primer Ingreso (Contratos)

Con el número de folio generado por la propuesta comercial, se genera el primer ingreso o contrato, donde el usuario puede verificar la información registrada como también hacer ajustes en los ítems o productos y los montos de descuento comercial aplicable para cada cliente dependiendo de su calificación.

El sistema valida de acuerdo al tipo de cliente con respecto a la generación de pagos, puntualidad respectiva generando una calificación que va desde la A hasta la Z, que es un cliente de alto riesgo, para este caso pasa por un proceso de evaluación y de negociación generándose un contrato con observaciones.

The screenshot shows a web application window titled 'Registrar Contrato'. It contains the following information:

- Contract Details:**
 - Ficha: 3482696
 - Campaña: LIM2012
 - REPORTADO
 - Loc. Contratación: 00001 - LIMA
 - Grupo: DMO1 - DIGITAL MEDIA 1
 - Calificación Financiera: **A**
 - Origen: CARTERA
 - F.Venta: 02/05/2012
 - Cliente: 50298 - FABRICA DE FIDEOS EL TRIUNFO S A
 - Vendedor: 0970 - ALICIA DEZA
 - Foto: 1600016
 - F.Reporte: 03/05/2012
 - Modificación S/V
- Financial Summary:**
 - Monto Bruto: 975
 - Dcto. Comercial: 0
 - Neto Comercial: 975
 - Dcto. Financiero: 0
 - Neto Total: 975
 - Dcto. MKT: 0
- Table of Items:**

Item	Prod	Guia	Edic	Paquete	Publicación	Rubro	Localidad	M.Bruto	Prom.	Dcto. Mkt	Dcto. Com	M.Netto				
<input type="checkbox"/>	1	ADP	NO	LIMA	NO	2013	1	R	00001	ABARROTES	00001	LIMA	975.00	0.00	0.00	975.00
- Summary and Actions:**
 - Total: 975.00
 - Dcto. Com: 0.00
 - M.Netto: 975.00
 - Buttons: Habilitar, Continuar, Borrar, Eliminar, Imprimir, Asignar, T.P. a todo, Nueva, Guardar, Cerrar

Figura 11: Pantalla Registro de Contrato

Bocetos

Esta pantalla permite la generación de los bosquejos automáticos de los anuncios que luego se van a imprimir o enviar en forma de PDF a un servidor de archivos.

Este sistema está integrado con un componente implementado en C++. Las tipografías se generan con las reglas de negocios y por el tipo de producto que se adquiere, de acuerdo a ello se define las tipografías que están almacenados por la base de datos.

En los textos o filas se registra los tipos y el texto, que puede ser:

1. Razón social
2. La dirección
3. Teléfonos del anunciante
4. Textos descriptivos del anuncio como la publicidad.

Favorites Diseño de Normalizado y Listas

Prd: AM Guía: AREQUIPA Item: 2.25T1-R Cod.: 415024 Rubro: 02394-PODÓLOGOS Loc.Publicación: 05008-AREQUIP
 Folio: T157501 Ficha: 785713t Cliente: DNI: 29445261 - AGUILAR YUPANOUI VICTORIA
 Firmante: 29445201 - AGUILAR YUPANOUI VICTORIA Dir.Contra: AV PERU NRO 206 JOSE LUIS BUSTAMANTE Tif: (054) 465094

Tipo	Texto
LBU - Azul 100%	CLINICA PODOLOGICA
LBU - Azul 100%	& DOSHA SPA
LTX - Negro 100%	Especialista a sus Pies...!-Pdga.Diplomada y
LTX - Negro 100%	Colegiada,ofrece servicio integral del Pie.!
LTX - Negro 100%	Deje sus Pies en manos del Profesional
LCD - Amarillo 20%	Av. Perú 206 Urb. Fecia JLB y Rivero-Arequipa
LCT - Amarillo 20%	- 989 635395 - 958 588329 -(054) 66 8642

**CLINICA PODOLOGICA
& DOSHA SPA**

Especialista a sus Pies...!-Pdga.Diplomada y
Colegiada,ofrece servicio integral del Pie.!
Deje sus Pies en manos del Profesional

**Av. Perú 206 Urb. Fecia JLB y Rivero-Arequipa
- 989 635395 - 958 588329 -(054) 66 8642**

Visualizar Ficha Grabar Cerrar

Existen avisos IAP (normalizados). Si desea copiar su contenido deberá dar click en el botón [Copiar Texto del IAP]

Figura 12: Pantalla Diseño de Boceto

Lado Derecho

Esta pantalla Muestra la forma de pago, las cuotas, montos de pago, medio de pago. Como también el agrupamiento de pago por el número de teléfono.

Fecha: 848896 Campaña: LDM2012 CARTERA Grupo: DM01 - DIGITAL MEDIA 1 Vendedor: 0970 - ALICIA DEZA Calificación Financiera: **A**

Folio: 2213261 Estado: VENDIDO Fecha Venta: 21/02/2012 Fecha Reporte: 22/02/2012 Pagare:

Monto Bruto: 112,058.00 Dcto.Comercial: 8,850.00 Dcto.Financiero: 0,00 Monto Neto: 103,208.00

Ciente: 50298 - FABRICA DE FIDEOS EL TRIUNFO S A Nombre Comercial:

Documento: N.L.T. 20100286981 N.L.T. 20100286981 Dir Contrat: JR PASTAZA NRO 757 BREÑA LIMA LIMA LLOS

Firmante: -40780300 - LOO KOC WA Tif. Dir Facturación: JR Pastaza NRO 757 BREÑA LIMA LIMA LLOS LIMA - LIMA Contratación

Atención a Igual a Firmante N° de operación: **VISA** Planilla

Forma Pago	Cuotas	Monto	Medio Pago	Fecha Pago	Teléfono	Lcc.	F/B	Nro. Pro.	D.F.
<input type="checkbox"/> CC - CREDITO	12	103,208.00	DS - PAGO DIRECTO	01/01/2013	4239834	00001	F	0	<input type="checkbox"/> JR Pastaza NRO 757

Botones: **Nueva** **Grabar** **Cerrar**

Figura 13: Pantalla Lado Derecho

También define la fecha de pago ya sea por cuotas o en efectivo, la operación puede estar asociado con el numero de operación cuenta visa u otras tarjetas. Siendo el firmante la persona que acredita la transacción.

3.3. ALCANCE DEL PROBLEMA

Para la solución se optó por usar ICEFaces 1.8 con JSF1.2, la mala integración y la mala reformulación de las pantallas de los flujos principales como son los Contratos, Primer Ingreso, Propuesta comercial y el Diseño de Boceto, interfaces principales y que tiene mucha carga por ser el proceso **core del negocio** y la cara hacia los vendedores.

3.4. REDEFINICIÓN DEL PROBLEMA

Los tiempos de respuestas con respecto a las ventas y a las continuas quejas de los usuarios con respecto a las interfaces principales del módulo comercial, debido a la cantidad de productos que pueden estar asociados a una propuesta comercial.

Para su evaluación más detallada seguiremos el diagrama causa y efecto (Diagrama de Ishikawa²) para ello se han determinado 6 categorías.

1. Estimaciones
2. Recursos Hombre

² Los Diagramas Causa-Efecto ayudan a los estudiantes a pensar sobre todas las causas reales y potenciales de un suceso o problema, y no solamente en las más obvias o simples. Además, son idóneos para motivar el análisis y la discusión grupal, de manera que cada equipo de trabajo pueda ampliar su comprensión del problema, visualizar las razones, motivos o factores principales y secundarios, identificar posibles soluciones, tomar decisiones y, organizar planes de acción.

El Diagrama Causa-Efecto es llamado usualmente Diagrama de "Ishikawa" porque fue creado por Kaoru Ishikawa, experto en dirección de empresas interesado en mejorar el control de la calidad; también es llamado "Diagrama Espina de Pescado" porque su forma es similar al esqueleto de un pez. Está compuesto por un recuadro (**cabeza**), una línea principal (**columna vertebral**), y 4 o más líneas que apuntan a la línea principal formando un ángulo aproximado de 70° (**espinas principales**). Estas últimas poseen a su vez dos o tres líneas inclinadas (**espinas**), y así sucesivamente (**espinas menores**), según sea necesario.

3. Recursos Equipos
4. Frameworks.
5. Metodologías
6. Modelo de Base de Datos.

Definición de las causas del problema por categorías

3.4.1 Estimaciones

- a. No hubo un completo levantamiento de observación, ni se estimaron los tiempos de construcción de manera objetiva.
- b. No se llevaron al nivel de detalle para cuantificar con mayor certeza los tiempos de codificación y las pruebas unitarias e integrales.
- c. No se establecieron indicadores de estimación para la construcción, de la migración de plataforma Cliente Servidor (Power builder) a Web (Java), la estimación y los tiempos de construcción muestran diferencias por su naturaleza y complejidad.

3. 4.2. Recursos Hombre

- a. Poca experiencia en la arquitectura. Debido a la tecnología era complicado encontrar en el mercado personas capacitadas que tenga conocimiento de las herramientas de ICEfaces y JSF 1.2.
- b. Los recursos tanto como los Analista Programadores no cubrían toda la envergadura del proyecto.
- c. El desconocimiento del gestor del proyecto ("Personal del Cliente") sobre los alcances del proyecto. Se crearon sobre la marcha nuevos

requerimientos incrementando la complejidad y los nuevos alcances de los requerimientos.

3.4.3. Recursos Equipos

a. Los equipos como PC de desarrollo, no cumplían con las necesidades para atender el desarrollo debido a que el software que se utiliza para la implementación consume muchos recursos.

3.4.4. Frameworks.

a. No se evaluó el performance y los tiempos de respuestas de los diferentes framework, así como sus extensiones (Interfaces web) para la implementación.

3.4.5. Metodologías

a. No se empleó una metodología de análisis orientada a un objeto, todo se basó en la copia de códigos propios de Power builder sin completar un buen análisis para el caso.

b. Demora en la implementación

3.4.6. Modelo de Base de Datos.

a. No se creó un nuevo diseño de base de datos con un nuevo modelamiento enfocado al framework a trabajar. Para este caso se utilizó JSF utilizando EE2. JPA obliga a establecer reglas de normalización más precisas para establecer

b. No se ha definido un nuevo modelo de base de datos, solo se está reutilizando con la antigua.

c. Se está re-usando el modelo de base de datos antiguo con errores de normalización que fueron creados y parchados productos de mantenimientos que fueron definidos sin ningún criterio de integración.

Diagrama de Causa Efecto

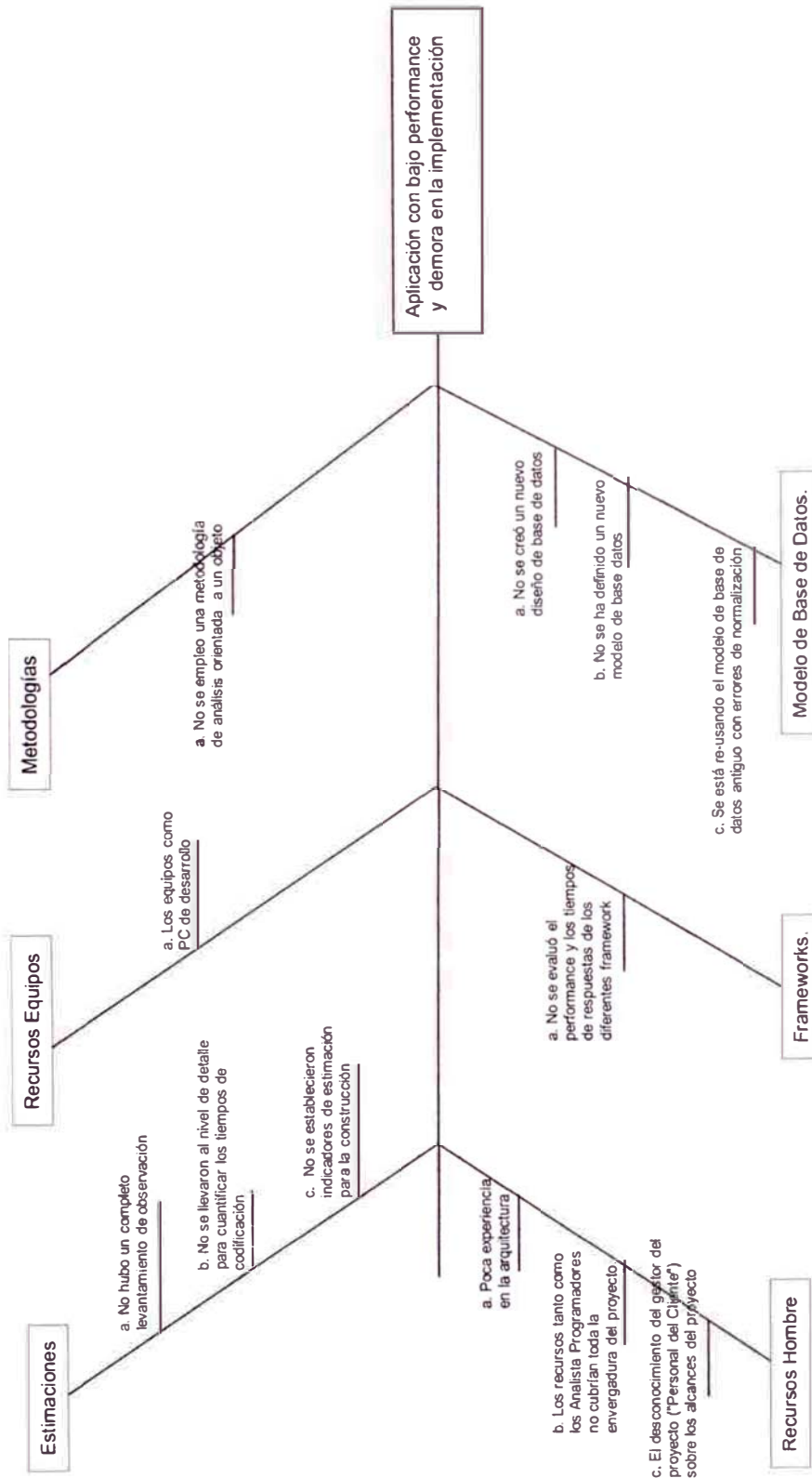


Figura 14: Diagrama Causa Efecto

3.5. OBJETIVOS DEL PROYECTO

Optimizar los tiempos de respuestas y mejorar la presentación de las interfaces principales.

Definiciones de patrones de construcción y buena práctica en desarrollo

3.6. ALTERNATIVAS DE SOLUCIÓN.

Para rediseño se han planteado las siguientes alternativas utilizando framework y sus extensiones:

Alternativa SOLUCIÓN 1:

Para esta alternativa Utilizar JSF 2.0 y además se va hacer una selección de las posibles extensiones dependiendo de las necesidades del cliente.

Ventajas

- Una de las principales ventajas de utilizar JSF es que ofrece una separación clara entre comportamiento y presentación. Una aplicación Web basada en JSF permite mapear peticiones HTTP a eventos y gestionar el estado de objetos.

Alternativa SOLUCIÓN 2:

Utiliza el framework Strut 2.0

Ventajas

- JSF trata la vista (el interfaz de usuario) de una forma algo diferente a lo que estamos acostumbrados en aplicaciones web. Sería más similar al estilo de Swing, Visual Basic o Delphi, donde la programación del interfaz se hace a través de componentes y basada en eventos (se pulsa un botón, cambia el valor de un campo, ...).

Alternativa SOLUCIÓN 3

Utilización de SPRING FRAMEWORK

Ventajas

- Una ventaja de Spring es su modularidad, pudiendo usar algunos de los módulos sin comprometerse con el uso del resto.

3.7. TOMA DE DECISIÓN

3.7.1. PESOS PARA LA EVALUACIÓN

Para determinar algún criterio para la selección del framework y las extensiones se utilizan valores que van desde 1 has 5 según criterio de expertos para de alguna manera dar un peso en los indicadores.

Tabla 4: Pesos para la evaluación de criterios de selección .

Descripción del peso	Peso
Muy malo	1
Malo	2
Bueno	3
Muy Bueno	4
Excelente	5

3.7.2. SELECCIÓN DEL CRITERIO DE EVALUACIÓN

Para la selección del criterio de evaluación se basen en el equipo técnico del proyecto y de las pruebas de conceptos del nuevo sistema.

3.7.3 ELECCIÓN DEL FRAMEWORK

Se puede definir indicadores para la evaluación como:

Presentación de la Interface el framework

- **Tiempo de entrenamiento**, bajo costo de capacitación del framework a los desarrolladores.
- **Confianza**, el framework no presenta bug o error en los tiempos de ejecución.
- **Performance**, tiempo de respuesta optimo en transacciones.
- **Costo de Implantación**(Se utiliza los pesos en forma inversa, a menor costo se le asigna el mayor puntaje, y a mayor costo menor puntaje)

- **Tiempo de Implementación** (Se utiliza los pesos en forma inversa, a menor tiempo se le asigna el mayor puntaje, y a mayor tiempo menor puntaje)
- **Flexibilidad**, capacidad de adecuación de cambios.
- **Soporte**, apoyo de la empresa proveedora de interfaz web(primefaces) en la solución de bug y errores posibles.
- **Aprendizaje**, capacitación de la interfaz web a los desarrolladores.
- **Reusabilidad**, reutilización de solución web o componente(tag) para ser reutilizados en otras interfaz HTML
- **Escalabilidad**, capacidad de versionamiento de los controles o componentes web.
- **Extensiones disponibles**, interfaces web (Primefaces, IceFaces, etc) disponible para ser integradas en la aplicación.

Nota: Los pesos varían desde 1: muy mal, hasta 5: excelente, tal como se indica en la Tabla 4.

Tabla 5: Selección de FrameWork

Nro.	Descripción	FRAMEWORK		
		JAVA SERVER FACES (JSF)	STRUTS 2	SPRING FRAMEWORK
1	Tiempo de entrenamiento	3	2	3
2	Confianza	3	3	3
3	Performance	4	2	3
4	Costo de Implantación	4	2	3
5	Tiempo de Implementación	3	3	4
6	Flexibilidad	4	3	4
7	Soporte	3	2	3
8	Aprendizaje	3	3	2
9	Reusabilidad	4	3	3
10	Escalabilidad	4	3	3
11	Extensiones disponibles	4	2	2
Total		39	28	33

De acuerdo al cuadro el framework con mayor peso es **JAVA SERVER FACES (JSF)**

3.7.4 ELECCIÓN DE LA EXTENSIÓN WEB DEL FRAMEWORK SELECCIONADO

Indicadores evaluaciones de las extensiones del framework(Entorno web)

- **Performance**, tiempo de respuesta en las consultas y envío de las transacciones.
- **Tiempo de Implementación**(Se utiliza los pesos en forma inversa, a menor tiempo se le asigna el mayor puntaje, y a mayor tiempo menor puntaje)
- **Soporte**, apoyo de la empresa proveedora de framework en la solución de bug y errores posibles.
- **Aprendizaje**, capacitación de la interfaz web a los desarrolladores.
- **Calidad de presentación web**, vistas amigable y intuitivas para el usuario.
- **Gestión de estilos con la interface**, los estilos son personalizables y se extienden en toda la aplicación del proyecto.
- **Flexibilidad**, capacidad para personalización de los componentes.
- Integración con los componentes JQuery y Ajax.
- Componentes y controles para la implementación.

Nota: Los pesos varían desde 1: muy mal, hasta 5: excelente, tal como se indica en la Tabla 4.

Hacemos la sumatorias de los pesos de los indicadores:

Tabla 6: Selección de Extensiones de JSF 2.0

Nro.	Descripción	EXTENSIONES DEL JSF 2.0			
		Rich Faces	ICEfaces	PrimeFaces	ADFFaces
1	Performance	3	2	4	3
2	Tiempo de Implementación	2	3	4	3
3	Soporte	3	3	3	3
4	Aprendizaje	3	3	3	3
5	Calidad de presentación web.	2	3	4	3
6	Gestión de estilos con la interface.	2	2	4	2
7	Flexibilidad	3	3	4	3
8	Integración con los componentes JQuery y Ajax.	4	3	4	3
9	Componentes y controles para la implementación.	3	3	5	3

Total	25	25	35	26
--------------	----	----	-----------	----

De acuerdo con el cuadro se muestra que **PrimeFaces** es la mejor **Framework** seleccionado es **JSF** y la **extensión** a utilizar es **Primefaces**

Tabla 7: Software y Extensiones Actual de la Nueva implementación

Motor de la base de datos	Oracle 11G
Servidor de aplicaciones JEE	Oracle Glassfish Server Edition v3.1.1
Plataforma de desarrollo	JEE v6.0, JSE v7.0, Eclipse 3.7, Netbean 7.1
Framework	JSF 2.0
Extensiones	Primefaces 2.1, 3.1
Sistema operativo	Unix

3.7.5 DESARROLLO DE LA ALTERNATIVA SELECCIONADA

De acuerdo a la alterativa de selección se desarrollará la arquitectura JSF y su extensión Primefaces de JSF.

a. JSF 2.0

JavaServer Faces (JSF) es una tecnología y framework para aplicaciones Web basadas en el patrón MVC que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. La interfaz de usuario se realiza mediante componentes y gestión de eventos.

Los principales componentes de JSF son:

- Un API para representar componentes de interfaz de usuario (UI) y gestionar su estado, manejar eventos, validación del lado servidor, conversión de datos, definir navegación de páginas, soporte de internacionalización y accesibilidad.
- Dos Custom Tag Library para representar los componentes UI.

Descripción Resumida funcional de JSF

En general, JSF es como cualquier otra aplicación web. Se compone de lo siguiente:

- Una serie de JSPs (aunque como dije antes, no se limita a que la presentación sea únicamente JSPs)
- Una serie de backing beans, que son JavaBeans con una serie de propiedades y funciones que definen un componente de la UI de una página.
- Un fichero de configuración, que define las reglas de navegación y configura los beans.
- El web.xml
- Opcionalmente, una serie de objetos que actúan de validadores, conversores, listeners, ...
- Una serie de custom tags para representar los componentes sobre la página.

Una aplicación JSF se encarga de procesar los eventos lanzados por componentes JSF. Estos eventos son provocados por acciones de

usuario, por ejemplo, cuando el usuario pulsa un botón se produce un evento. El programador decide qué hará la aplicación JSF cuando se lanza el evento. Esto se hará creando listener de eventos (managed beans). La siguiente imagen ilustra lo comentado.

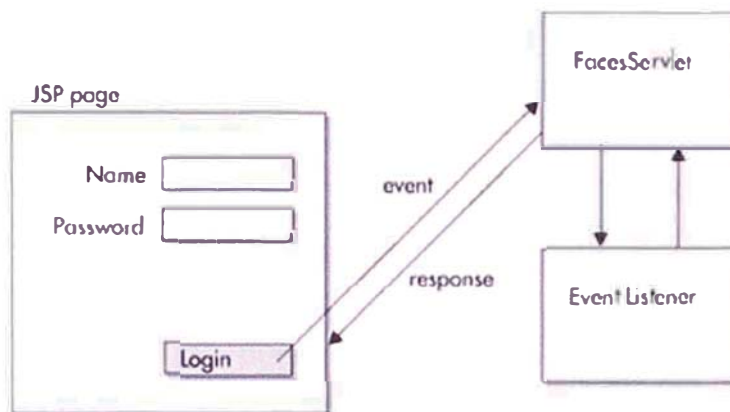
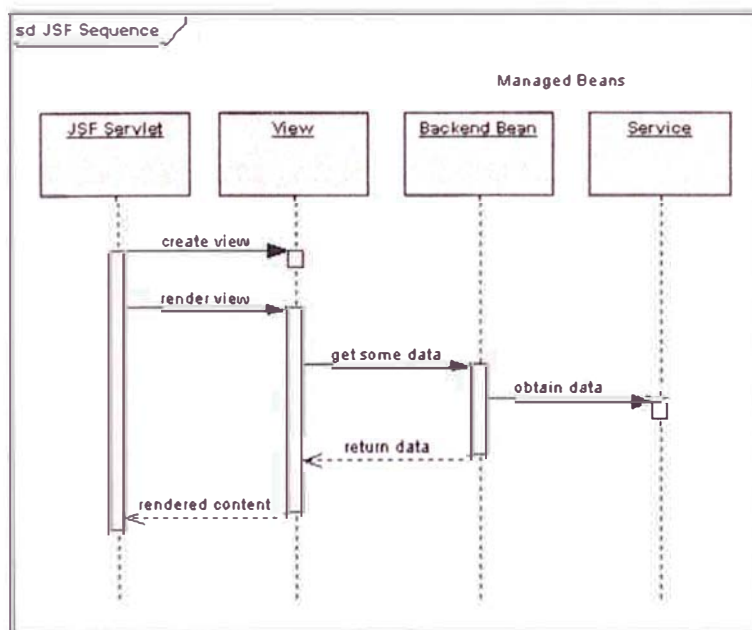


Figura 14: Diagrama de Interacción JSF

Cuando se produce un evento se notifica con una petición HTTP al servidor, donde existe un servlet, FacesServlet, que procesa la petición. El siguiente diagrama de secuencia muestra cómo gestiona JSF una petición.



Ventajas de utilizar JSF

Una de las principales ventajas de utilizar JSF es que ofrece una separación clara entre comportamiento y presentación. Una aplicación Web basada en JSF permite mapear peticiones HTTP a eventos y gestionar el estado de objetos.

Con JSF también nos vemos liberados de la tecnología que utilizamos en la presentación. Generalmente, la presentación en una aplicación web se realiza con JSPs, mientras que JSF, además de soportar esta tecnología permite utilizar otras tecnologías de presentación, como XML, de forma que la misma aplicación pueda visualizarse en distintos dispositivos cliente. Y lo principal, JSF proporciona una arquitectura que permite gestionar el estado de los componentes, procesar los datos de los componentes, validar la entrada del usuario y gestionar eventos.

b. Primefaces

Es una librería de componentes visuales open source desarrollada y mantenida por Prime Technology, una compañía Turca de IT especializada en consultoría ágil, JSF, Java EE y Outsourcing.

Las principales características de Primefaces son:

1. soporte nativo de Ajax, incluyendo Push/Comet.
2. kit para crear aplicaciones web para móviles.
3. es compatible con otras librerías de componentes, como [JBoss RichFaces](#).
4. uso de javascript no intrusivo (no aparece en línea dentro de los elementos, sino dentro de un bloque <script>).
5. es un proyecto open source, activo y bastante estable entre versiones.

Algunos inconvenientes podrían ser:

- para utilizar el soporte de Ajax tenemos que indicarlo explícitamente, por medio de atributos específicos de cada componente.
- no podemos utilizar el soporte de Ajax de JSF 2 (mediante <f:ajax>) con los componentes de Primefaces.

Este tutorial no pretende hacer un análisis exhaustivo de todos los componentes de Primefaces (más de 100 en el momento de escribir este tutorial), sino servir de introducción y mostrar lo necesario para poder utilizar la librería en nuestras aplicaciones web.

b.1 Dependencias

PrimeFaces se ejecuta en Java 5 + y una implementación JSF2.0 como obligatoria las dependencias. Hay algunas bibliotecas opcionales para ciertas funciones.

Dependencia	Versión	Tipo	Descripción
JSF runtime	2.0+ Required	Requerido	Apache MyFaces or Oracle Mojarra
ltext	2.1.7	Opcional	Apoyo a la exportación de PDF para DataExporter componente
apache poi	3.2- FINAL	Opcional	Excel para el apoyo a la exportación DataExporter componente
commons-fileupload	1.2.1	Opcional	Carga de Archivos
commons-io	1.4	Opcional	Carga de Archivos
atmosphere-runtime	0.5.1	Opcional	Ajax soporte
atmosphere-compatible	0.5.1	Opcional	Ajax soporte

b.2 Configuración

PrimeFaces no requiere ninguna configuración obligatoria.

b.3 Conjuntos de Componentes

Mostraremos la lista de los componentes hasta la versión 2.2 de primefaces

Nro.	Nombre Componente
1	3.1 AccordionPanel
2	3.2 AjaxBehavior
3	3.3 AjaxStatus
4	3.4 AutoComplete
5	3.5 BreadCrumb
6	3.6 Button
7	3.7 Calendar
8	3.8 Captcha
9	3.9 Carousel
10	3.10 CellEditor
11	3.11 Charts
12	3.12 Collector
13	3.13 Color Picker
14	3.14 Column
15	3.15 Columns
16	3.16 ColumnGroup
17	3.17 CommandButton
18	3.18 CommandLink
19	3.19 ConfirmDialog
20	3.20 ContextMenu
21	3.21 Dashboard
22	3.22 DataExporter
23	3.23 DataGrid
24	3.24 DataList

Nro	Nombre Componente
52	3.50 InputTextarea
53	3.51 Keyboard
54	3.52 Layout
55	3.53 LayoutUnit
56	3.54 LightBox
57	3.55 Media
58	3.56 Menu
59	3.57 Menubar
60	3.58 MenuButton
61	3.59 MenuItem
62	3.60 Message
63	3.61 Messages
64	3.62 NotificationBar
65	3.63 OutputPanel
66	3.64 Panel
67	3.65 Password
68	3.66 PickList
69	3.67 Poll
70	3.68 Printer
71	3.69 ProgressBar
72	3.70 Push
73	3.71 Rating
74	3.72 RemoteCommand
75	3.73 Resizable

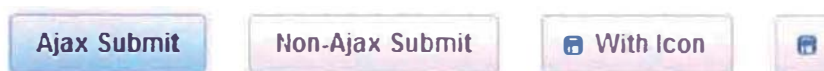
Nro.	Nombre Componente
25	3.25 DataTable
26	3.26 Dialog
27	3.27 Divider
28	3.28 Drag&Drop
29	3.28.1 Draggable
30	3.28.2 Droppable
31	3.29 Dock
32	3.30 Editor
33	3.31 Effect
34	3.32 Fieldset
35	3.33 FileDownload
36	3.34 FileUpload
37	3.35 Focus
38	3.36 Galleria
39	3.37 GMap
40	3.38 GMapInfoWindow
41	3.39 GraphicImage
42	3.40 GraphicText
43	3.41 Growl
44	3.42 HotKey
45	3.43 IdleMonitor
46	3.44 ImageCompare
47	3.45 ImageCropper
48	3.47 Inplace
49	3.46 ImageSwitch
50	3.48 InputMask
51	3.49 InputText

Nro	Nombre Componente
76	3.74 Resource
77	3.75 Resources
78	3.76 Row
79	3.77 RowEditor
80	3.78 RowExpansion
81	3.79 RowToggler
82	3.80 Schedule
83	3.81 Separator
84	3.82 Slider
85	3.83 Spacer
86	3.84 Spinner
87	3.85 Submenu
88	3.86 Stack
89	3.87 Tab
90	3.88 TabView
91	3.89 Terminal
92	3.90 ThemeSwitcher
93	3.91 Toolbar
94	3.92 ToolbarGroup
95	3.93 Tooltip
96	3.94 Tree
97	3.95 TreeNode
98	3.96 TreeTable
99	3.97 Watermark
100	3.98 IZard

A continuación detallaremos las funcionalidades de los principales componentes:

CommandButton

Uso de CommandButton es similar al estándar h:commandButton, por defecto commandButton presenta su forma envolvente con ajax, se puede muestra de diferente.



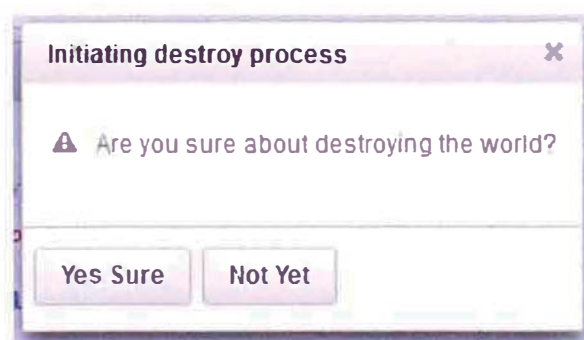
CommandLink

CommandLink se utiliza como el estándar h:commandLink, la diferencia es la forma es submitted con ajax por defecto. Muestra los siguientes controles

[Ajax Submit](#) [Non-Ajax Submit](#)

ConfirmDialog

ConfirmationDialog es una interfaz para confirmar la operación de una transacción..



InputText

InputText permite el registro de información para ser procesado.



Masked Input

Máscara de entrada componente fuerzas entrada ser formateado de una forma específica



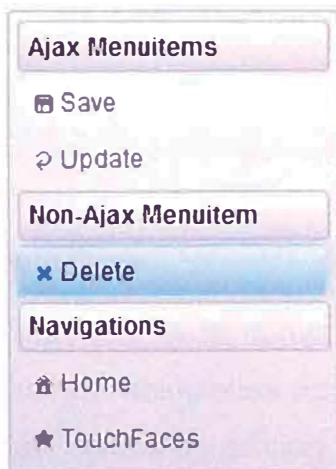
The screenshot shows a window titled "Masks" with several input fields, each with a specific mask applied to the text entered:

- Date: 12/03/2012
- Phone: (232) 323-2323
- Phone with Ext: (223) 232-3222 x2232
- taxid: 12-1212121
- SSN: 121-23-2323
- Product Key: as-121-s121

At the bottom of the form are two buttons: "Reset" and "Submit".

Menu

Es un componente de navegación es altamente personalizable que admite posicionamiento dinámico y estático.



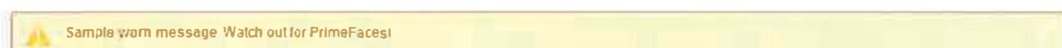
Messages

Mensajes, es un componente que permite gestionar mensajes son altamente personalizados, tienes 4 formas de presentación:

1. Informativos



2. Advertencia



3. Error



4. Error de sistema



Message

Default: * form2 default: Validation Error: Value is required.

Text Only: * form2 textOnly: Validation Error: Value is required

Icon Only: *

TreeTable

TreeTable se utiliza para presentar datos jerárquicos en un formato tabular.

Document Viewer		
Name	Size	Type
Documents		Folder
Pictures		Folder
barcelona.jpg	30 KB	JPEG Image
logo.jpg	45 KB	JPEG Image
optimusprime.png	96 KB	PNG Image
Movies		Folder

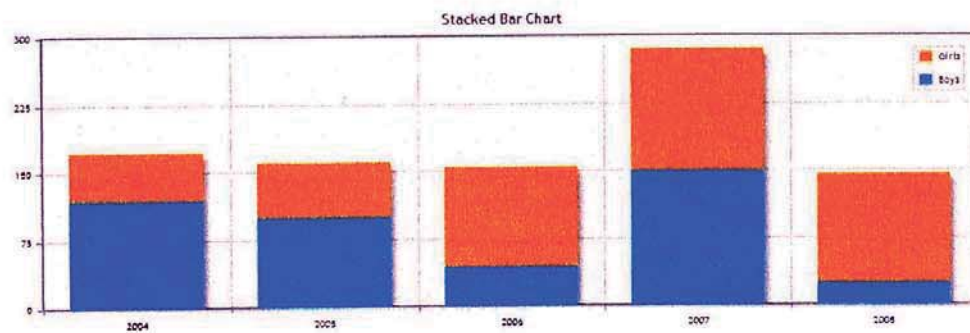
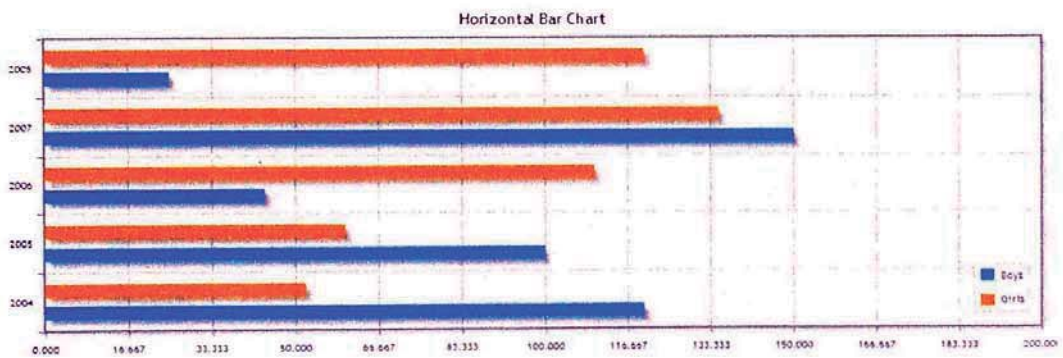
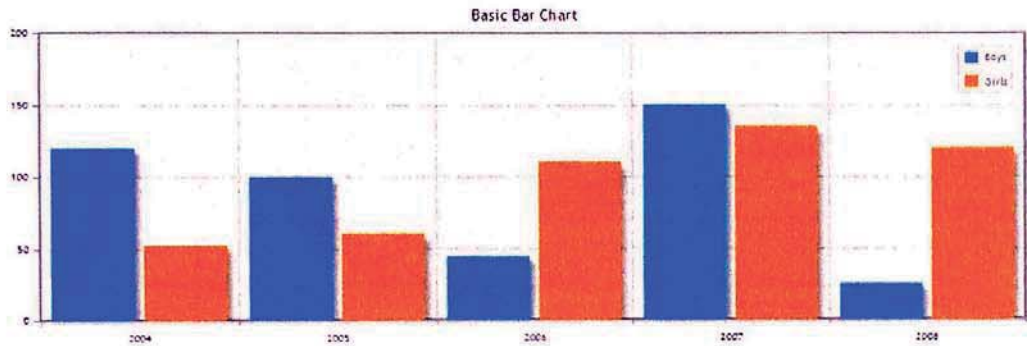
DataTable

DataTable es un componente de datos con paginación ajax, ordenar, filtrar, diversas soluciones de selección de fila, encabezados anidados, ampliables filas, celdas de edición, soporte de datos desplazable..

RadioButton Based Selection				
	Model	Year	Manufacturer	Color
<input type="radio"/>	226f8051	1971	Mercedes	Yellow
<input type="radio"/>	452b095d	1980	Mercedes	Black
<input type="radio"/>	b653b784	1971	Audi	White
<input type="radio"/>	4e6bd0aa3	1970	Mercedes	Red
<input type="radio"/>	fd95bea4	2006	Chrysler	Brown
<input type="radio"/>	54303150	1998	BMW	Black
<input type="radio"/>	5a7e49ca	1950	Volkswagen	Maroon
<input type="radio"/>	9e442884	1980	Ford	Red
<input type="radio"/>	1c2ae59c	2001	Mercedes	Silver
<input type="radio"/>	19066f0d	1987	BMW	Black

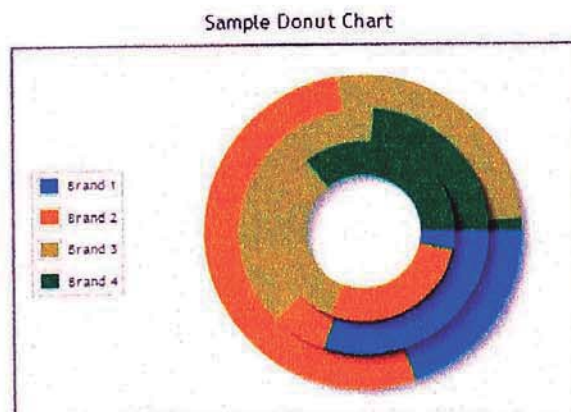
Charts - Bar

BarChart muestra la información en histograma.

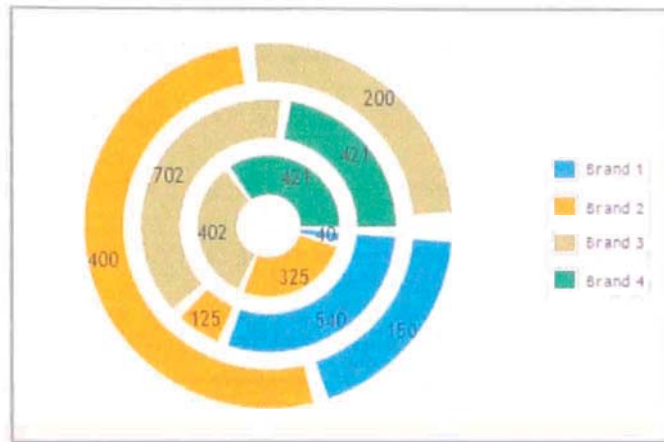


Donut Chart

DonutChart se crea con un DonutChartModel.



Custom Options



PickList

Lista de selección de un componente de entrada de lista doble con arrastre y basado reordenación..

Basic PickList

Istanbul
Ankara
Izmir
Antalya
Bursa

Submit

Available

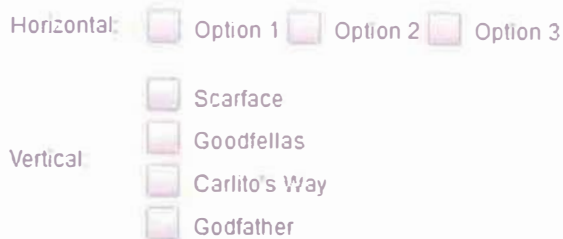
Messi
Iniesta
Villa
Alves

Starting

Submit

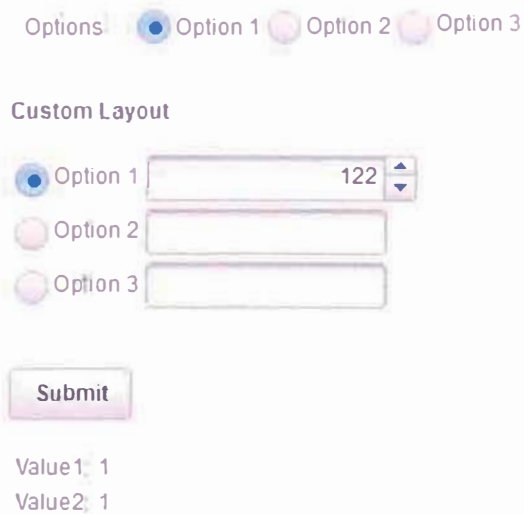
SelectManyCheckbox

SelectManyCheckbox se extiende SelectManyCheckbox estándar con capacidades de cambios en los estilos (revestimiento).



SelectOneRadio

SelectOneRadio extiende SelectOneRadio estándar con capacidades de revestimiento y característica de diseño personalizado.



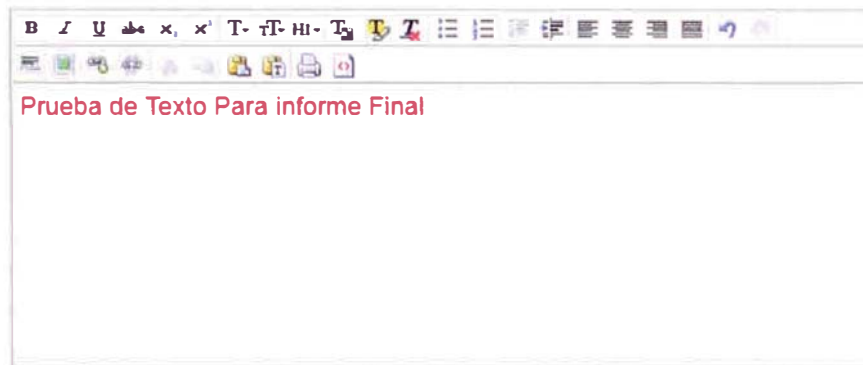
SelectOneListbox

SelectOneListbox se extiende SelectOneListbox estándar.



Editor

Editor es un componente de entrada con funciones de edición de texto enriquecido. Barra de herramientas puede configurarse con controles personalizados.



```
editor.xhtml | EditorBean.java
01. <h:form id="form">
02.
03.     <p:editor id="editor" value="#{editorBean.value}" width="600"/>
04.
05.     <h:panelGrid columns="2" style="margin-top:10px">
06.         <p:commandButton id="submitButton" value="Submit" update="display" oncomplete="dlg.show()"
07.             icon="ui-icon-disk" />
08.         <p:commandButton id="clearButton" type="button" value="Clear" onclick="editor.clear()"
09.             icon="ui-icon-close" />
10.     </h:panelGrid>
11.
12.     <p:dialog header="Content" widgetVar="dlg" showEffect="fade" hideEffect="fade" modal="true">
13.         <h:outputText id="display" value="#{editorBean.value}" cscapt="false" />
14.     </p:dialog>
15.
16. </h:form>
17.
```

```
editor.xhtml | EditorBean.java
01. package org.primefaces.examples.view;
02.
03. public class EditorBean {
04.
05.     private String value;
06.
07.     public String getvalue() {
08.         return value;
09.     }
10.
11.     public void setvalue(String value) {
12.         this.value = value;
13.     }
14.
15. }
```

b.4 Representación parcial y Procesamiento

PrimeFaces proporciona una representación parcial y la función de procesamiento de la vista basado en la norma API JSF 2 para permitir la elección de lo que debe procesar en el Ciclo de Vida de JSF y lo que va a representar en el final con el Ajax.

Representación parcial

Además de los componentes como Autocompletar, tabla de datos, slider con capacidades integradas de ajax, PrimeFaces también proporciona un PPR genérico (de representación de páginas parcial) mecanismo para actualizar componentes JSF con el Ajax. Varios componentes están dotados de los atributos comunes (por ejemplo, actualización, el proceso, OnStart, onComplete).

- Infraestructura

PrimeFaces Ajax Framework se basa en la API estándar del lado del servidor de JSF 2. No hay adicional

Artifacts como medida AjaxViewRoot, AjaxStateManager, AjaxViewHandler, filtros de servlets,

HtmlParsers, los PhaseListeners y así sucesivamente. PrimeFaces pretende mantenerlo limpio, rápido y ligero.

En el lado del cliente en lugar de utilizar implementaciones de la API del cliente secundario de las implementaciones de JSF, como Mojarra y MyFaces, PrimeFaces se basan en la biblioteca de javascript más popular; jQuery que es mucho más probada y estable en relación con el Ajax, el manejo libertad de desplazamiento de un árbol de JSF implementado con la secuencias de comandos.

- Uso de identificadores

Primeros pasos

Cuando se utiliza PPR es necesario especificar qué componente (s) para actualizar con ajax. Si el componente que desencadena la solicitud PPR está en el mismo NamingContainer (forma por ejemplo) con el componente (s) que

Hace la solicitud, puede utilizar los identificadores de servidor directamente. En esta sección, aunque vamos a estar utilizando CommandButton,

Lo mismo se aplica a cada componente que es capaz de generar la misma respuesta como `commandLink`, `poll`, `remoteCommand` y etc.

```
<h:form>
<p:commandButton update="display" />
<h:outputText id="display" value="#{bean.value}"/>
</h:form>
```

PrependId

Configuración de ajuste `prependId` de una forma no tiene ningún efecto sobre la forma PPR se utiliza.

```
<h:form prependId="false">
<p:commandButton update="display" />
<h:outputText id="display" value="#{bean.value}"/>
</h:form>
```

ClientId

También es posible definir el número de cliente del componente para actualizar

```
<h:form id="myform">
<p:commandButton update="myform:display" />
<h:outputText id="display" value="#{bean.value}"/>
</h:form>
```

Distintos contenedores de nomenclatura (NamingContainer)

Si su página tiene diferentes contenedores de nomenclatura (por ejemplo, dos formularios), también hay que añadir el ID del contenedor para buscar la expresión de PPR, para que pueda manejar las peticiones que se desencadenan dentro de un `NamingContainer` que actualiza otra `NamingContainer`. Lo que sigue es el camino sugirió el uso de caracteres como un separador de prefijo, tenga en cuenta que este utiliza el algoritmo de búsqueda mismo estándar de JSF implementación de la 2;

```

<h:form id="form1">
  <p:commandButton update=":form2:display" />
</h:form>
<h:form id="form2">
  <h:outputText id="display" value="#{bean.value}"/>
</h:form>

```

Usando clientIds absolutos y también funciona como una extensión de PrimeFaces sin embargo, podría eliminar en un futura versión para alinearse con las especificaciones de JSF.

```

<h:form id="form1">
  <p:commandButton update="form2:display" />
</h:form>
<h:form id="form2">
  <h:outputText id="display" value="#{bean.value}"/>
</h:form>

```

Múltiples componentes

Varios componentes para actualizar se puede especificar con la provisión de una lista de identificadores separados por una coma, espacios en blanco o incluso ambos.

Coma

```

<h:form>
  <p:commandButton update="display1,display2" />
  <h:outputText id="display1" value="#{bean.value1}"/>
  <h:outputText id="display2" value="#{bean.value2}"/>
</h:form>

```

Espacio Blanco

```

<h:form>
  <p:commandButton update="display1 display2" />
  <h:outputText id="display1" value="#{bean.value1}"/>
  <h:outputText id="display2" value="#{bean.value2}"/>
</h:form>

```

- Procesamiento parcial

En representación de páginas parciales, sólo los componentes especificados se representan, de manera similar en el procesamiento de los componentes definidos parcial sólo se procesan. De elaboración se entiende la ejecución de Aplicar Valores de Respuesta, validaciones de procesos, modelos y Actualizar invocar la aplicación del ciclo de vida de JSF fases sólo en los componentes definidos.

Esta característica es una simple pero lo suficientemente potente para hacer validaciones de grupo, evitando la validación de los componentes no deseados, eliminando la necesidad de utilizar inmediatos y muchos más casos de uso. Diversos componentes, como CommandButton, commandLink están equipadas con atributo de proceso, en los ejemplos que vamos a utilizar CommandButton.

- La validación parcial

Un caso de uso común de proceso parcial está haciendo validaciones parciales, suponga que tiene un simple formulario de contacto con los dos componentes desplegables para seleccionar la ciudad(componte selectOneMenu con id= "cities") y el suburbio(componte selectOneMenu con id= "suburbs"), también hay una inputText que se requiere. Cuando se selecciona la ciudad, los suburbios de la ciudad relacionadas con el seleccionado se rellena en el suburbio desplegable.

```
<h:form>
<h:selectOneMenu id="cities" value="#{bean.city}">
<f:selectItems value="#{bean.cityChoices}" />
<p:ajax listener="#{bean.populateSuburbs}" update="suburbs"
process="@all"/>
</h:selectOneMenu>
<h:selectOneMenu id="suburbs" value="#{bean.suburb}">
<f:selectItems value="#{bean.suburbChoices}" />
</h:selectOneMenu>
<h:inputText value="#{bean.email}" required="true"/>
```

Cuando la ciudad desplegable se cambia una solicitud de ajax se envía a ejecutar populateSuburbs método que rellena suburbChoices y, finalmente, actualizar los suburbios desplegable. El problema es que populateSuburbs método no se ejecuta como ciclo de vida se detendrá después de proceso de fase de validación para saltar la respuesta de render como entrada de correo electrónico no está prevista. La razón es p:Ajax tiene @all como el valor indicando para procesar cada componente en la página pero no hay necesidad de procesar la inputText.

La solución es definir lo que debe procesar en p: ajax. Como sólo estamos haciendo una solicitud de cambio de la ciudad, sólo proceso que debe pasar es ciudades desplegable.

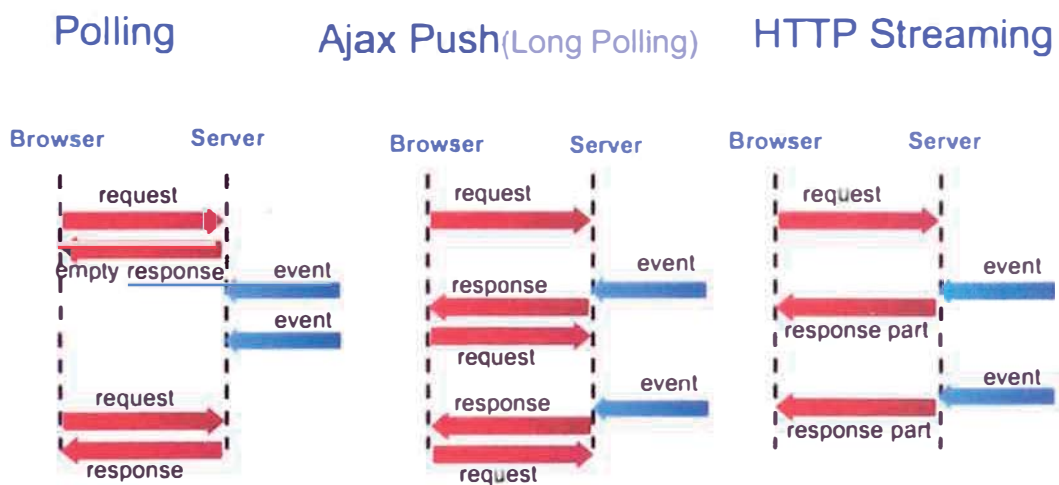
```
<h:form>
<h:selectOneMenu id="cities" value="#{bean.city}">
<f:selectItems value="#{bean.cityChoices}" />
<p:ajax actionListener="#{bean.populateSuburbs}"
event="change" update="suburbs" process="@this"/>
</h:selectOneMenu>
<h:selectOneMenu id="suburbs" value="#{bean.suburb}">
<f:selectItems value="#{bean.suburbChoices}" />
</h:selectOneMenu>
<h:inputText value="#{bean.email}" required="true"/>
</h:form>
```

Es decir, ahora el método populateSuburbs será llamado y la lista de los suburbios se rellena. Nótese que valor por defecto para la opción de proceso es @ this ya para p: Ajax como se indica en AjaxBehavior documentación, que se define explícitamente aquí para darle una mejor comprensión de cómo el procesamiento parcial obras.

- Ajax Push/Comet

Comet es un modelo que permite un servidor web para enviar datos a los navegadores. Subastas y el chat son bien conocidos los casos de ejemplo de uso de la técnica del cometa. El cometa se puede implementar en un par de maneras.

Lo que sigue es un esquema que describe estas técnicas.



6. Themes/Temas

PrimeFaces está integrado con un potente marco ThemeRoller CSS. Actualmente hay 30 prediseñada, temas que se pueden ver y descargar desde la galería tema PrimeFaces.

<http://www.primefaces.org/themes.html>

- Aplicación de un tema

Aplicación de un tema a su proyecto PrimeFaces es muy fácil. Cada tema se empaqueta como un archivo jar, descargar el tema que desea utilizar, añadir a la ruta de clases de su solicitud y luego definir primefaces.THEME parámetro de contexto en su descriptor de despliegue (web.xml) con el tema nombre como el valor.

```
<context-param>  
  <param-name>primefaces.THEME</param-name>  
  <param-value>aristo</param-value>  
</context-param>
```



c. Utilities

c.1 RequestContext

RequestContext es una sencilla utilidad que ofrece soluciones útiles, tales como la adición de parámetros a Ajaxcallback funciones.

RequestContext puede obtenerse de manera similar a FacesContext.

```
RequestContext requestContext = RequestContext.getCurrentInstance();
```

c.2 RequestContext API

Método	Descripción de Método
isAjaxRequest()	Devuelve un valor booleano si la solicitud actual es una PrimeFaces petición AJAX.
addCallbackParam (String name, Object value)	Añade los parámetros a devoluciones de llamada de AJAX como onComplete.
addPartialUpdateTarget(String target);	Especifica componente (s) para actualizar en tiempo de ejecución.

d. La integración con Java EE

PrimeFaces es todo acerca de front-end y puede ser respaldada por la aplicación empresarial favorita marco. Siguiendo marcos son totalmente compatibles;

- Spring Core (JSF centric JSF-Spring Integrations)
- Primavera WebFlow (Spring Centric JSP-Spring Integrations)
- Los EJBs

d.1 CDI and EJBs

PrimeFaces es totalmente compatible con un entorno de Java EE 6 con el CDI y EJB.

d.2 Spring WebFlow


Nosotros, como el trabajo en equipo PrimeFaces estrechamente con el equipo Primavera WebFlow, PrimeFaces es sugerido por SpringSource,

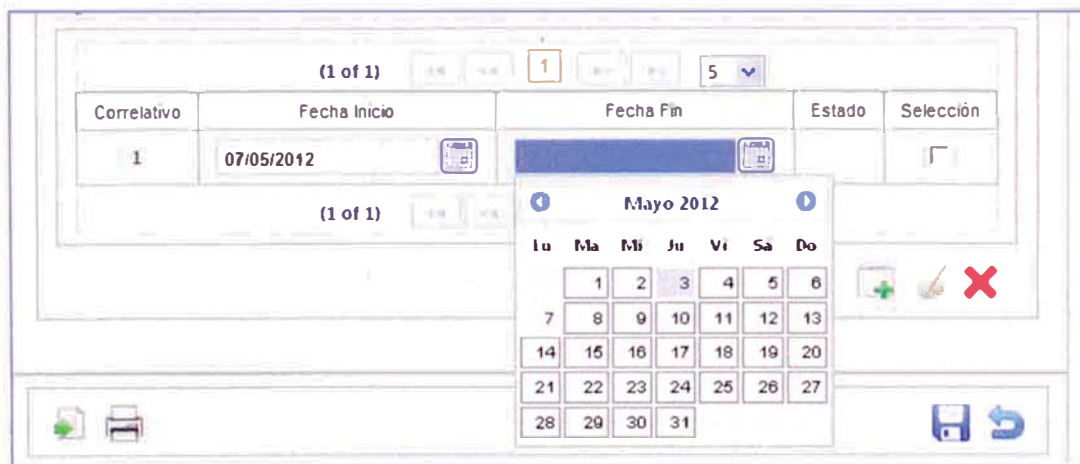
como la suite de componentes JSF preferida para aplicaciones SWF. SpringSource repositorio cuenta con dos muestras basadas en SWF PrimeFaces; un pequeño escaparate y el ejemplo de reserva de las caras.

e. Interfaces y Pantallas Obtenidas

A continuación mostraremos los prototipos de pantallas que se obtenidas a partir de los componentes primefaces.

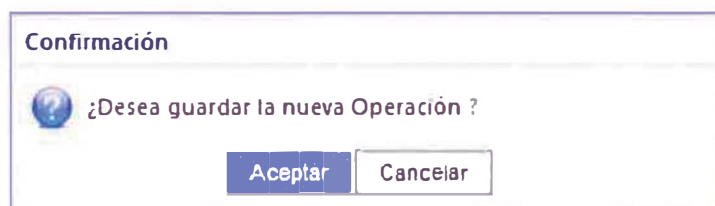
Pantalla: Ingreso de fechas de vencimientos, edición y eliminación

Permite ingresadas las fechas de vigencias en cada fila, si se da clic en el icono del calendario el sistema responde mostrando el calendario a la fecha actual, El usuario da clic en el  el sistema responde mostrando el mensaje de confirmación.



Pantalla: Interfaz de confirmación de transacción

El as operaciones luego de las validaciones correspondientes el sistema responde mostrando la pantalla de confirmación.



Si el usuario da clic en el botón aceptar, el sistema responde mostrando transacción exitosa.

Pantalla: Administración

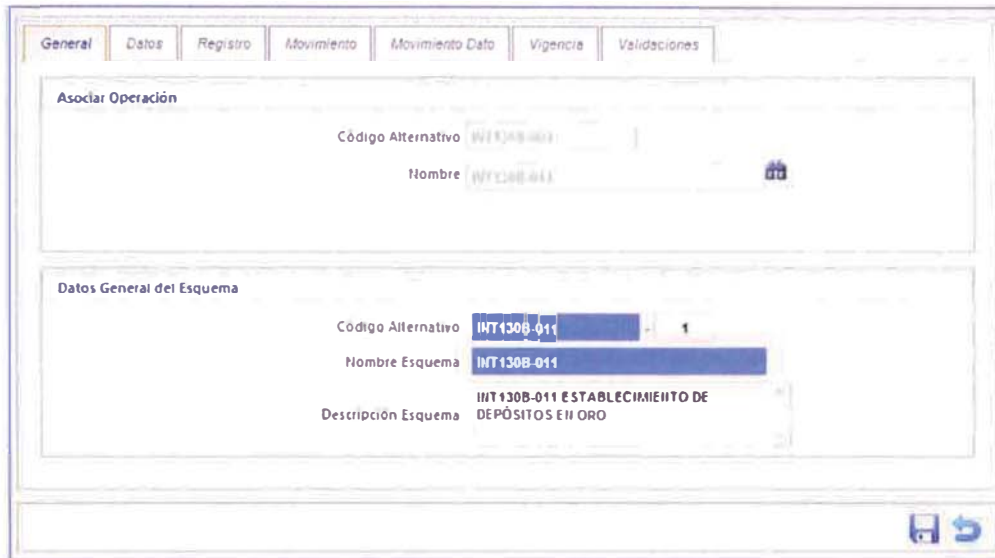
La pantalla permite gestionar desde la creación, edición, consulta y eliminación.

Código Alternativo	Descripción	Fecha Vigencia		Selección
		Fecha Inicio	Fecha Fin	
FIN330E-001-1	FIN330E-001 prueba - MIHJ	12/06/2012		<input type="checkbox"/>
INT130A-001-1	Esquema INT130A-001	19/06/2012		<input type="checkbox"/>
INT130A-002-1	Esquema INT130A-002	07/06/2012		<input type="checkbox"/>
INT130A-004-1	Esquema INT130A-004	07/06/2012		<input type="checkbox"/>
INT130B-0014-1	INT130B-0014 TRANSFERENCIA DE DEPÓSITOS EN ORO	08/06/2012		<input type="checkbox"/>

Pantalla: Administración utilizando tab.

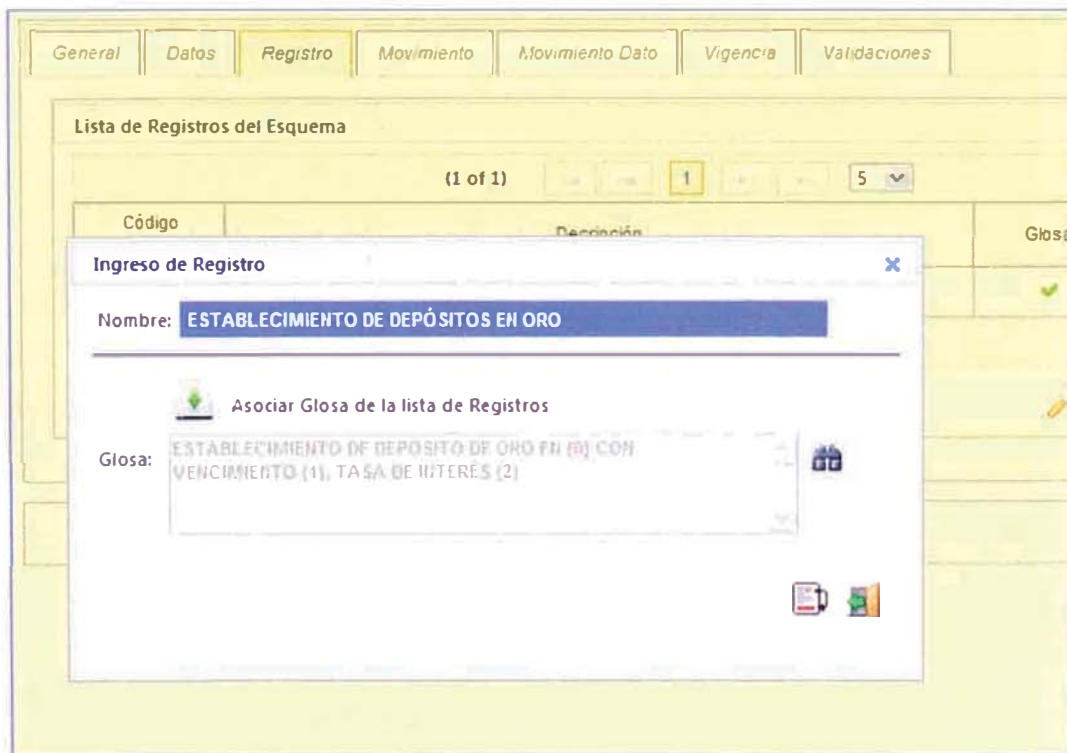
Si el usuario da clic en el icono  de edición, el sistema responde mostrando la pantalla.

Don se muestra los tab, cada uno esta amarrando una transacción, cuando se completa todos los trabs, el sistema está listo para completarse la transacción caso contrario el sistema validaría en ingreso de los datos en los tablas correspondientes.



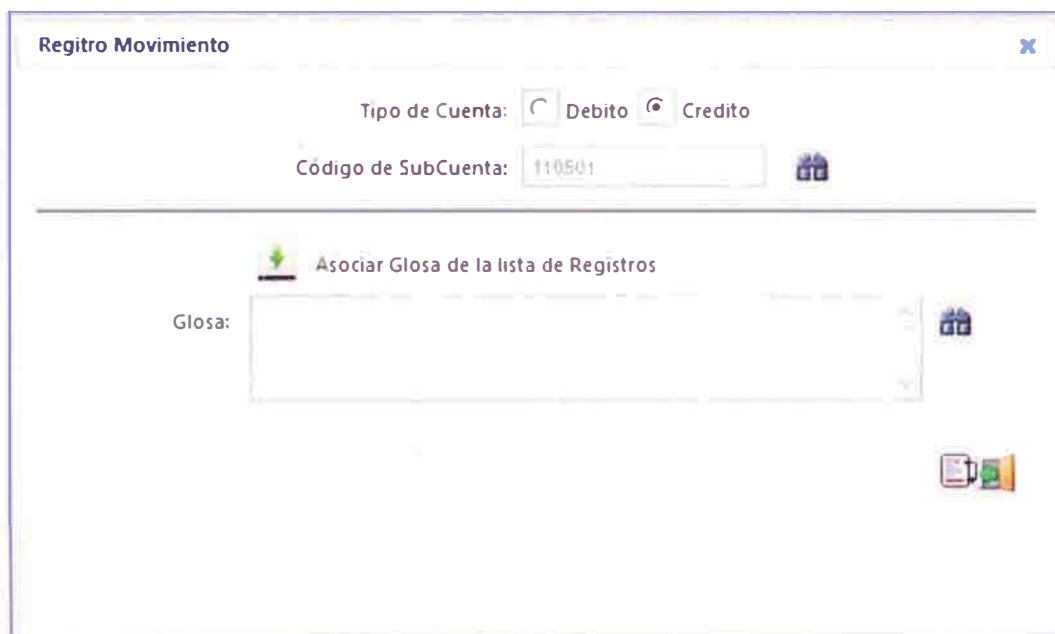
Pantalla: Interfaz emergente

Esta pantalla es un ejemplo de una emergente, permite una rápida interacción con el usuario evitando la navegabilidad mostrando comportamiento a un aplicativo de escritorio.



Pantalla: Carga elementos en la pantalla llamando a los popups que están enlazadas por el icono y adicionar los elementos a la lista.

Esta ventana es un popup proveniente de la interfaz anterior, donde se muestra particularidades comunes como carga de elementos desde otro lista por la interacción ajax entre las interfaces a nivel del cliente, en cada icono invoca a otro popup ejecutándose de manera ligera las operaciones, en estos caso las transacciones completadas por los componentes primefaces utilizan el ciclo completo de las actualizaciones los componen estés.



Pantalla: Utilizando componentes dinámicos de datos.



Cuando se da clic en el botón adicionar se actualiza el detalle de la tabla y las descripción.

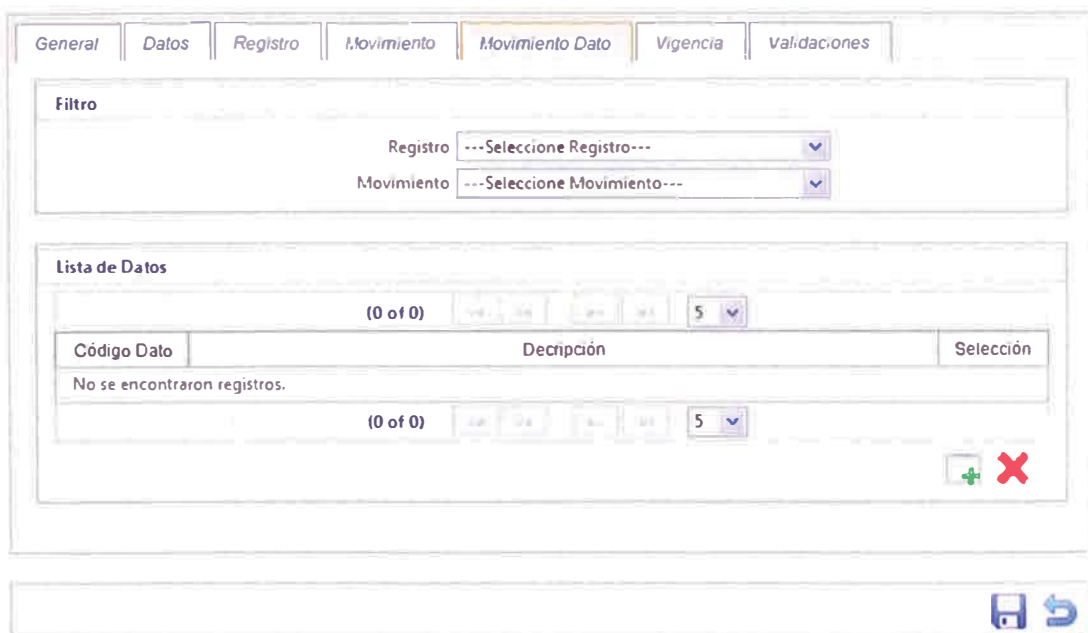


Pantalla: Selección de dinámica de filtros

La pantalla permite una selección dinámica de elementos,

Si el usuario selecciona el registro, muestra la colección de movimientos asociadas a cada registro.

Si el usuario selecciona el movimiento, el sistema responde mostrando la pantalla con a lista de datos asociada al movimiento, caso contrario se permitirá asociar o disociar con los botones "Agregar" , "Elimina" .



CAPÍTULO IV

ANÁLISIS COSTO BENEFICIO

Evaluaremos los beneficios obtenidos versus el costo de la inversión.

Medir el beneficio es complicado ya que queda mezclado con varios factores pero trataremos de estimar un valor, en base a logros concretos, para aquellos que son tangibles. El costo se evalúa en base a la inversión de Recursos y tiempo invertido por la organización.

4.1 COSTO DE LA IMPLEMENTACIÓN.

Las estimaciones se basaran en horas hombre por el tipo de recursos tomando como **tope la puesta en marcha de sistema con la cantidad esperada de usuarios (100)**

a). COSTO DE LA SOLUCIÓN

Proyectado para 1 año

Tabla 10: Estimación de costo de la nueva aplicación

Recurso	Actividad	Cantidad	Días	Horas/Día	Costo/Hora	Consto/Recurso
Analista Programador	I	3	360	8	30	S/. 259,200.00
Arquitecto	I	1	100	4	50	S/. 20,000.00
Jefe de Proyecto	I	1	360	8	50	S/. 144,000.00
Analista Funcional	I	2	240	8	40	S/. 153,600.00
Documentador	I	1	360	8	10	S/. 28,800.00

Monto Total del Proyecto	S/. 605,600.00
--------------------------	----------------

4.2 BENEFICIOS

Considerando el rediseño técnico planteado a partir de la solución final del proyecto.

4.2.1 BENEFICIOS TANGIBLES

A continuación se muestra el cuadro resumen de los valores de los Beneficios tangibles.

- El Sistema es más estable, ya no se permite realizar ventas a clientes morosos, clientes clonados que debían a la empresa, la conexión con las Centrales Crediticia, lo que hace es que las ventas no sean anuladas posteriormente. **Antes era de 25% ahora 3% del total de las ventas.**

Las venta antes del cierre era 120, 000,000 y al final del cierre real las ventas llegaban a 90, 000,000 dado que la diferencia era producto de ventas ficticias. Como los vendedores vendieron a gente que debía a la empresa, gente que se clonaban para ser aparentemente otro cliente, en el fondo era el mismo cliente producto de que el sistema podía registrar ventas ficticias.

Se muestra la tabla de ahorros generados por la reducción de ventas erróneas.

Estimación para un contrato con venta 120, 000,000 antes del cierre

Tabla 11: Consumo de combustible por un periodo de un año de vendedores.

Áreas de Ventas	Numero de Vendedores	Costo Combustible Por Día	Consumo Por Día	Días Útiles	Total
Gerencia Norte	64	40	S/. 2,560	260	S/. 665,600.00
Gerencia Sur	64	40	S/. 2,560	260	S/. 665,600.00
Tele ventas y provincia	24	40	S/. 960	260	S/. 249,600.00
Consumo Total Anual de combustible					S/. 580,800.00

Analizando la reducción de los contratos validos de 25% a 3% para un Año.

Tabla 12: Consumo de combustible.

Consumo de combustible	Porcentaje	Venta erradas
Ventas Netas antes del cierre	100%	S/. 1,580,800.00
Ventas Herradas antes de la implementación	25%	S/. 395,200.00
Ventas Herradas después de la	3%	S/. 47,424.00

implementación		
Ahorro anual en combustible	22%	SI. 347,776.00

Con el desarrollo del sistema de información genera un ahorro de SI. 347,776.00 Nuevos soles anuales por el consumo de combustible.

4.2.2 BENEFICIOS NO TANGIBLES

La siguiente matriz muestra los beneficios asignadas a las unidades operativas del área de comercialización.

Tabla 13: Matriz de beneficios no tangible

Nro	Beneficios	Jefe de Ventas	Vendedores	Asesores de Venta
1	Monitoreo de las ventas desde cualquier lugar de internet.	X		
2	Registro de contratos y proformas desde cualquier lugar de internet.		X	X
3	Ahorro de tiempo en las aprobaciones de los contratos, pasaron de ser manual a automática ahorrando tiempos de 15 a 6 días.	X		
4	Incremento en los niveles de confianza y de aceptación de la implantación.	X	X	X
5	Mejora en la calidad visual y la interacción de la aplicación con el usuario final.	X	X	X
6	Reducción en los tiempos en la generación de contratos y proformas de ventas de anuncios.		X	
7	Reducción en los costos operativos y cobros a los clientes morosos	X	X	X

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES;

1. El nuevo sistema implementado con las mismas condiciones planteadas se redujo significativamente los tiempos de respuestas en las transacciones en las áreas de Facturación y Cobranza.

Tabla 14: Estimación de tiempo de respuesta de la nueva aplicación

Interface	Tiempo Promedio de	
	Transacción (Segundos)	Operación (minutos)
Propuesta Comercial	4	7
Primer Ingreso(Contrato)	3	5
Bocetos	2	4

2. Se redujo los procesos repetitivos y también los costos de soporte y administración de las aplicaciones.

Factorización de las aplicaciones:



Figura 18: Diagrama de Integración de Aplicación

3. Se redujeron significativamente los tiempos de codificación y de integración con las demás interfaces planteadas en la solución inicial **(Propuestas, Primer Ingreso, Boceto y otros interfaces de soporte a la aplicación)**.
4. Se consiguió un alto nivel de aprobación de la calidad y de los tiempos de respuestas de las interfaces de los módulos de facturación y cobranza.
5. Se consiguió una alta satisfacción del usuario por versatilidad y flexibilidad del sistema.

RECOMENDACIONES;

De la experiencia en la implementación de soluciones comerciales de alto nivel de transacciones, podemos llegar a las recomendaciones siguientes:

1. Realizar un detallado levantamiento de información, de esta manera determinar con mayor aproximación los tiempos en la ejecución de las soluciones.
2. Cuantificar los niveles de transacciones y de las trazabilidad del sistema
3. Definir el alcance técnico de la solución, de esta manera se puede determinar la participación de un especialista para definir la arquitectura(Arquitecto) para el caso

BIBLIOGRAFÍA

1. User's Guide PrimeFaces, tomado del url:
<http://www.primefaces.org>

2. Tutorial Spring, tomado del url:
<http://www.Springframework.org>

3. Introducción a JavaServer
<http://netbeans.org/kb/docs/web/jsf20-intro.html>

4. Tutoriales Struts
<http://struts.apache.org/2.x/docs/tutorials.html>

GLOSARIO.

Boceto, Es el diseño del anuncio plantilla, puede ser grafico o textual siguiendo las tipográficas de acuerdo al producto de Publicación y detalle del anuncio.

Firmante, Persona Natural o jurídica que representa legalmente a una empresa.

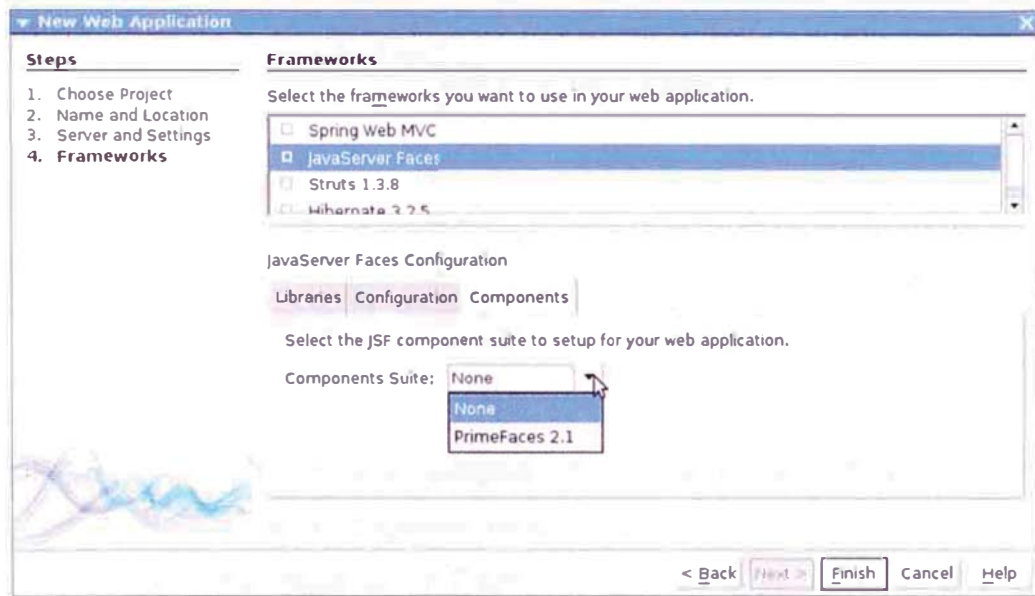
Modularidad, Es la capacidad que tiene un sistema de ser estudiado, visto o entendido como la unión de varias partes que interactúan entre sí y que trabajan para alcanzar un objetivo común, realizando cada una de ellas una tarea necesaria para la consecución de dicho objetivo. Cada una de esas partes en que se encuentre dividido el sistema recibe el nombre de módulo.

ANEXOS

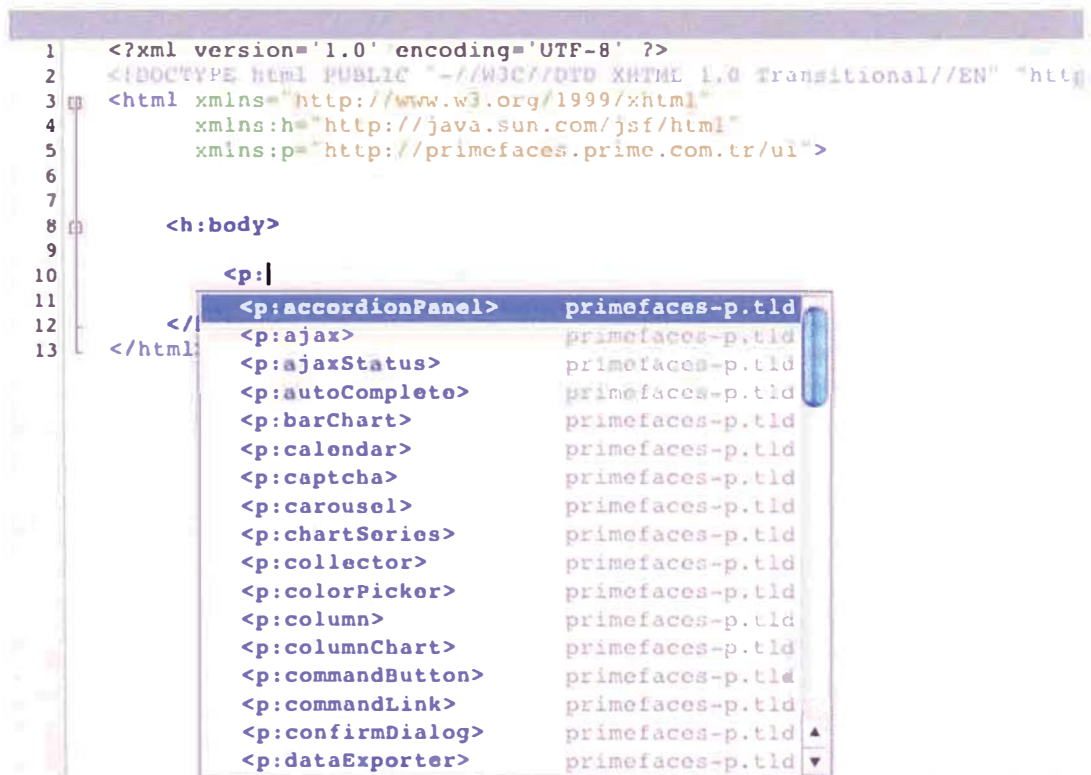
Compatibilidad con IDE

1 NetBeans

NetBeans 7.0 + PrimeFaces paquetes, al crear un nuevo proyecto que puede seleccionar a partir PrimeFaces ficha componentes;




Finalización de código con el apoyo de NetBeans 6.9 +;



```

html
1  <?xml version='1.0' encoding='UTF-8' ?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "h
3  <html xmlns="http://www.w3.org/1999/xhtml"
4      xmlns:h="http://java.sun.com/jsf/html"
5      xmlns:p="http://primefaces.prime.com.tr/ui">
6
7
8      <h:body>
9
10         <p:accordionPanel |
11
12     </h:body>
13 </html>

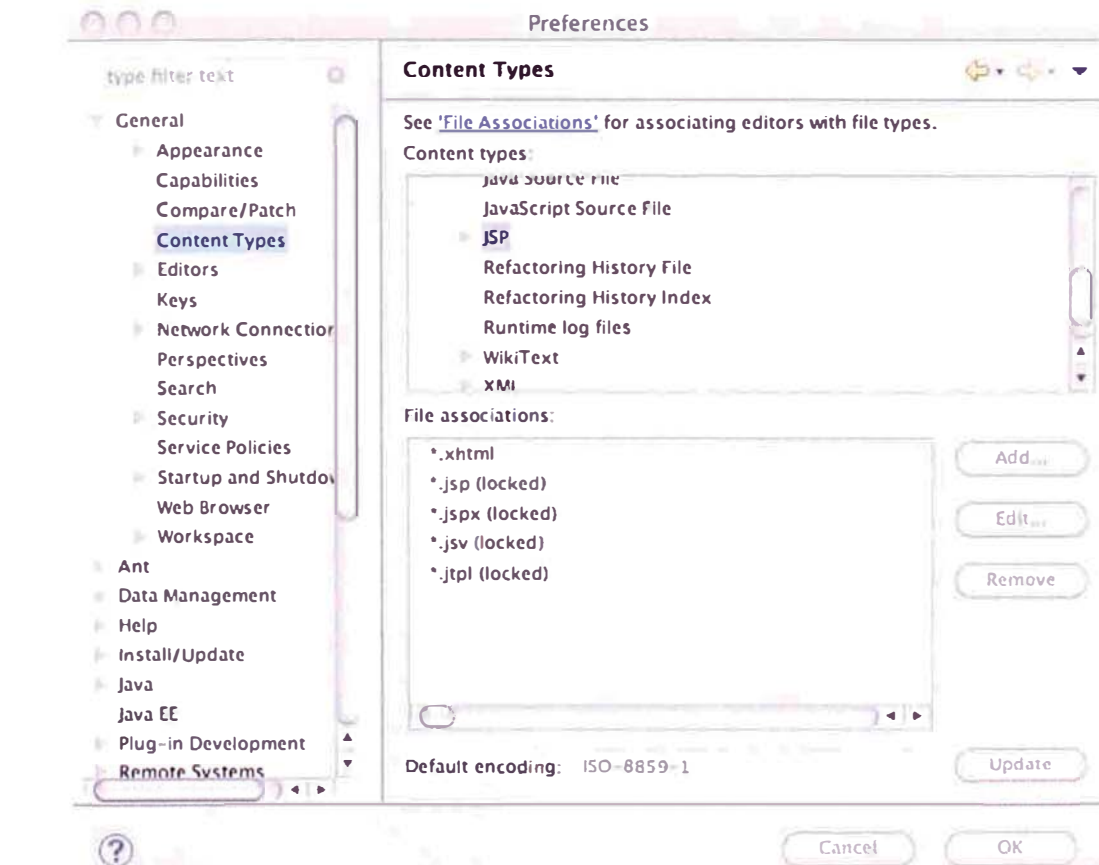
```



2 Eclipse

La implementación funciona en Eclipse Helios, cuando la faceta JSF está activada, Eclipse mayores requiere de un pequeño cambio para habilitar la compatibilidad con la finalización Facelets.

Open *Preferences* -> *General* -> *Content Types* -> *Text* -> *JSP* and add *.xhtml extension to the list.



Con este ajuste, los componentes PrimeFaces puede obtener la etiqueta /finalización atributo cuando se abrió con jsp editor de las versiones de Eclipse antes de Helios.

