

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

**“SISTEMA DE GENERACIÓN Y SEGUIMIENTO DE REPARACIÓN DE
CONTENEDORES USANDO AJAX”**

TESINA DE SEMINARIO

Previa a la obtención del Título de:

INGENIERO EN COMPUTACIÓN

ESPECIALIZACIÓN SISTEMAS DE INFORMACIÓN

Presentada por:

OSCAR STEVE RAMIREZ VILLON

GABRIEL ANTONIO SANCHEZ SAN ANDRES

Guayaquil - Ecuador

2010

AGRADECIMIENTOS

A Dios por estar conmigo en cada paso que doy, y permitirme culminar esta importante etapa de mi vida. A mi familia por todo su amor y apoyo incondicional que me han brindado siempre.

Gabriel Sánchez San Andrés

Agradezco a Dios, por haberme dado las fuerzas necesarias para superarme y ser mi guía en los momentos difíciles. A mis Padres por su apoyo incondicional. A mis hermanos por estar conmigo siempre.

Oscar Ramírez Villón

DEDICATORIA

A mi MADRE por su constante ayuda y cuidados. A mi PADRE por todo su apoyo y lo que me ha enseñado durante su vida. A mi HERMANA por estar conmigo siempre. A mis ABUELOS por su cariño, apoyo y comprensión.

Gabriel Sánchez San Andrés

A MIS PADRES que han llenado mi vida de virtudes, y me han enseñado a enfrentarme a los retos y obstáculos, a no desmayar y aprender a levantarme cuando he tropezado. A mis HERMANOS por su cariño sincero.

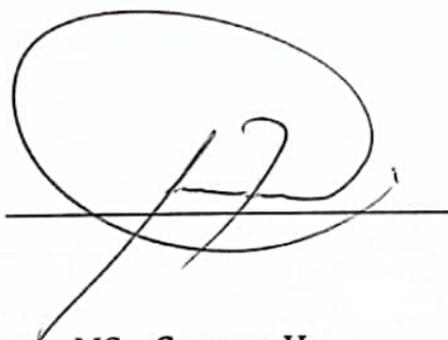
Oscar Ramírez Villón

TRIBUNAL DE SUSTENTACIÓN

A handwritten signature in black ink, appearing to read 'Carlos Martín', written over a horizontal line.

MSc. Carlos Martín

PROFESOR DEL SEMINARIO DE GRADUACIÓN

A handwritten signature in black ink, appearing to read 'Carmen Vaca', written over a horizontal line.

MSc. Carmen Vaca

PROFESOR DELEGADO DEL DECANO

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Trabajo de Graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la **Escuela Superior Politécnica del Litoral**”.

(Reglamento de exámenes y títulos profesionales de la ESPOL).



Oscar Steve Ramírez Villón



Gabriel Antonio Sánchez San Andrés

RESUMEN

En este proyecto de Graduación se ha elaborado una aplicación web empleando tecnología AJAX, para nuestro cliente Conami S.A. empresa que brinda servicios de auditoría, logística y operación portuaria y que tiene como uno de sus puntos fuertes el Mantenimiento y Reparación de contenedores. A su vez se tiene un registro con información básica tanto de los clientes, sus contenedores y finalmente las reparaciones que se le realicen a los mismos.

Con nuestra aplicación resolveremos un problema que se está presentando en el proceso administrativo de dicha empresa el cual es “La Elaboración de Estimativos.”

¿Qué es un Estimativo?

Las empresas que se dedican a asuntos portuarios en este caso reparación de contenedores necesitan generar un documento donde se especifica en detalle las reparaciones que se hicieron a cada contenedor que ha ingresado a sus talleres y además información muy valiosa para la empresa como por ejemplo datos del cliente, datos del contenedor, datos de cada reparación, fecha de elaboración del estimativo, total del costo del estimativo basado en los datos de costos de cada reparación y cálculos adicionales que son reglas del negocio establecidas por la

empresa, dicho documento ha sido llamado "ESTIMATIVO" , la generación de estimativos es una actividad diaria para este tipo de empresas pero actualmente el realizar dicho documento en hojas de Excel toma tiempo que puede ser reducido para invertirlo en otras tareas importantes ; de aquí nace la necesidad de automatizar este proceso con nuestro sistema de información.

El objetivo principal de nuestro sistema es disminuir el tiempo de elaboración de un estimativo, mantener la información almacenada y segura en una base de datos y generar reportes para llevar un seguimiento de los estimativos los cuales reflejan la actividad diaria de la empresa.

Este documento permitirá seguir la evolución de la construcción del sistema, así como nos ayudará a familiarizarnos con distintas herramientas usadas en el desarrollo de aplicaciones web como Hibernate, JQuery y la librería YUI. La aplicación está orientada al modelo MVC (Modelo Vista Controlador), ya que facilita el manejo y el mantenimiento de la misma.

El documento se ha dividido en cuatro secciones de la siguiente manera.

En el Capítulo 1 se introduce el concepto de Ajax como técnicas de desarrollo web para creación de aplicaciones interactivas y es el tema principal de nuestro Seminario de Graduación.

En el Capítulo 2 se podrá encontrar la información resumida de la empresa Conami S.A. a quien le elaboramos el sistema, además se describen las generalidades del proyecto, y a su vez los objetivos planteados en el presente trabajo.

En el Capítulo 3 se describen las reglas del negocio al cual está orientado la aplicación a desarrollar, además los modelos y las librerías utilizadas en este proyecto.

En el Capítulo 4 se detalla la implementación del sistema, arquitectura, mapeo de hardware y software junto con la descripción de los módulos implementados.

Para finalizar el informe, se señalan conclusiones y recomendaciones del proyecto donde se señala los beneficios que generó su implementación y sugerencias para la mejora del sistema haciéndolo más eficiente.

INDICE GENERAL

AGRADECIMIENTOS.....	i
DEDICATORIA.....	ii
TRIBUNAL DE SUSTENTACIÓN	iii
DECLARACIÓN EXPRESA.....	iv
RESUMEN.....	v
INDICE GENERAL	viii
ABREVIATURAS.....	xii
CAPÍTULO 1	1
MARCO TEÓRICO	1
1.1. INTRODUCCION.....	1
1.2. LA TECNOLOGÍA AJAX YA EXISTÍA	2
1.3. ENTENDIENDO AJAX.....	3
1.4. ¿CÓMO ES DIFERENTE AJAX?.....	5
1.5. ¿QUIEN ESTÁ USANDO AJAX?	6
CAPÍTULO 2	8
DESCRIPCIÓN DEL PROYECTO	8

2.1.	ANTECEDENTES	8
2.2.	DESCRIPCIÓN DEL PROBLEMA	10
2.3.	OBJETIVOS DEL PROYECTO	11
2.3.1.	Objetivo General	11
2.3.2.	Objetivos Específicos.....	11
2.4.	ANÁLISIS Y ESPECIFICACIONES	12
2.4.1.	Descripción del Funcionamiento del Sistema	12
2.4.2.	Análisis de Requisitos Funcionales.....	13
2.4.3.	Análisis de Requisitos NO Funcionales.....	14
2.4.3.1.	Costo	15
2.4.3.2.	Web.....	15
CAPÍTULO 3		16
ANÁLISIS Y DISEÑO		16
3.1.	ANÁLISIS GENERAL DEL NEGOCIO.....	16
3.2.	PLATAFORMA TECNOLÓGICA	20
3.2.1.	NetBeans IDE 6.5.1.....	20
3.2.2.	MySQL Query Browser	21
3.3.	DETALLE DE LAS HERRAMIENTAS UTILIZADAS.....	22
3.3.1.	Hibernate	22
3.3.2.	YUI Library	23
3.3.3.	JQuery	24

3.4.	DISEÑO Y METODOLOGÍA UTILIZADA	26
3.5.	ROLES Y USUARIOS.....	27
3.6.	MÓDULO “ESTIMATIVOS”	28
3.7.	DIAGRAMA DE CLASES	33
3.8.	DIAGRAMA DE CASOS DE USO.....	34
CAPÍTULO 4	35
IMPLEMENTACIÓN	35
4.1.	INTRODUCCIÓN.....	35
4.2.	MAPEO DE HARDWARE Y SOFTWARE.....	36
4.3.	ARQUITECTURA.....	37
4.3.1.	Modelo de Acceso a Datos.....	39
4.4.	USO ESTRATÉGICO DE LA TECNOLOGÍA.....	40
4.4.1.	Persistencia usando Hibernate	40
4.4.2.	CRUD (Crear, Leer, Modificar y Eliminar).....	41
4.4.3.	Controladores	43
4.4.4.	Vistas	44
4.5.	MÓDULO USUARIOS	45
4.6.	MÓDULO CLIENTES	47
4.7.	MÓDULO CONTENEDORES	49
4.8.	MÓDULO REPARACIONES	51
4.9.	MÓDULO ESTIMATIVOS	52

4.10. SEGURIDAD	56
4.5.1. Sesiones	57
4.5.2. Filtro para el inicio de sesión	58
4.5.2. Filtro para el acceso a las páginas	59
CONCLUSIONES Y RECOMENDACIONES	0
CONCLUSIONES.....	
RECOMENDACIONES.....	
GLOSARIO.....	
REFERENCIAS BIBLIOGRÁFICAS	

ABREVIATURAS

API	Application Programming Interface
AJAX	Asynchronous Javascript and XML
CSS	Cascading Style Sheets
DOM	Document Object Model
HTTP	Hypertext Transfer Protocol
JS	Java Script
JSP	JavaServer Pages
JVM	Java Virtual Machine
MVC	Modelo Vista Controlador
SQL	Structured Query Language

URL	Uniform Resource Locator
XHTML	Hypertext Markup Language
XML	Extensible Markup Language
YUI	Yahoo User Interface

CAPÍTULO 1

MARCO TEÓRICO

1.1. INTRODUCCION

Hace un tiempo AJAX parece ser la palabra de moda en el “mundo” del desarrollo de aplicaciones Web; de hecho muchos lo escuchan nombrar pero pocos saben qué es realmente y, menos aún, saben en donde buscar información clara sobre qué es esta nueva “maravilla”.

¿Por qué es tan interesante AJAX? Porque en realidad AJAX no es una tecnología, sino la unión de varias tecnologías que juntas pueden lograr cosas realmente

impresionantes como GoogleMaps, Gmail o algunas otras aplicaciones muy conocidas: AJAX, en resumen, es el acrónimo para Asynchronous JavaScript + XML es un conjunto de tecnologías usadas del lado del cliente **(5)** y el concepto es: mantenerse en la misma página mientras scripts y rutinas van al servidor buscando, en background, los datos que son usados para actualizar la página, mostrando u ocultando porciones de la misma. Lo que evita que se pierda la continuidad de la página en que se encuentra actualmente el usuario.

1.2. LA TECNOLOGÍA AJAX YA EXISTÍA

¿Quién no ha utilizado alguna vez un frame oculto en una página web con el objetivo de bajarse algunos datos del servidor sin tener que recargar la página actual? Esto se hace desde tiempos inmemoriales y cumple con la definición de lo que es AJAX. Entonces, ¿por qué tanto revuelo?

El éxito actual de la tecnología AJAX se debe a que los navegadores más importantes han estandarizado un objeto Javascript, llamado XMLHttpRequest, que permite hacer peticiones al servidor desde la página actual sin recargarla. Además, la respuesta puede ser tratada desde Javascript, bien sea como texto, bien sea como un objeto DOM (XML).

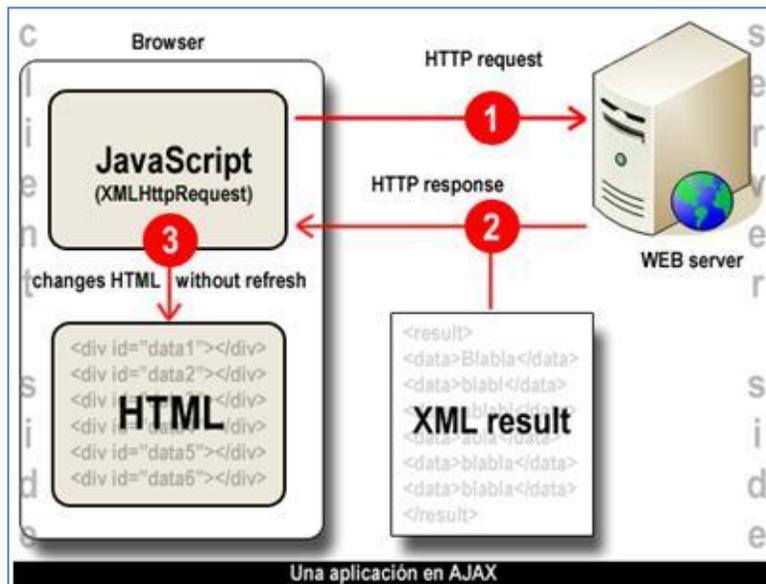


Figura 1: Ejemplo de una aplicación AJAX

1.3. ENTENDIENDO AJAX

AJAX no es una tecnología. Es realmente muchas tecnologías, cada una floreciendo por su propio mérito, uniéndose en poderosas nuevas formas. AJAX incorpora:

- Presentación basada en estándares usando XHTML y CSS;
- Exhibición e interacción dinámicas usando el Document Object Model;
- Intercambio y manipulación de datos usando XML;

- Recuperación de datos asincrónica usando XMLHttpRequest;
- Y JavaScript poniendo todo junto.

Su compatibilidad con nuevas tecnologías abren un abanico de posibilidades para el desarrollador, una de estas nuevas tecnologías es la librería JQuery de código abierto que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web

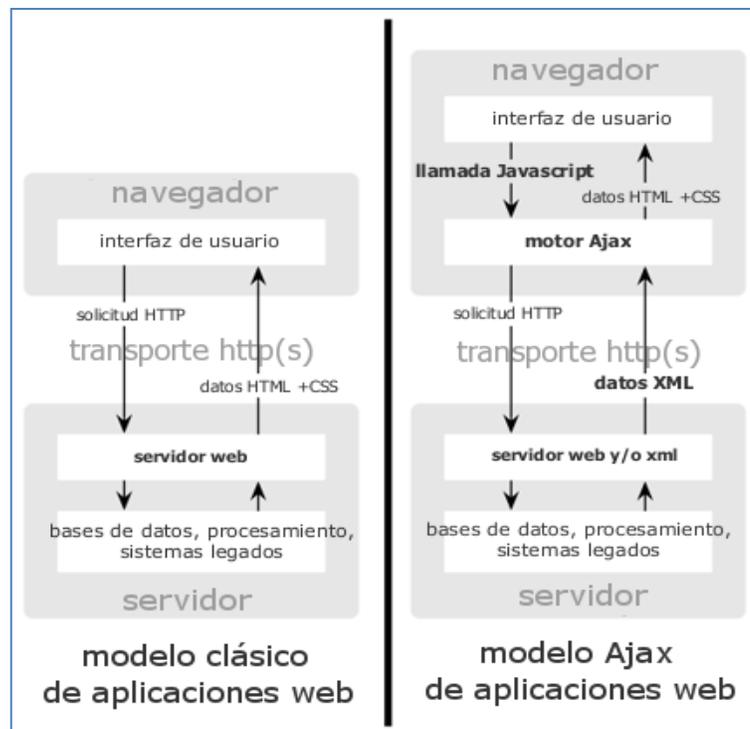


Figura 2: El modelo tradicional para las aplicaciones Web (izq.) comparado con el modelo de AJAX (der.). (10)

1.4. ¿CÓMO ES DIFERENTE AJAX?

Es diferente porque en vez de cargar un página Web, al inicio de la sesión, el navegador carga al motor AJAX (escrito en JavaScript y usualmente ubicado en un frame oculto). Este motor es el responsable por direccionar la interfaz que el usuario ve y por comunicarse con el servidor en nombre del usuario. El motor AJAX permite que la interacción del usuario con la aplicación suceda asincrónicamente (independientemente de la comunicación con el servidor). Así el usuario nunca estará mirando una ventana en blanco del navegador y un icono de reloj de arena esperando a que el servidor haga algo.

Al hablar de sincronismo, se refiere a la interacción que tiene el usuario y la aplicación con el servidor, al decir que la comunicación se realiza de manera síncrona, quiere decir que lo que el usuario ve, está sincronizado con el procesamiento del servidor, esto significa que la visualización de la aplicación web y el procesamiento del servidor, van uno detrás de otro, no al mismo tiempo, en cambio, cuando hablamos de que la comunicación se realiza de manera asíncrona, quiere decir que tanto lo que se refiere a la visualización de la aplicación web como lo que es el procesamiento en el servidor, ambos pueden ocurrir al mismo tiempo.

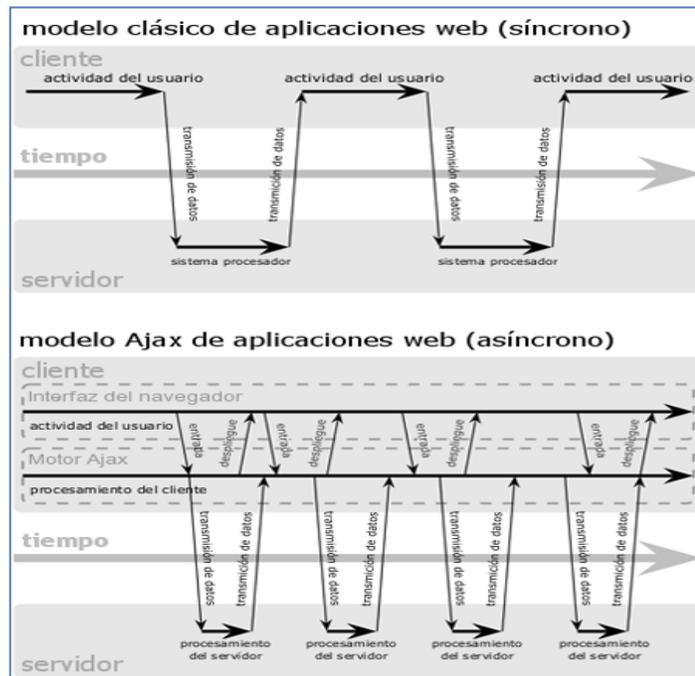


Figura 3: Patrón de interacción sincrónica de una aplicación Web tradicional (arriba) comparada con el patrón asincrónico de una aplicación AJAX (abajo). (10)

1.5. ¿QUIEN ESTÁ USANDO AJAX?

Google está haciendo una significativa inversión en el acercamiento AJAX. Todos los grandes productos que Google ha introducido en el último año (Gmail, la última versión de Google Groups, Google Suggest, y Google Maps) son aplicaciones AJAX. Otros están siguiendo la tendencia: muchas de las funciones que la gente ama en Flickr dependen de AJAX, y el motor de búsqueda de Amazon A9.com aplica tecnologías similares.

Estos proyectos demuestran que AJAX no es solo técnicamente importante, sino también prácticos para aplicaciones en el mundo real. Esta no es otra tecnología que solo trabaja en un laboratorio, las aplicaciones AJAX pueden ser de cualquier tamaño.

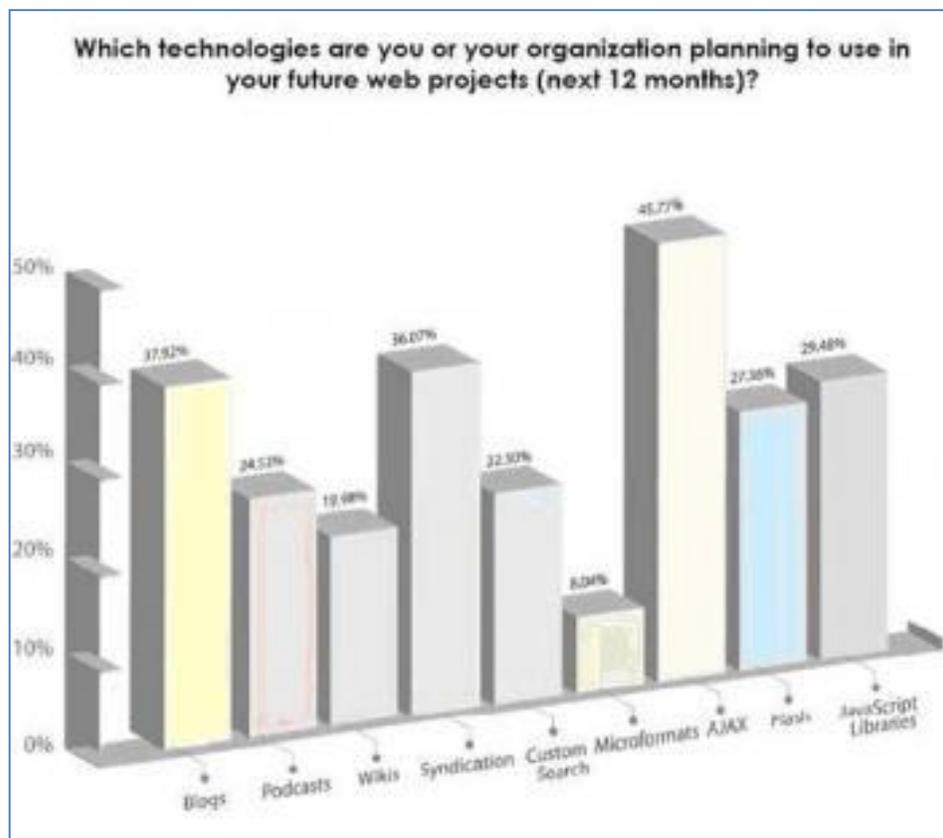


Figura 4: Tendencias de las TICs para el 2010 (2)

CAPÍTULO 2

DESCRIPCIÓN DEL PROYECTO

2.1. ANTECEDENTES

Conami S.A. (1) es una empresa con 14 años en el mercado, fundada en el año de 1996. Ofrece diversos servicios de auditoría, logística y operación portuaria, los cuáles, son de gran importancia para actividades comerciales como importación, exportación, producción, seguros y otros negocios relacionados.

Durante el tiempo que tiene en el mercado, ha servido y sirve a importantes empresas ecuatorianas y extranjeras. Lo que ha generado resultados positivos en las actividades operativas y logísticas de los clientes y ha brindado la oportunidad de desarrollar excelentes relaciones comerciales.

Los años de experiencia, trabajo calificado y constancia, les han permitido ser representantes y agentes para el Ecuador de reconocidas empresas internacionales y se puede citar actualmente a **UNICON INTERNATIONAL** (que opera en 15^o países con una cartera de más de 200 clientes, realizando un promedio de 80,000 inspecciones mensuales de contenedores).

Por otro lado, conociendo que parte de la misión de la ESPOL; es la de formar profesionales de excelencia, líderes, emprendedores, con sólidos valores morales y éticos que contribuyan al desarrollo del país, para mejorarlo en lo social, económico, ambiental y político; la Empresa Conami S.A. en conocimiento de la labor que ejecuta la ESPOL solicitó nuestra ayuda como egresados, con una propuesta para elaborar un sistema que les permita automatizar el proceso de elaboración de Estimativos y así permitir el control de las reparaciones que se realizan a los contenedores de los clientes tanto nacionales como internacionales que maneja la empresa entre ellos **GRUPO PALMAR** y **GRUPO NOBOA**. Es así como

por medio del Seminario de Graduación dirigido por nuestro tutor el Msc. Carlos Martín procedimos a elaborar dicho sistema.

2.2. DESCRIPCIÓN DEL PROBLEMA

Hoy en día elaborar los estimativos es una tarea compleja y que consume más tiempo de lo que debería, tiempo que puede ser empleado para otras actividades administrativas importantes para el progreso de la empresa.

Los estimativos son hojas de Excel, lo que implica que todo su proceso es hecho manualmente, toda la información debe ser tipada por la persona encargada de dicha tarea.

Para llenar el detalle del estimativo se debe consultar la información de las reparaciones, dicha información se encuentra en una hoja de Excel por lo que buscar información y utilizarla es una tarea nada sencilla, toma mucho tiempo buscar algo de información para poder llenar el estimativo y si consideramos que un estimativo puede contener muchas reparaciones entonces nos damos cuenta que el tiempo de búsqueda es muy grande.

2.3. OBJETIVOS DEL PROYECTO

2.3.1. Objetivo General

Desarrollar una aplicación web que permita el control y seguimiento, en la Generación y Reparación de los Contenedores, para la empresa Conami S.A. utilizando AJAX.

2.3.2. Objetivos Específicos

- Desarrollar un módulo que permita el ingreso, consulta y actualización de los datos de los contenedores que ingresan en la empresa.
- Desarrollar un módulo que permita el ingreso, consulta y actualización de los datos de los clientes de la empresa.
- Desarrollar un módulo que permita el ingreso, consulta y actualización de los datos de las reparaciones que realiza la empresa.
- Desarrollar un módulo que permita el ingreso, consulta y actualización de los datos de los usuarios del sistema que son los mismos empleados de la empresa.

- Elaborar un módulo para la emisión de los Estimativos (documento similar a una factura) que permita el ingreso, consulta, actualización y seguimiento de los éstos en la empresa.
- Generar los reportes de los Estimativos con los datos de las Reparaciones que se les hayan realizado a cada contenedor.

2.4. ANÁLISIS Y ESPECIFICACIONES

Analizaremos la parte teórica del sistema a desarrollar, haremos una descripción del funcionamiento del sistema y se detallaran los requisitos funcionales y no funcionales.

2.4.1.Descripción del Funcionamiento del Sistema

Las empresas que se dedican a asuntos portuarios en este caso REPARACIÓN DE CONTENEDORES necesitan generar un documento donde se especifica en detalle las reparaciones que se hicieron a cada contenedor. Estos contenedores previamente han ingresado a los talleres de CONAMI y además poseen información muy valiosa para la empresa como por ejemplo: datos del cliente dueño del contenedor, datos adicionales del contenedor, datos de cada reparación que se

realizó, fecha de la elaboración del estimativo y el total del costo del estimativo. Los datos de los costos de cada reparación y cálculos adicionales son propios del negocio. La generación de estimativos es una actividad diaria para este tipo de empresas; de aquí nace la necesidad de automatizar este proceso con esta aplicación web y elevar su rendimiento usando AJAX.

Por último, se tendrá en cuenta el rol del empleado y así definir las limitaciones que tiene sobre el sistema. Por eso se implementará un sistema de control de usuarios, cargos y permisos para la aplicación.

2.4.2. Análisis de Requisitos Funcionales

En esta parte de la documentación detallaremos los requisitos funcionales de cada uno de los módulos del sistema.

MÓDULO CONTENEDORES

- Ingreso de un nuevo Contenedor.
- Consulta de los Contenedores.
- Actualización de los datos de los Contenedores.

MÓDULO CLIENTES

- Ingreso de un nuevo Cliente.

- Consulta de los Clientes.
- Actualización de los datos de los Clientes.

MÓDULO REPARACIONES

- Ingreso de una nueva Reparación.
- Consulta de las Reparaciones.
- Actualización de los datos de las Reparaciones.

MÓDULO USUARIOS

- Ingreso de un nuevo Usuario.
- Consulta de los Usuarios.
- Actualización de los datos de los Usuarios.

MÓDULO ESTIMATIVOS

- Ingreso de un nuevo Estimativo.
- Consulta de los Estimativos.
- Actualización de los datos de los Estimativos.
- Generar los reportes de los Estimativos de cada Contenedor.

2.4.3. Análisis de Requisitos NO Funcionales

Como sabemos de antemano los requisitos no funcionales son implícitos al sistema y no son funcionalidades concretas de este.

2.4.3.1. Costo

La aplicación será construida con software libre, será más fácil para el cliente mantenerla ya que no necesitará comprar licencias para actualizar sus programas ya que se ha optado por trabajar con software libre, como son NetBeans IDE, YUI, JQuery.

2.4.3.2. Web

La aplicación será desarrollada en forma web por las siguientes razones:

Primero Conami S.A. tiene oficinas y talleres en las ciudades de Guayaquil y Puerto Bolívar, al tener una aplicación web los empleados (usuarios de la aplicación) podrán acceder a ella independientemente de la ciudad en que se encuentren.

Otra de las razones por la cual el proyecto fue desarrollado como aplicación web es la inversión; actualmente el costo de mantener una aplicación web es de \$200 anuales, más el costo por la adquisición de un equipo específico para el almacenamiento de datos, es decir, un servidor de bases de datos cuyo costo está por los \$700. Convirtiéndose el gasto en una cantidad muy accesible para empresas de pequeña y mediana escala.

CAPÍTULO 3

ANÁLISIS Y DISEÑO

3.1. ANÁLISIS GENERAL DEL NEGOCIO

Un estimativo es un documento que refleja la actividad de la empresa Conami S.A. en cuanto a contenedores y reparaciones por lo cual dicho documento contiene información importantísima para la empresa y para los clientes de ella, ya que el documento original permanece en las oficinas de Conami S.A. y una copia es entregada al cliente.

Cabe recalcar que un Estimativo es un documento esencial para llevar la contabilidad de la empresa. Según esta descripción de lo que es un Estimativo podríamos comparar a éste con una Factura.

A continuación le presentamos una imagen de cómo nos mostraron un Estimativo al momento de visitar la empresa:

Sec.	location	Concept	Damage	Qty	Hours	Labores	Material
1	R/L	TOP COVING SEAL 12MT	LOOSE	1	2.00	16.00	20.00
2	RIGHT	INT DOOR STR/SEAL FRAME 120CM	LOOSE	1	0.50	4.00	5.20
3	R/L	INT TOP COVING RPL RIVET X 18	MISSING	18	0.75	6.00	7.20
MACHINERY PARTS							
1	MACH	DIGITAL DISPLAY REPAIR	DMG	1	1.00	8.00	220.00
AMOUNT STRUCTURAL PARTS:			\$	58.40			
AMOUNT MACHINERY PARTS:			\$	228.00			
TOTAL TECNICO					4.25	\$34.00	\$252.40
Surveyor : JCS						Total :	\$286.40

Figura 5: Imagen de un Estimativo

A continuación se presenta una figura que muestra la manera en que se está manejando la información de las reparaciones; en ella se puede observar la

descripción de cada reparación, el costo de la hora de trabajo, el costo del material y el incremento que finalmente se le realiza a estos costos.

REEFER STRUCTURE REPAIR SCHEDULE									
DESCRIPTION	CM/QUANTITY	H/H	INCR/ADD	H/H	Base	C. Mat.	Valor	C. Mat.	
					Base	Increment	Incremento		
1. TOP RAILS									
WELD ALL TOP RAILS STEEL/ALU	15CM	0.25	15CM	0.15		1.00		1.00	
STR/WELD ALL TOP RAILS STEEL/ALU	15CM	0.50	15CM	0.25					
STR TOP SIDE RAIL STEEL	30CM	0.50	15CM	0.15		2.50		1.70	
STR TOP SIDE RAIL ALU	15CM	0.25	15CM	0.10		0.00		0.00	
SECTION TOP S RAIL STEEL	30CM	4.20	30CM	0.90		34.89		29.41	
SECTION TOP S RAIL ALU/WITH SPLICE PLATES	100CM	6.00	15CM	0.50		231.00	60CM	71.83	
REPL TOP S RAIL STEEL 20'		22.00				455.45			
REPL TOP S RAIL STEEL 40'		34.00				857.65			
REPL TOP S RAIL ALU 20'		24.00				732.70			
REPL TOP S RAIL ALU 40'		36.00				1385.70			
INSERT TOP S RAIL STEEL	30CM	1.94	15CM	0.46		22.10		9.50	
INSERT TOP S RAIL ALU	30CM	2.00	15CM	0.25		20.72		10.20	
STR DOOR HEADER STEEL	15CM	0.35	15CM	0.15		1.70		1.70	
SECTION DOOR HEADER STEEL W/FOAM	30CM	2.25	15CM	0.71		24.50		12.25	
REPL CMPL DOOR HEADER STEEL		9.00							
INSERT DOOR HEADER STEEL W/FOAM	30CM	1.50	15CM	0.46		7.83		4.90	
INER FULL DOOR HEADER		7.78				76.46			
STR FRT TOP RAIL STEEL	15CM	0.83	15CM	0.28		0.88		0.20	
SECTION FRT TOP RAIL STEEL W/FOAM	30CM	2.50	15CM	0.71		17.41		7.78	
REPL FRT TOP RAIL STEEL W/FOAM		11.88				126.33			

Figura 6: Imagen de las Reparaciones

En la figura 7 se detalla el manual de códigos de las reparaciones; la tabla consta del código de la reparación, la descripción de la misma, el costo de la hora de trabajo y finalmente el costo del material.

	A	B	C	D	E
1	MANUAL DE CODIGOS				
2	CODIGO	DESCRIPCION	H H	C MAT	
3	TA0001	REPLACE LLAVE DE PARTLOW Sujetador	0.30	21.00	
4	TA0002	REPLACE PLUG 230V	0.50	445.38	
5	TA0003	REPLACE PERNO DE COVER EVAP.	0.10	1.50	
6	TA0004	REPLACE PERNO DE COVER COMPRESOR	0.10	1.50	
7	TA0005	REPLACE DIAGRAMA ELECTRICO	0.10	15.71	
8	TA0006	REPLACE SENSOR COMBINATION 12-00500-015V		90.00	CARRIER
9	TA0007	REPLACE VAVE SOLENOIDE DE SUCCION		210.00	CARRIER
10	TA0008	REPLACE SENSOR COMPRESOR SABROE 414300		160.00	
11	TA0009	REPARACION DE MICRO MK2i		450.00	
12	TA0010	REPLACE KIT DE MODULACION (TODO TERMOKING)	2.00	280.00	
13	TA0011	REPLACE SENSOR COMPRESOR SCROLL 412447		230.00	
14	TA0012	REPLACE GRILL COND. SCROLL 98-5925		220.00	
15	TA0013	REPLACE MODULO DE POWER SCROLL 45-1806		870.00	
16	TA0014	REPLACE MODULO DE SALIDA SCROLL 451853		1,300.00	
17	TA0015	REPARACION DE MODULO POWER SCROLL		450.00	
18	TA0016	REPARACION DE MODULO DE SALIDA SCROLL		500.00	
19	TA0017	REPARACION DE MICRO UP-D	2.00	600.00	
20	TA0018	REPLACE VALVE RELIEF		35.00	

Figura 7: Imagen de las Reparaciones por Códigos

Entonces cuando alguien debe hacer un estimado debe abrir esta hoja de Excel y para llenar las reparaciones, buscar la descripción de la reparación a consultar, fijarse en los datos de la misma como por ejemplo el valor hora hombre en dólares, el costo del material entre otros.

La propuesta es desarrollar un sistema que permita a nuestro cliente automatizar el proceso de elaboración de los estimativos, reduciendo el tiempo empleado en dicha tarea, además que permita almacenar la información en una base de datos la cual permita acceder y modificar su contenido de una manera fácil y casi inmediata.

3.2. PLATAFORMA TECNOLÓGICA

3.2.1. NetBeans IDE 6.5.1

El IDE de NetBeans **(7)** es un entorno integrado de desarrollo muy conocido, disponible para:

- Windows.
- Mac.
- Linux.
- Solaris.

El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicaciones que permiten a los desarrolladores crear rápidamente aplicaciones web, empresas, escritorio y aplicaciones móviles utilizando la plataforma Java, así como:

- JavaFX
- PHP
- JavaScript y Ajax
- Ruby
- C/C++.



Figura 8: NetBeans IDE (7)

El proyecto NetBeans es compatible con una vibrante comunidad de desarrolladores y ofrece una amplia documentación y recursos de capacitación, así como una amplia selección de plug-ins de terceros.

3.2.2. MySQL Query Browser

MySQL Query Browser **(7)** es una herramienta gráfica proporcionada por MySQL AB que permite crear, ejecutar, y optimizar consultas en un ambiente gráfico, donde el MySQL Administrator está diseñado para administrar el servidor MySQL. MySQL Query Browser está diseñado para ayudarle a consultar y analizar datos almacenados en su base de datos MySQL.

Aunque todas las consultas ejecutadas en el MySQL Query Browser pudieran ser también ejecutadas en la utilidad de línea de comando, MySQL Query Browser permite de una manera más intuitiva y grafica la consulta y la edición de datos.



Figura 9: MySQL Query Browser (7)

3.3. DETALLE DE LAS HERRAMIENTAS UTILIZADAS

3.3.1. Hibernate

Hibernate **(6)** busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora

(orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

3.3.2. YUI Library

Yahoo User Interface (YUI) **(9)**, es una serie de bibliotecas escritas en JavaScript, para la construcción de aplicaciones interactivas; que fueron liberadas bajo licencia BSD por parte de la compañía Yahoo. Dichas bibliotecas son utilizadas para el desarrollo web específicamente para ser usadas como la programación de

aplicaciones de escritorio, con componentes vistosos y personalizables y con una amplia implementación con AJAX.

La biblioteca está completamente documentada en su página web y se compone de seis componentes: Núcleo YUI, utilidades, controles UI, componentes CSS, herramientas de desarrollo y de construcción.

Entre sus principales características podemos mencionar:

- El núcleo de YUI es un ligero conjunto de herramientas para manejar eventos y manipular el árbol DOM.
- YAHOO Global Object: Los Objetos Globales Yahoo contienen utilidades y otras infraestructuras base para la biblioteca.
- Dom Collection: Ayuda para la manipulación del árbol DOM, incluyendo posicionamiento de elementos y gestión de estilos CSS.
- Event Utility: Permite acceder de forma segura y sencilla a los eventos de un navegador web y, mediante el objeto *Custom Event*, publicar y suscribirse a eventos customizados.

3.3.3. JQuery

JQuery es un framework Javascript **(11)**, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

Este framework Javascript, nos ofrece una infraestructura con la que tendremos mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, con JQuery obtendremos ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de AJAX, etc. Simplemente debemos conocer las librerías del framework y programar utilizando las clases, sus propiedades y métodos para la consecución de nuestros objetivos.

Entre sus principales características podemos mencionar:

- Selección de elementos DOM.
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- AJAX.
- Plugins del Javascript.

3.4. DISEÑO Y METODOLOGÍA UTILIZADA

Para empezar, los JSPs y servlets se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java para él. Cada servlet (o JSP, a partir de ahora lo usaremos de forma indistinta) se ejecuta en un hilo distinto; pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo. Su persistencia le permite también manejar de forma más eficiente: la conexión a la base de datos y el manejo de sesiones.

Para el desarrollo del software se ha llevado una organización estratégica para almacenar el código, dividiendo en paquetes, a continuación se procede a detallar lo mencionado:

- **Web Pages:** Archivos con extensión “.jsp”, son las páginas web del sistema, en este grupo se maneja todo el entorno que será llamado “Vista”, ya que es la interfaz con el usuario.
- **Java Scripts (4):** Aquí se almacenan los archivos con extensión “.js” organizados según los objetos que se instancian en la aplicación, estos

archivos contiene los scripts que se usan de librerías externas que facilitan el desarrollo y aumentan su eficiencia.

- **Servlets:** En esta sección se almacenan los archivos con extensión “.java” que manejan la información y las transacciones con la base de datos para luego transmitirla a las páginas web.
- **Filtros:** Aquí encontramos los archivos con extensión “.java” que manejan la identificación de usuarios para darles permiso a las diferentes funcionalidades del sistema.
- **Clases:** Gracias a las bondades de la tecnología Hibernate aquí podemos almacenar las clases que son mapeadas de la base de datos que servirán para instanciar objetos y poder usar sus atributos y métodos para lograr las funcionalidades q tenemos como objetivo.

3.5. ROLES Y USUARIOS

El control de acceso de la aplicación se lo hará a través de roles, lo que permite distribuir de manera más práctica las funciones que van a tener los usuarios y facilitará el trabajo del administrador de la aplicación.

Por el efecto se han determinado los siguientes roles:

- **Administrador:** Este rol tiene acceso a todos los módulos del sistema. A su vez cuenta con un módulo al que sólo él puede acceder, que permite crear y modificar usuarios.
- **Empleado:** Este rol, a diferencia del rol de administrador, cuenta con el acceso a todos los módulos del sistema menos al módulo usuarios.

3.6. MÓDULO “ESTIMATIVOS”

Cuando un contenedor entra al taller para ser reparado o para darle mantenimiento, Conami S.A. elabora un detalle de las reparaciones que se le hicieron a dicho contenedor, reparaciones que tienen su valor en dólares y al final se presenta una sumatoria a la cual se le agregan impuestos presentando un valor total que será cobrado al dueño del contenedor.

Además un Estimativo tiene información importante como el Cliente al cual pertenece el contenedor, la fecha de elaboración del estimativo, también la identificación del contenedor ya que Conami lleva un registro de todos los que han entrado a sus talleres y les asigna una identificación única.

En un estimativo también se especifica en qué buque arribó el contenedor para manejar un historial más preciso que podría ser necesario en algún momento. Cada

estimativo tiene su identificación, que será registrada en este documento con el nombre E.I.R.

En la figura 11 se detalla el modelo de negocio de la empresa Conami S.A. descrito en los párrafos anteriores.

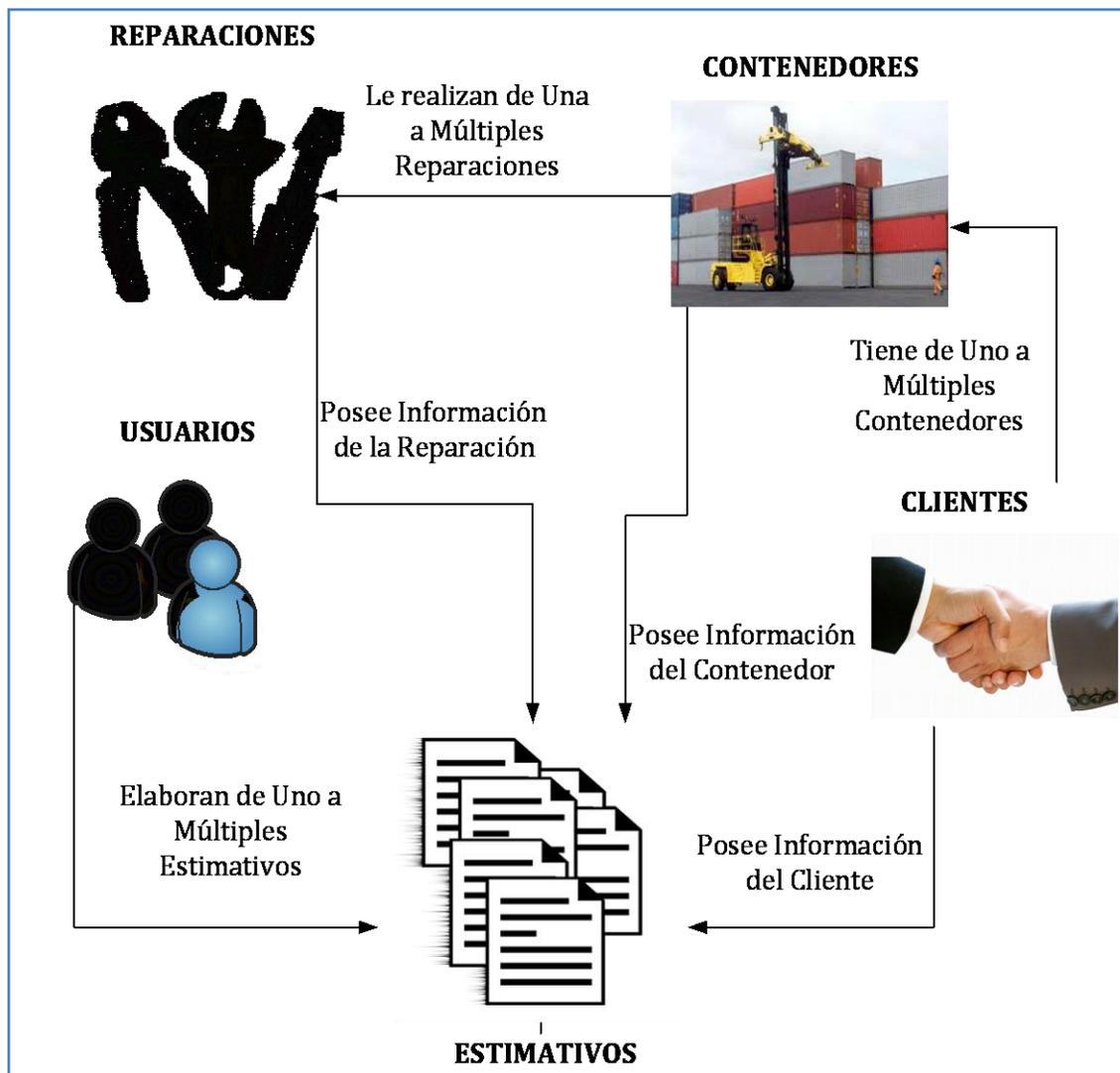


Figura 11: Diagrama operacional de la empresa Conami S.A.

La información que tiene un Estimativo es:

Línea

Nombre del cliente de Conami S.A. dueño del contenedor del que se hace referencia en el estimativo.

N° del Contenedor

Identificación asignada por Conami S.A. a cada contenedor que ha ingresado a sus talleres ya que se lleva un historial de dichos contenedores.

Tamaño

Tamaño en pulgadas del contenedor.

Fecha

Fecha en la cual se elabora el estimativo. Información importante y que no puede faltar.

M/N

Buque de donde se descargó el contenedor.

Ubicación

Taller de Conami S.A. donde se encuentra el contenedor.

Costo hora Hombre

Valor en dólares del costo de hora hombre.

E.I.R. #

Es un número que viene membretado en un documento físico un papel que se relaciona a un estimativo digital.

Locación

Se refiere a la ubicación de la reparación en el contenedor LEFT, RIGHT.

Concepto

Es el Nombre o Descripción de la reparación.

Daño

Es el tipo de daño que ha sufrido un contenedor.

Cantidad

Es la cantidad de reparaciones que se han hecho al contenedor. Puede darse el caso que Cantidad =2, quiere decir que fue necesario realizar 2 reparaciones del mismo tipo al contenedor.

Horas

Costo en Dólares de cada hora trabajada de la reparación referenciada.

Labores

Es la multiplicación de Horas x Costo Hora Hombre.

Material

Costo en Dólares de los materiales de una reparación.

Número de Piezas Estructurales

Es la sumatoria del Costo de Labores y Material de las reparaciones de Tipo ESTRUCTURA. Las reparaciones se dividen en dos tipos: Estructura y Maquinaria.

Número de Piezas de Maquinaria

Es la sumatoria del Costo de Labores y Material de las reparaciones de Tipo MAQUINARIA.

Supervisor

Son las iniciales del nombre del responsable que aprueba el estimativo.

Total

Es la sumatoria de Número de Piezas Estructurales y Número de Piezas de Maquinaria.

3.7. DIAGRAMA DE CLASES

En el diagrama a continuación se presenta de manera resumida las entidades más importantes que se utilizan en la aplicación, dando un enfoque general de cómo está constituido el sistema.

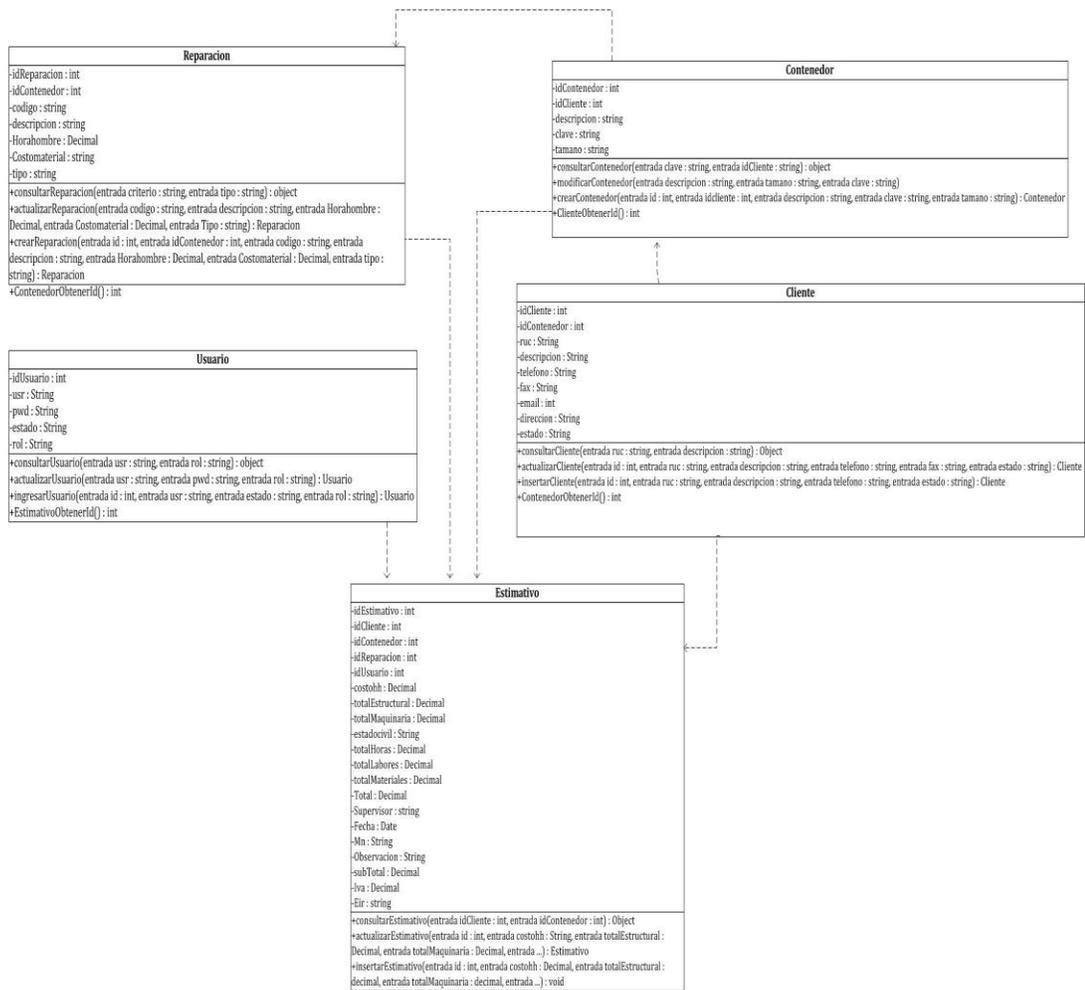


Figura 12: Diagrama Clases del Sistema.

3.8. DIAGRAMA DE CASOS DE USO

En el diagrama que se presenta a continuación, mostramos la relación entre los actores y la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. Los casos de uso (funcionalidades) están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea.

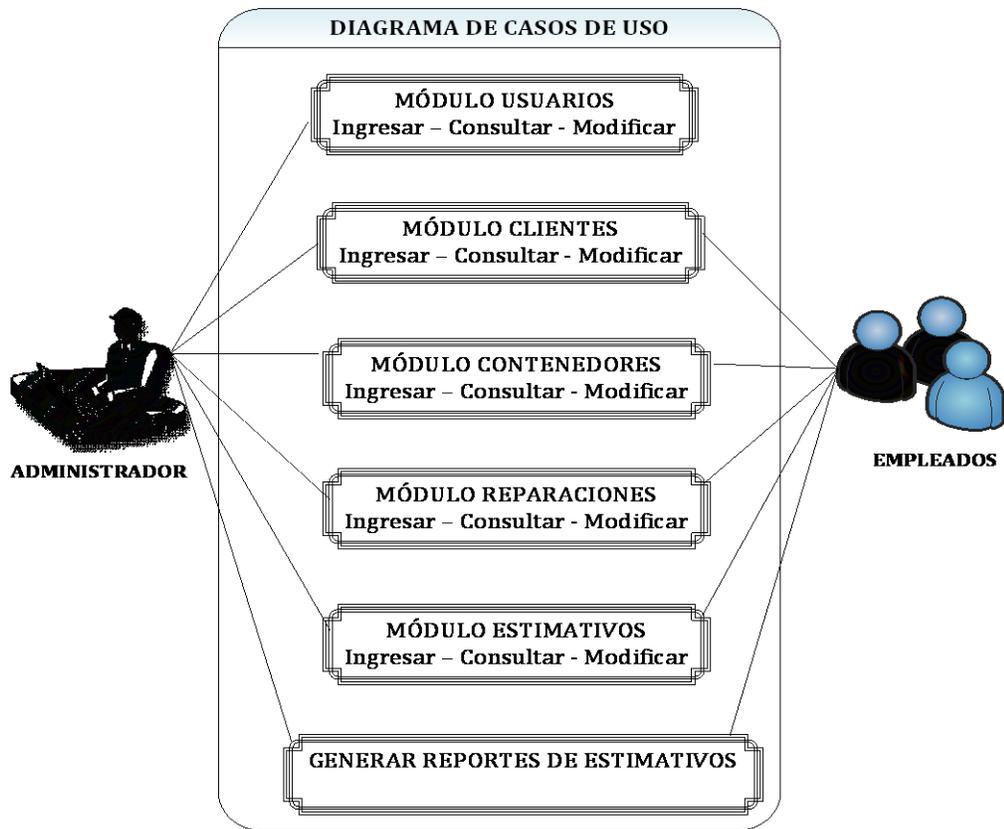


Figura 13: Diagrama de Casos de Uso del Sistema

CAPÍTULO 4

IMPLEMENTACIÓN

4.1. INTRODUCCIÓN

Luego de haber planteado las ideas que sustentan este proyecto y haber elaborado un diseño que ayude a plasmar estas ideas se debe de pensar cómo implementarlo.

Es importante aplicar conceptos de arquitectura de software para diseñar y elaborar un sistema, esto nos permite evitar problemas de reescritura de código a causa de cambios no esperados dentro del proyecto aquí descrito, nos brinda a su

vez una buena base para que el sistema sea escalable, mantenible, razones que sirvieron para aplicar en este proyecto el modelo de 3 capas.

4.2. MAPEO DE HARDWARE Y SOFTWARE

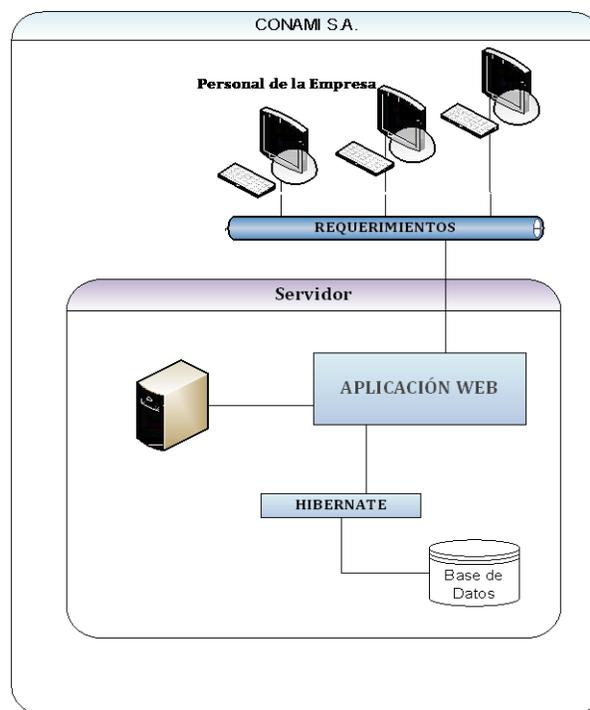


Figura 14: Mapeo de Hardware y Software

En la figura anterior mostramos como entrará en funcionamiento la aplicación una vez entregada a la empresa Conami S.A. La aplicación web estará instalada en un servidor a través de cual accederán las maquinas que se encuentran en las oficinas

de Conami S.A., por el momento serán tres las computadoras que accederán a la aplicación.

Los requerimientos necesarios para hacer funcionar la aplicación en estas computadoras se encuentra detallado más adelante en la tabla de la arquitectura del sistema; pero tenemos la ventaja de que estas maquinas cuentan ya con todo lo necesario para ejecutar el sistema.

4.3. ARQUITECTURA

Para el acceso a datos se aplica el modelo de 3 capas, tomando como referencia los conceptos básicos de la Arquitectura en la Ingeniería de Software **(8)**, que se explica a continuación.

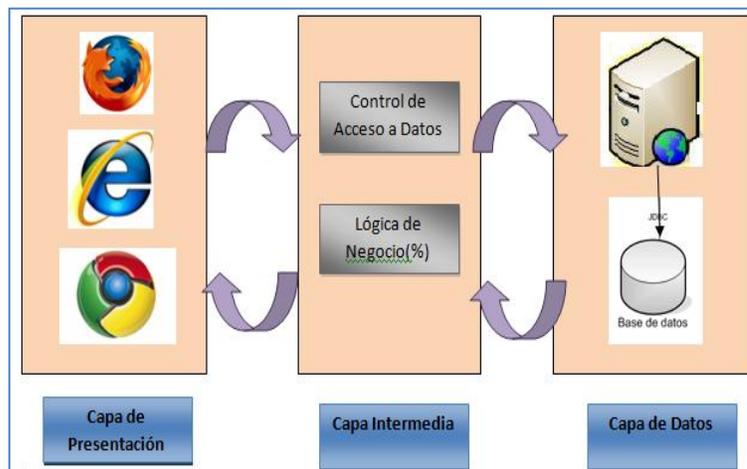


Figura 15: Modelo de 3 capas (8)

Capa de Acceso a Datos: Maneja las peticiones a la base de datos cumpliendo funciones de tipo consultas, ingresos y actualizaciones.

Capa Intermedia (lógica o capa de negocios): Esta capa invoca a la capa de acceso a datos y es la que realiza el procesamiento debido de la información para enviarla a la capa de presentación.

Capa de Presentación: Es la interfaz de usuario, se compone de formularios Web en los que se presentan la información requerida, conteniendo las reglas del negocio casi en su totalidad.

El sistema para la **GENERACIÓN Y SEGUIMIENTO DE REPARACIÓN DE CONTENEDORES** según los lineamientos dados está compuesto por la siguiente arquitectura de software:

Sistema Operativo	Windows Vista
Base de Datos	MySQL
Manejo de Persistencia	Hibernate
Lenguaje de Programación	Java usando modelo MVC y JSP
Herramienta de Desarrollo	Netbeans

Tabla 1: Arquitectura del Sistema.

4.3.1. Modelo de Acceso a Datos

Al estudiar cuál modelo sería el adecuado para manejar los datos del sistema acordamos usar el modelo de repositorio ya que comparten los datos a través de un almacén común teniendo así una mejor disponibilidad cuando cada uno de los módulos lo requieran.

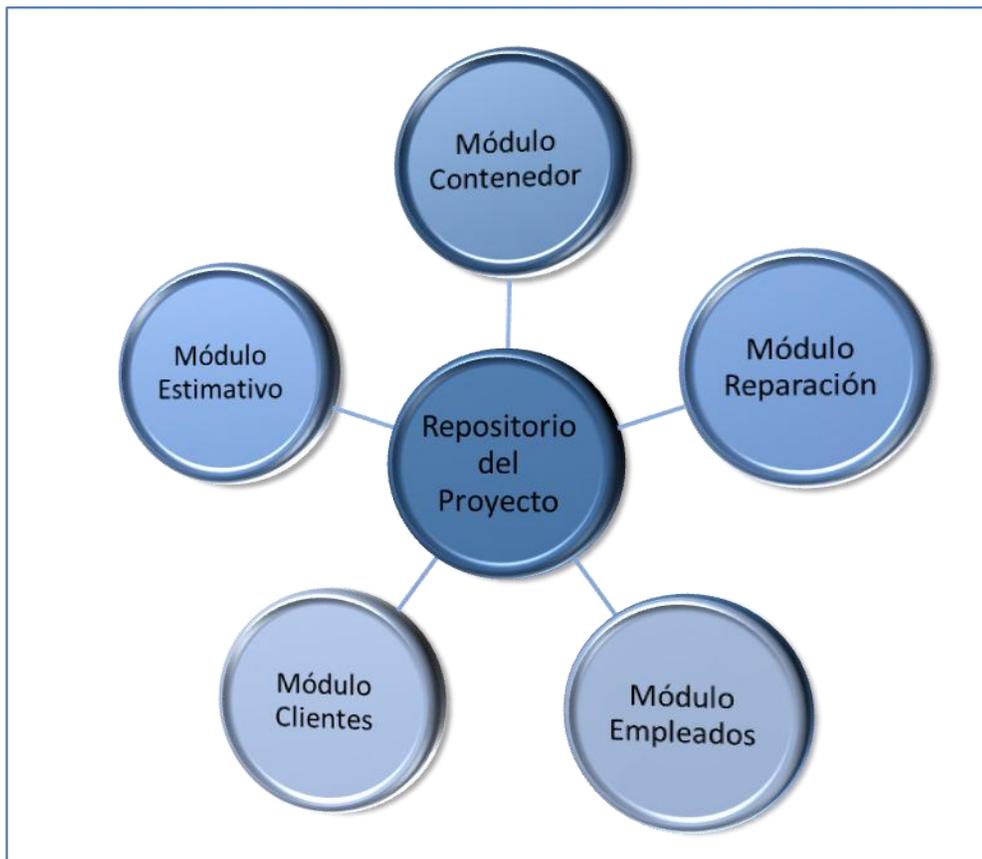


Figura 16: Modelo Repositorio de los datos del sistema

4.4. USO ESTRATÉGICO DE LA TECNOLOGÍA

4.4.1. Persistencia usando Hibernate

Aprovechando las bondades que ofrece la tecnología hoy en día en este caso Hibernate, podemos acelerar el proceso de desarrollo ingresando a las tablas de la base de datos de una manera sencilla utilizando el mapeo de una tabla a una clase.

Esto permite mantener consistencia en la base de datos al crear una sesión y de esta manera lograr el correcto uso de transacciones.

```
31 @Entity
32 @Table(name = "estimativo")
33 @NamedQueries({@NamedQuery(name = "Estimativo.findAll", query = "SELECT e FROM Estimativo e"),
34 @NamedQuery(name = "Estimativo.findById", query = "SELECT e FROM Estimativo e WHERE e.id = :id"),
35 @NamedQuery(name = "Estimativo.findByCostohh", query = "SELECT e FROM Estimativo e WHERE e.costohh = :costohh"),
36 @NamedQuery(name = "Estimativo.findByTotalestructural", query = "SELECT e FROM Estimativo e WHERE e.totalestructural = :totalestructural"),
37 @NamedQuery(name = "Estimativo.findByTotalmaquinaria", query = "SELECT e FROM Estimativo e WHERE e.totalmaquinaria = :totalmaquinaria"),
38 @NamedQuery(name = "Estimativo.findByTotalhoras", query = "SELECT e FROM Estimativo e WHERE e.totalhoras = :totalhoras"),
39 @NamedQuery(name = "Estimativo.findByTotallabores", query = "SELECT e FROM Estimativo e WHERE e.totallabores = :totallabores"),
40 @NamedQuery(name = "Estimativo.findByTotalmateriales", query = "SELECT e FROM Estimativo e WHERE e.totalmateriales = :totalmateriales"),
41 @NamedQuery(name = "Estimativo.findByTotal", query = "SELECT e FROM Estimativo e WHERE e.total = :total"),
42 @NamedQuery(name = "Estimativo.findBySupervisor", query = "SELECT e FROM Estimativo e WHERE e.supervisor = :supervisor"),
43 @NamedQuery(name = "Estimativo.findByFecha", query = "SELECT e FROM Estimativo e WHERE e.fecha = :fecha"),
44 @NamedQuery(name = "Estimativo.findByMn", query = "SELECT e FROM Estimativo e WHERE e.mn = :mn"),
45 @NamedQuery(name = "Estimativo.findByObservacion", query = "SELECT e FROM Estimativo e WHERE e.observacion = :observacion"),
46 @NamedQuery(name = "Estimativo.findBySubtotal", query = "SELECT e FROM Estimativo e WHERE e.subtotal = :subtotal"),
47 @NamedQuery(name = "Estimativo.findByIva", query = "SELECT e FROM Estimativo e WHERE e.iva = :iva"), @NamedQuery(name = "Estimativo.findByObservacion", query = "SELECT e FROM Estimativo e WHERE e.observacion = :observacion")
48 public class Estimativo implements Serializable {
49     private static final long serialVersionUID = 1L;
50     @Id
51     @GeneratedValue(strategy = GenerationType.IDENTITY)
52     @Basic(optional = false)
53     @Column(name = "id")
54     private Integer id;
55     @Column(name = "costohh")
56     private BigDecimal costohh;
57     @Column(name = "totalestructural")
58     private BigDecimal totalestructural;
59     @Column(name = "totalmaquinaria")
```

Figura 17: Mapeo de la Tabla Estimativo

El mapeo consiste en la representación de la tabla en una clase de tal manera que podemos instanciar dicha clase y tener un objeto en cual podemos identificar que los atributos son las columnas de la tabla mapeada.

Para acceder a los datos de una manera transparente se utilizan etiquetas “@NamedQuery” que representan consultas preestablecidas a la base de datos.

Esta tecnología nos libera de realizar un mapeo manual de la base de datos, durante la etapa de implementación siempre pueden surgir cambios debido a requerimientos nuevos, incluir un nuevo campo en una tabla de la base de datos puede representar un alto costo de tiempo de desarrollo pero usando Hibernate este tiempo es reducido a la mitad.

4.4.2. CRUD (Crear, Leer, Modificar y Eliminar)

Esta es la forma sencilla de realizar transacciones en la base datos: insertar, actualizar, eliminar, consulta; es el medio común que usan todos los objetos para interactuar con la base de datos, esta clase ofrece métodos generales con los cuales se pueden lograr los objetivos de la aplicación y funcionalidades implementadas.

```

23 public class Conexion {
24     private Session session;
25     public Conexion() {
26         session=HibernateUtil.getSessionFactory().openSession();
27     }
28     public List consultar(String consulta){
29         List lista;
30         lista=getSession().createQuery(consulta).list();
31         return lista;
32     }
33     public void ingresar_y_o_actualizar(Object obj){
34         Transaction tx= getSession().beginTransaction();
35         getSession().save(obj);
36         getSession().flush();
37         tx.commit();
38     }
39     public void eliminar(Object obj){
40         Transaction tx= getSession().beginTransaction();
41
42         getSession().delete(obj);
43         getSession().flush();
44         tx.commit();
45     }
46     public String xml(String consulta,String fila){
47         List lista;
48         String xml="<?xml version=\"1.0\" encoding=\"ISO-8859-1\"?><"+fila+"s>";
49
50         lista=getSession().createQuery(consulta).list();
51         for(int i=0;i<lista.size();i++){
52             xml = xml + "<"+fila+">";
53             xml = xml + lista.get(i).toString();
54             xml = xml + "</"+fila+">";
55     }

```

Figura 18: Descripción de la clase conexión

Las clases que son mapeadas de la base y que nos permiten instanciar los objetos quedan liberadas de implementar métodos para acceso a la base, es decir no es necesario que cada clase tenga los métodos de transacciones evitando escribir una y otra vez en el mismo código.

4.4.3. Controladores

```
28 protected void processRequest(HttpServletRequest request, HttpServletResponse response)
29 throws ServletException, IOException {
30     String accion=request.getParameter("accion");
31     HttpSession session = request.getSession();
32     if(accion!=null){
33         if(accion.equals("crearCliente")){
34             String descripcion= request.getParameter("descripcion");
35             String ruc= request.getParameter("ruc");
36             String telefono= request.getParameter("telefono");
37             String fax= request.getParameter("fax");
38             String email= request.getParameter("email");
39             String direccion=request.getParameter("direccion");
40             Cliente c= new Cliente();
41             c.setDescription(descripcion);
42             c.setDireccion(direccion);
43             c.setEmail(email);
44             c.setEstado("A");
45             c.setFax(fax);
46             c.setRuc(ruc);
47             c.setTelefono(telefono);
48
49             Conexion conex= new Conexion();
50             conex.ingresar_y_o_actualizar(c);
51             conex.close();
52         }
53         else if(accion.equalsIgnoreCase("consultarClientes")){
54             String descripcion= request.getParameter("descripcion");
55             Conexion conex= new Conexion();
56             String msj=conex.xml("from Cliente where descripcion like '"+descripcion+"'", "cliente");
57
58             conex.close();
59             response.setContentType("application/xml;charset=ISO-8859-1");
60             PrintWriter out = response.getWriter();
61             out.println(msj);
62         }
63     }
64     else if(accion.equalsIgnoreCase("consultarCliente")){
65         String ruc= request.getParameter("ruc");
66         String descripcion = request.getParameter("descripcion");
67         Conexion conex= new Conexion();
68         String msj="";
69         if(ruc==null)ruc="";
70         if(descripcion==null)descripcion="";
71
72         msj=conex.xml("from Cliente where ruc like '"+ruc+"' and descripcion like '"+descripcion+"'", "c
73
74         //JOptionPane.showMessageDialog(null, "entro","Mensaje", JOptionPane.WARNING_MESSAGE);
75         conex.close();
76         response.setContentType("application/xml;charset=ISO-8859-1");
77         PrintWriter out = response.getWriter();
78         out.println(msj);
79         out.close();
80     }
81     else if(accion.equalsIgnoreCase("editar")){
82         String descripcion= request.getParameter("descripcion");
83         String ruc= request.getParameter("ruc");
84         String telefono= request.getParameter("telefono");
85         String fax= request.getParameter("fax");
86         String email= request.getParameter("email");
87         String direccion=request.getParameter("direccion");
88         String id= request.getParameter("id");
89         Conexion conex= new Conexion();
90         Cliente c=(Cliente)conex.objeto(conex.consultar("from Cliente where id="+id));
91         c.setDescription(descripcion);
92         c.setRuc(ruc);
93         c.setDireccion(direccion);
94     }
95 }
```

Figura 19: Descripción del servlet cliente

En la figura 19 se describe como se controlan las acciones que los usuarios realizan en la vista, aquí podemos observar el poder de la tecnología de extracción y manipulación de datos (persistencia), a través del uso de la clase conexión como medio único que implementa los métodos necesarios para el manejo de

transacciones CRUD. Las respuestas que se envían a la vista son datos en formato XML conteniendo la información solicitada por el usuario.

4.4.4. Vistas

En esta sección se explica cómo una vista (páginas web) consume la información en formato XML que recibe desde el servidor. La opción de consultas del sistema en el cual se muestran las sugerencias servirá para esta explicación.

En la figura se muestra una función que realiza las acciones necesarias para mostrar los resultados enviados desde el controlador.

```
123
124
125 function sugerenciaReparacion() {
126     var contexto=$("#contexto").val();
127     var arregloReparacion=["reparacion","descripcion","id","horahombre","costomaterial","tipo"];
128     var miAC=autocompletar(contexto+"/ServletEstimativo", "accion=sugerenciaReparacion", arregloReparacion, "_concept", "miCor
129     miAC.formatResult=function(oResultItem, sQuery) {
130         //var codigo = oResultItem[0];
131         var descripcion = oResultItem[0];
132         var id=oResultItem[1];
133         var horahombre=oResultItem[2];
134         var costomaterial=oResultItem[3];
135         var tipo=oResultItem[4];
136         var sMarkup = "<label title='"+id+"@'+horahombre+'@'+costomaterial+'@'+tipo+'>"+descripcion+"</label>";
137
138         return (sMarkup);|
139     };
140     miAC.itemSelectEvent.subscribe(function(sType, Args){
141         var myAC = Args[0]; // reference back to the AC instance
142         var eLI = Args[1]; // reference to the selected LI element
143         var oData = Args[2]; // object literal of selected item's result data
144         var item=$(eLI).children("label");
145         $("#_hhrefreparacion").attr("value", item.attr("title").split("@")[1]);
146         $("#_costoMaterial").attr("value", item.attr("title").split("@")[2]);
147         $("#_idreparacion").attr("value", item.attr("title").split("@")[0]);
148         $("#_tiporeparacion").attr("value", item.attr("title").split("@")[3]);
149
150     });
151
152 }
```

Figura 20: Detalle de la implementación de las sugerencias

En este segmento de código se define:

Datos que se van a recibir: Descripción, id, hora-hombre, costo-material.

Recurso que nos envía los datos: ServletEstimativo, acción = SugerenciaReparación

Datos que se van a presentar:

```
"<label title="+id+"@"+horahombre+"@"+costomaterial+"@"+tipo+">"+descripcion+"</label>"
```

Esta línea de código nos dice que la información recibida se va a presentar en una lista y cada ítem de la lista va a ser una etiqueta <label>.

Acciones que se realizan con la información:

```
$("#_hhreparacion").attr("value", item.attr("title").split("@")[1]);
```

Usando JQuery vamos a manipular el atributo value de un elemento que lo identificamos mediante su id “_hhreparacion” introduciendo la información recibida del controlador.

4.5. MÓDULO USUARIOS

Este módulo maneja la administración de usuarios que pueden acceder al sistema.

Sirve para ingresar a la base de datos los nombres de usuarios y sus contraseñas, a cada usuario se les asigna funcionalidades específicas del sistema ya que no todas

las personas podrán tener acceso a todas las bondades de la aplicación como se explicó en la descripción de los roles.

Las opciones que este módulo ofrece al usuario son:

- **Ingreso de Usuarios:** El usuario puede acceder a esta opción cada vez que necesite ingresar un nuevo usuario del sistema y asignarle un rol, ya que el sistema maneja dos roles de usuario “Administrador” y “Empleado” cada uno de estos tipos tienen establecidas las opciones a las cuales tiene permiso.
- **Consulta de Usuarios:** El usuario puede consultar una lista de los usuarios ingresados al sistema y el sistema le ofrece una búsqueda personalizada gracias a parámetros de búsqueda establecidos como por ejemplo: Usuario, Rol.
- **Actualización de Usuarios:** Al acceder a esta opción el usuario puede visualizar en una tabla todos los usuarios ingresados, y puede realizar una consulta mediante el criterio de búsqueda “Rol” para encontrar más rápido un registro y gracias a la tecnología AJAX puede presionar un solo ícono para que la información de la tabla se cargue en unas cajas de texto y así poder modificar los datos que considere necesario. En esta opción el usuario puede realizar dos transacciones que son Búsqueda y

Modificación, acciones que gracias a AJAX podemos realizarlas sin hacer una recarga de la página completa.

Editar Usuarios

Seleccione el rol: todos

Consultar

Aquí usted podrá editar los usuarios, elija el usuario a actualizar dando click en "Editar" y luego ingrese la información que ud considere necesaria

Listado de Clientes

Usuario	Contrasena	Rol	Editar
oramirez	oramirez	administrador	[Editar]
gsanchez	gsanchez	administrador	[Editar]
hpenafiel	hpenafiel	empleado	[Editar]
kramirez	kramirez	empleado	[Editar]
oteran	oteran	Empleado	[Editar]
ocarvajal	ocarvajal	Administrador	[Editar]
mvillon	mvillon	Empleado	[Editar]
olopez	olopez	empleado	[Editar]
operez	operez	empleado	[Editar]

Usuario: gsanchez

Rol: Administrador

Contraseña: *****

Actualizar Cancelar

Figura 21: Pantalla del Sistema para Actualizar Usuario

4.6. MÓDULO CLIENTES

Este módulo maneja la administración de clientes de la empresa dueña del sistema.

Sirve para ingresar a la base de datos la información de los clientes.

Las opciones que este módulo ofrece al usuario son:

- **Ingreso de Clientes:** El usuario puede acceder a esta opción cada vez que necesite ingresar un nuevo cliente al sistema, se debe digitar la información solicitada y presionar un botón para guardar los datos. Aquí

se realizan las validaciones necesarias como por ejemplo: información obligatoria, validar que la cantidad de dígitos en el número de teléfono sea la correcta, validar el número de cédula entre otros.

- **Consulta de Clientes:** El usuario puede consultar una lista de los clientes ingresados al sistema ofreciendo una búsqueda personalizada según criterios de búsqueda establecidos como por ejemplo: RUC, Descripción.

Consulta de Clientes

Ingrese el RUC:

Descripción:

Consulta de un Cliente

Aquí usted podrá consultar clientes con respecto a la información que ingrese, sino ingresa ningún dato se mostrarán todos los clientes

Listado de Clientes

RUC	Descripción	Email	Teléfono	Fax	Dirección
1234567890123	KIMTECTH	ange_r18@hotmail.com	042317690	042317690	Carlos Julio Aroseman km 4.5
0943127653111	Bareti	baretisa@hotmail.com	042367401	042367401	Alborada x etapa
0922554372111	Grupo Palmar	grupo-palmar@hotmail.com	042231643	042231643	Guayaces calle 4
0912327490111	ISBELNI S.A.	isbelnisa@hotmail.com	042398754	042398754	Rosa Borja de Icaza y calle 4ta
0942987652111	Transmabo	transmabosa@hotmail.com	042324216	042324217	Francisco de Orellana y otra
0976452391111	Jorge Abudeye	jorge-abudeye@hotmail.com	042687643	042687643	Sauces v y peatonal 10ma
0976564213111	Carvajal s.a.	carvajalsa@hotmail.com	042657320	042657320	Carlos Julio Arosemena y ave 2da
0965473829111	Sanchez	sanchezasociado@hotmail.com	042311452	042311452	Juan Tanga Marengo y ave 5ta

Figura 22: Pantalla del Sistema para Consultar Clientes

- **Actualización de Clientes:** El usuario visualiza en una tabla los clientes ingresados, pero también puede realizar una consultar para encontrar un cliente de una manera más rápida ingresando el “RUC” o la

“Descripción” y luego con solo dar un clic sobre un icono, los datos se cargan en unas cajas de texto para poder ser modificados, todo esto usando AJAX es decir que en ningún instante se observa que se recarga la página completa.

4.7. MÓDULO CONTENEDORES

Este módulo maneja la administración de contenedores que ingresan a los talleres de la empresa para ser reparados.

Las opciones que este módulo ofrece al usuario son:

- **Ingreso de Contenedores:** El usuario puede acceder a esta opción cada vez que necesite ingresar la información de un contenedor que va a ingresar al taller, se debe digitar la información solicitada y presionar un botón para guardar los datos.
- **Consulta de Contenedores:** El usuario puede consultar una lista de contenedores ingresados al sistema ofreciendo una búsqueda personalizada según criterios de búsqueda establecidos como por ejemplo: código, descripción, cliente.

Usando AJAX pudimos implementar una búsqueda con “Sugerencia” es decir a medida que el usuario va digitando letras de algún nombre se

despliega una listas de coincidencias, cabe recalcar que la lista desplegada es una consulta directa a la base de datos que trae los registro que contengan las letras digitadas.

- **Actualización de Contenedores:** Para actualizar la información de un contenedor el sistema nos ofrece una búsqueda rápida mediante criterios establecidos y en el campo cliente que es uno de los criterios mencionados se presenta “Sugerencias” y para modificar los datos solo basta con dar clic al icono de edición y los datos se ubican en cajas de textos listos para ser modificados. Estas acciones no realizan una recarga de la página ya que estamos usando tecnología AJAX.

Actualizar Contenedor

Cliente: la información que ud considere necesaria.

Listado de Reparaciones

Código	Descripción	Size	Propietario	Editar
A001	Contenedor1	22	KIMTECTH	
A002	Contenedor2	31	KIMTECTH	
A003	ContendorA	24	Transmabo	
A011	ContenedorB	32	Bareti	
GP001	Contenedor 1	17	Grupo Palmar	
A004	Contenedor Big	32	KIMTECTH	
A005	Contenedor Medium	18	Bareti	
A006	Contenedor Big	35	ISBELNI S.A.	
S001	Contenedor small	15	Transmabo	
B001	Contenedor Big	43	Jorge Abudeye	
B002	Contenedor Medium	25	Jorge Abudeye	

Figura 23: Pantalla del Sistema para Actualizar Contenedor

4.8. MÓDULO REPARACIONES

Este módulo maneja la administración de reparaciones que se realizan a un contenedor , puesto que las reparaciones suelen ser las mismas para uno y otro contenedor el usuario puede almacenar estas reparaciones para luego poder consultarlas y elaborar el estimativo de una manera más eficiente disminuyendo el consumo de tiempo y recursos.

Las opciones que este módulo ofrece al usuario son:

- **Ingreso de Reparaciones:** El usuario puede acceder a esta opción cada vez que necesite ingresar la información de una reparación, se debe digitar la información solicitada y presionar un botón para guardar los datos. El sistema valida la información digitada como por ejemplo enviando alertas o mensajes cuando el usuario digita una letra en algún campo que solicita números.
- **Consulta de Reparaciones:** El usuario puede consultar una lista de reparaciones ingresadas al sistema ofreciendo una búsqueda personalizada según criterios establecidos como por ejemplo: código, descripción, tipo.

Usando AJAX pudimos implementar una búsqueda de la información sin hacer una recarga de la página.

Consulta de Reparaciones

Ingrese el código: 

Descripción:

Tipo: Todos ▾

Consulta de una Reparación

Aquí usted podrá consultar las reparaciones con respecto a la información que ingrese, sino ingresa ningún dato se mostrarán todas las reparaciones.

Listado de Reparaciones

Código	Descripción	Tipo	Costo hora hombre(\$)	Costo material(\$)
TA008	Replace sensor compresor SABROE 414300	maquinaria	20.00	160.00
TA0307	Soldar fuga evaporador	maquinaria	1.50	200.00
TD032	Replace Tunnel Rail	maquinaria	8.50	240.00
TD033	Replace Tunnel Assy	maquinaria	10.00	612.00
TD012	INSERT FULL DOOR HEADER	estructura	7.78	76.40

Figura 24: Pantalla del Sistema para Consultar Reparaciones

- **Actualización de Reparaciones:** Para actualizar la información de una reparación podemos realizar una búsqueda rápida mediante criterios establecidos, luego de encontrar la información damos clic al icono de edición y los datos se cargan en cajas de textos para poder hacer modificaciones. Esta acción no realizan una recarga de la página ya que estamos usando tecnología AJAX.

4.9. MÓDULO ESTIMATIVOS

Este módulo se encarga de la gestión de reparaciones hechas a un contenedor, se registra datos del cliente que solicita la reparación, datos del contenedor y un detalle de las reparaciones hechas como por ejemplo descripción del daño y costos.

Las opciones que este módulo ofrece al usuario son:

- **Ingreso de Estimativos** El usuario accede a esta opción para ingresar un estimativo es decir un detalle de las reparaciones hechas a un contenedor.

Gracias al uso de AJAX el sistema le ofrece al usuario “Sugerencias” en cajas de texto como por ejemplo en el campo “Cliente” es decir a medida que el usuario va digitando letras se despliega una lista de nombres que coinciden con los dígitos ingresados y que proviene de una consulta directa a la base de datos.

Para ingresar la fecha el usuario no necesita digitarla puesto que si coloca el cursor sobre la caja de texto se presenta un “calendario” procediendo a elegir la fecha deseada.

- **Consulta de Estimativos:** El usuario puede consultar una lista de estimativos ingresados al sistema ofreciendo una búsqueda personalizada según criterios de búsqueda establecidos como por ejemplo: Número de estimativo, cliente, contenedor.

Cliente: Este campo ofrece al usuario “Sugerencias” es decir que a medida que va digitando letras se realiza una consulta directa a la base de datos para desplegar una lista con los registros que contienen las letras digitadas. Esto lo podemos realizar usando AJAX.

Contenedor: Puesto que cada cliente puede tener uno o más contenedores este campo también ofrece “Sugerencias” luego de ingresar un cliente el usuario podrá visualizar una lista de contenedores a medida que va ingresando la identificación del contenedor, dicha lista se presenta gracias a una consulta directa a la base de datos usando AJAX. El usuario puede obtener una “Vista previa” de los estimativos ya que la lista que se muestra contiene datos básicos del estimativo usamos “Drag & Drop” para que el usuario pueda tomar el registro y soltarlo en un recuadro para que en la misma página se pueda observar el estimativo completo es decir con todos sus datos.

No.Estimativo	Cliente	Cod.Contenedor	Fecha	Subtotal	Iva	Total
1	Grupo Palmar	GP001	2010-08-08	1610.94	10.00	1772.03
2	Bareti	A003	2010-08-08	424.94	0.00	424.94
3	KIMTECTH	A002	2010-07-22	668.00	12.00	748.16
4	KIMTECTH	A004	2010-08-12	708.00	10.00	778.80
5	ISBELNI S.A.	A006	2010-06-20	966.64	0.00	966.64
6	Bareti	A005	2010-08-15	1268.88	12.00	1421.15

Figura 25: Uso del Drag & Drop para generar la vista previa del Estimativo

- **Actualización de Estimativos:** Para actualizar la información de un estimativo podemos realizar una búsqueda rápida mediante los criterios establecidos como el número de estimativo, el cliente o por el código del contenedor, luego de encontrar la información damos clic al número del

estimativo y los datos se ubican en una nueva página similar a la de Ingreso de Estimativos.

No. Estimativo	Cliente	Cod. Contenedor	Fecha	Subtotal	Iva	Total
<u>1</u>	Grupo Palmar	GP001	2010-08-08	1610.94	10.00	1772.03
<u>2</u>	Bareti	A003	2010-08-08	424.94	0.00	424.94
<u>3</u>	KIMTECTH	A002	2010-07-22	668.00	12.00	748.16
<u>4</u>	KIMTECTH	A004	2010-08-12	708.00	10.00	778.80
<u>5</u>	ISBELNI S.A.	A006	2010-06-20	966.64	0.00	966.64
<u>6</u>	Bareti	A005	2010-08-15	1268.88	12.00	1421.15

Figura 26: Elegir el Estimativo mediante su número

Línea: Ubicación:

Contenedor: Costo Hora Hombre:(S)

Tamaño: E.I.R #:

M/N: Fecha:

Supervisor:

Detalles

Concepto: H.H.:(S) Material:(S)

Locación: Daños: Cantidad:

Location	Descripción	Tipo	Damage	Cantidad	CostoHora	CostoLabores	CostoMat.	Elimin.
LEFT	Replace sensor compresor SARBCE	maquinaria	MISSING	2	42.00	336.00	160.00	-

Figura 27: Datos cargados para actualizar los Estimativos

En la tabla de Detalles de reparaciones podemos usar “Drag & Drop” es decir podemos tomar un ítem y arrastrarlo hacia la parte superior (Figura 28) donde se encuentran las cajas de texto soltamos el ítem y los

datos se colocan en sus respectivas cajas para poder realizar los cambios.

Detalles

RIGHT Replace sensor compresor SABROE 414300 maquinaria H.MISSING 2 42.00 336.00 160.00

Locación: LEFT Daños: MISSING Cantidad:

Agregar

Location	Descripción	Tipo	Damage	Cantidad	CostoHora	CostoLabores	CostoMat.	Elimin.
RIGHT	Replace sensor compresor SABROE 414300	maquinaria	MISSING	2	42.00	336.00	160.00	
LEFT	Soldar fuga evaporador	maquinaria	MISSING	2	3.00	24.00	200.00	
LEFT	Replace Tunnel Assy	maquinaria	MISSING	1	10.00	80.00	610.00	

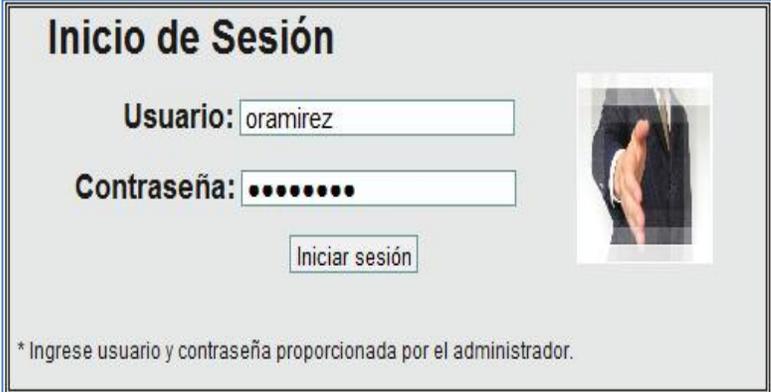
Figura 28: Pantalla con la tabla de detalles del Estimativo

4.10.SEGURIDAD

Al desarrollar una aplicación web es de vital importancia tener en cuenta las posibles amenazas a las que el sistema está expuesto. Por esa razón es necesario el manejo de sesiones y la asignación de roles a cada usuario, para evitar el ingreso de personas no autorizados al sistema.

4.5.1. Sesiones

Es el método de seguridad usado en el sistema. Cada vez que un usuario ingresa al sistema debe colocar su usuario y su contraseña en la página de login.



The image shows a login form with the following elements:

- Inicio de Sesión**: Title of the form.
- Usuario:** Input field containing the text "oramirez".
- Contraseña:** Input field containing seven black dots.
- Iniciar sesión**: A button to submit the login information.
- * Ingrese usuario y contraseña proporcionada por el administrador.**: A note at the bottom of the form.
- Image:** A small photograph of a person's back and shoulder, located to the right of the password field.

Figura 29: Inicio de sesión del Sistema

Si el usuario se encuentra registrado en la base de datos, se devuelve un objeto de tipo usuario, el cual contiene su nombre, apellido, usuario, contraseña y tipo de usuario, crea una sesión y redirige al usuario a la página principal del sistema.

Por el contrario, si el usuario no se encuentra registrado en la base de datos e intenta ingresar al sistema, le aparecerá un mensaje de error indicándole que no se encuentra registrado.

4.5.2. Filtro para el inicio de sesión

```
106 public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
107     throws IOException, ServletException {
108     HttpServletRequest req=(HttpServletRequest)request;
109     HttpServletResponse res=(HttpServletResponse)response;
110     String usuario=req.getParameter("txtusuario");
111     String password=req.getParameter("txtpassword");
112     String accion=req.getParameter("accion");
113     HttpSession sesion = req.getSession();
114     Usuario usr=(Usuario)sesion.getAttribute("usuario");
115     if(usuario==null && usr==null && password==null){
116         res.sendRedirect(req.getContextPath()+"/index.jsp");
117     }
118     else if (usr == null){
119         Usuario nuevo = new Usuario();
120         Conexion con = new Conexion();
121         nuevo = (Usuario)con.objeto(con.consultar("from Usuario where usr='"+usuario+"' and pwd='"+password+"'"));
122         if (nuevo == null)
123             res.sendRedirect(req.getContextPath()+"/index.jsp");
124         else{
125             sesion.setAttribute("usuario", nuevo);
126             sesion.setMaxInactiveInterval(-1);
127             ArrayList menu=ConamiMenu.crearMenu(nuevo);
128             sesion.setAttribute("menu", menu);
129         }
130         chain.doFilter(req,res);
131     }else {
132         if (accion != null){
133             if (accion.equalsIgnoreCase("logout")){
134                 sesion.invalidate();
135                 res.sendRedirect(req.getContextPath()+"/index.jsp");
136                 return;
137             }
138         }
139         chain.doFilter(req,res);
140     }
141 }
```

Figura 30: Detalle de la implementación del login filtro

En este filtro en primer lugar obtenemos los parámetros “Usuario” y “Password” enviados desde la página y validamos que existan, si uno de los dos valores no son enviados se envía a la página de inicio.

Una vez que se haya comprobado que los parámetros son recibidos correctamente verificamos si el usuario existe en la base de datos, si el usuario está ingresado en la base le damos acceso al sistema caso contrario lo enviamos a la página de inicio.

Cuando el usuario ha pedido abandonar el sistema (logout) se invalida la sesión del usuario y se lo direcciona a la página de inicio.

4.5.2. Filtro para el acceso a las páginas

```
99  */
100  public void doFilter(ServletRequest request, ServletResponse response,
101                      FilterChain chain)
102  throws IOException, ServletException {
103      HttpServletRequest req=(HttpServletRequest)request;
104      HttpServletResponse res=(HttpServletResponse)response;
105      String accion=req.getParameter("accion");
106      String path=req.getRequestURI();
107      HttpSession sesion = req.getSession();
108      Usuario usr=(Usuario)sesion.getAttribute("usuario");
109      if(usr==null){
110          res.sendRedirect(req.getContextPath()+"/index.jsp");
111      }
112      else if(path.contains("empleado")){
113          if(usr.getRol().equalsIgnoreCase("administrador")||usr.getRol().equalsIgnoreCase("empleado")){
114              chain.doFilter(req, res);
115          }
116          else{
117              res.sendRedirect(req.getContextPath()+"/index.jsp");
118          }
119      }
120      else if(path.contains("admin")){
121          if(usr.getRol().equalsIgnoreCase("administrador")){
122              chain.doFilter(req, res);
123          }
124          else{
125              sesion.setAttribute("cambiarUsuario", true);
126              sesion.setAttribute("ruta",path);
127              res.sendRedirect(req.getContextPath()+"/index.jsp");
128          }
129      }
130  }
```

Figura 31: Detalle de la implementación del filtro páginas

En este filtro damos permiso a los usuarios para acceder a las páginas.

Cada usuario tiene asignado un rol, según esta característica se permite acceder a las carpetas que administran las páginas web, los nombres de dichas carpetas son “Empleado” y “Admin”. En este filtro lo que se hace es preguntar si en la URI se encuentra el nombre de alguna de las dos carpetas, finalmente que hemos identificado a que carpeta se quiere acceder preguntamos si el rol del usuario tiene permiso para acceder a estos recursos.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. Hoy en día uno de los puntos claves en el análisis y diseño de software es el tiempo de desarrollo; es decir, el tiempo en el que el producto estará listo para salir a producción, los desarrolladores luchan por disminuir este tiempo. Actualmente una de las herramientas para lograr este objetivo es el uso de tecnologías como las que se presentan en este proyecto: Persistencia- Hibernate y el framework YUI, este último ofrece código estandarizado plasmado en funciones que pueden ser fácilmente utilizadas y personalizadas.
2. El análisis de la aplicación, para la Empresa Conami S.A., da solución, en gran medida, a muchos de los problemas que actualmente esta Empresa presenta en lo que al manejo de la información de los Estimativos respecta, permitiéndole a quienes allí laboran poder acceder a ésta de manera más rápida, eficiente y confiable. Si bien es cierto que existen en la Empresa algunos elementos que podrían facilitar la utilización del sistema debe reconocerse que en lo que a los usuarios respecta, podría obtenerse una actitud negativa con miras al cambio;

pero con una buena explicación de los beneficios y mejoras que ofrecemos con el sistema todos saldremos favorecidos.

3. El uso de AJAX permite mejorar el tiempo de respuestas de una aplicación web, pero cabe recalcar que cuando se realiza la primera carga del proyecto, el usuario va a percibir una respuesta un poco lenta a la solicitud requerida; esto se debe a que Hibernate detecta o reconoce todas las tablas en la base de datos a través del mapeo, representándola de forma remota tomando un tiempo prudencial para realizar esto.
4. Administrativamente nuestra aplicación ofrece la facilidad de generar reportes en cuanto a la cantidad de Estimativos que se realice a cada contenedor que ingrese a la empresa, así como las actividades de los clientes y usuarios que trabajan dentro de la institución.
5. Con la aplicación desarrollada se reduce el tiempo en la elaboración de los Estimativos; si antes teniendo los datos y elaborándolos en hojas de Excel a un empleado de Conami le tomaba de 10 a 15 minutos, esta tarea con el uso de la

aplicación se reduce de 3 a 4, mejorando de esta manera la eficiencia y calidad del servicio ofrecido.

RECOMENDACIONES

1. Una vez que el sistema se encuentre operando en la empresa el administrador será el encargado de manipular a los usuarios (entendiendo a estos como los empleados de la empresa) es recomendable que dicho administrador sepa de su responsabilidad y tome conciencia de su rol en la manipulación de la aplicación ya que no se le puede dar todos los privilegios a cualquier empleado ya que la información que se maneja es de vital importancia para la empresa.
2. Para el manejo de las distintas funcionalidades de la aplicación es necesario la capacitación y adiestramiento al personal que va estar a cargo de la manipulación del sistema.
3. Capacitar al personal que labora en las dependencias de Conami S.A en el manejo de las herramientas básicas para la correcta ejecución de la aplicación.

4. Realizar un mantenimiento preventivo y si es el caso correctivo en las máquinas que se encuentran en la empresa para que se adapten a los requerimientos propios de la aplicación.

5. Se sugiere realizar un análisis periódico del sistema durante los primeros tres meses a fin de determinar posibles problemas que no se hayan identificado a tiempo y que en el futuro puedan representar impedimentos potenciales para el funcionamiento y desarrollo del mismo.

6. En un futuro puesto que Conami S.A. es una institución con visiones claras de ampliar su infraestructura, aumentar la automatización en el número de servicios que ofrecen, para mejorar la calidad de la empresa; se deberá realizar una revisión del sistema ya que se pueden incluir mencionados servicios y así evitar incurrir en mayores gastos.

GLOSARIO

API

Application Programming Interface, es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

ARQUITECTURA CLIENTE/SERVIDOR

Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta.

BEAN

Es un componente software que tiene la particularidad de ser reutilizable y así evitar la tediosa tarea de programar los distintos componentes uno a uno. Se puede decir que existen con la finalidad de ahorrarnos tiempo al programar.

CLASES DE JAVA

Una clase es una agrupación de datos (variables o campos) y de funciones (métodos) que operan sobre esos datos.

CSS

Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

DAO

Objeto de Acceso a Datos, es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo.

ESCALABILIDAD

Capacidad de un software o de un hardware de crecer, adaptándose a nuevos requisitos conforme cambian las necesidades del negocio.

FILTROS

Un filtro es un proceso que se aplica a los datos que se reciben o se envían por el servidor. Los datos enviados por los clientes al servidor son procesados por filtros de entrada mientras que los datos enviados por el servidor se procesan por los filtros de salida.

FRAMEWORK

Un framework es la extensión de un lenguaje mediante una o más jerarquías de

clases que implementan una funcionalidad y que (opcionalmente) pueden ser extendidas.

JAVASCRIPT

Es un lenguaje de programación del lado del cliente, utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web.

MYSQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario.

QUERY

Este término generalmente se utiliza para hacer referencia a una interacción con una base de datos. Es la parte de una URL que contiene los datos que deben pasar a aplicaciones web.

SERVLETS

Un servlet es un programa ejecutado en el servidor. Reciben peticiones y mandan resultados en HTTP, siendo el formato más común de salida una página HTML o un archivo XML.

REFERENCIAS BIBLIOGRÁFICAS

- (1)** CONAMI S.A., Quienes Somos, [http:// conami-sa.com/conami-sa/](http://conami-sa.com/conami-sa/), [En línea] [Citado el: 13 de Septiembre del 2010].
- (2)** PONTIFICIA UNIVERSIDAD CATÓLICA DE PERÚ, Desarrollo Web, <http://dia.pucp.edu.pe/portal/content/view/142/942/>, [En línea] [Citado el: 13 de Septiembre del 2010].
- (3)** MICROSOFT, Model-View-Controller, <http://msdn.microsoft.com/en-us/library/ff649643.aspx>, [En línea] [Citado el: 14 de Septiembre del 2010].
- (4)** WIKIPEDIA: Orchard Leslie M, Prototype JavaScript Framework, http://en.wikipedia.org/wiki/Prototype_JavaScript_Framework, [En línea] [Citado el: 17 de Agosto de 2010].

- (5) WIKIPEDIA, AJAX (programming),
http://en.wikipedia.org/wiki/Ajax_%28programming%29, [En línea]
[Citado el: 9 de Agosto de 2010].
- (6) HIBERNATE, Relational Persistence for Java and .NET,
<http://www.hibernate.org/>, [En línea] [Citado el: 14 de Agosto de 2010].
- (7) NETBEANS Y MYSQL, Herramientas en el Desarrollo de la Ingeniería de Software,
<http://netbeans.org/community/releases/69/>;
<http://dev.mysql.com/doc/query-browser/es/mysql-query-browser-introduction.html>, [En línea] [Citado el: 24 de Septiembre de 2010].
- (8) SLIDESHARE Ing. Ana María Pacheco, Arquitectura básica en la Ingeniería de Software, <http://www.slideshare.net/siis/struts-en-java>,
[En línea] [Citado el: 24 de Septiembre de 2010].
- (9) WIKIPEDIA, YUI Library (Yahoo User Interface),
http://es.wikipedia.org/wiki/Yahoo!_UI_Library, [En línea] [Citado el:
24 de Septiembre de 2010].

(10) GARRETT, JESSE JAMES, AJAX: A New Approach to Web Applications,
<http://adaptivepath.com/ideas/essays/archives/000385.php>, [En línea]
[Citado el: 12 de Agosto de 2010].

(11) THE JQUERY PROJECT, JQuery Framework de JavaScript,
<http://jquery.com>, [En línea] [Citado el: 21 de Agosto de 2010].