



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

**“DESARROLLO DE UNA APLICACIÓN PARA EL SERVICIO DE
ENCOMIENDAS Y VENTA DE BOLETOS, DE LA COOPERATIVA
DE TRANSPORTE “CIUDAD DE SUCUA” BAJO LA NORMA DE
CALIDAD ISO/IEC 9126-2”**

TRABAJO DE TITULACIÓN: **PROYECTO TÉCNICO**

Para optar el Grado Académico de:

INGENIERO EN SISTEMAS INFORMÁTICOS

AUTOR: DARÍO JAVIER JIMÉNEZ CHILÁN

TUTOR: ING. JONNY ISRAEL GUAÍÑA YUNGAN

Riobamba – Ecuador

2017

©2017, DARÍO JAVIER JIMÉNEZ CHILÁN

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

El Tribunal del Trabajo de Titulación certifica que: El trabajo técnico: “DESARROLLO DE UNA APLICACIÓN PARA EL SERVICIO DE ENCOMIENDAS Y VENTA DE BOLETOS, DE LA COOPERATIVA DE TRANSPORTE “CIUDAD DE SUCUA” BAJO LA NORMA DE CALIDAD ISO/IEC 9126-2”, de responsabilidad del señor Darío Javier Jiménez Chilán, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

NOMBRE	FIRMA	FECHA
Ing. Washington Luna Encalada DECANO FACULTAD INFORMATICA Y ELECTRONICA	_____	_____
Ing. Patricio Moreno Costales DIRECTOR ESCUELA INGENIERIA EN SISTEMAS	_____	_____
Ing. Jonny Guaiña Yungan DIRECTOR DE TRABAJO DE TITULACIÓN	_____	_____
Dr. Omar Gómez Gómez MIEMBRO DEL TRIBUNAL	_____	_____

Yo, Darío Javier Jiménez Chilán soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación, y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo.

Darío Javier Jiménez Chilán

DEDICATORIA

El presente trabajo de titulación está dedicado a Dios quien estuvo conmigo durante estos años lejos de mi familia, a mis padres quienes fueron y son pilares fundamentales en mi vida y gracias al apoyo de ellos he sido capaz de lograr esta meta, a mi hermana quien a su corta edad me enseñó que esta vida está llena de retos y hay que seguir adelante por más difícil que se vea el camino, a mis compañeros, a los docentes que me ayudaron a culminar esta etapa; a todos les agradezco de por siempre estar conmigo cuando lo necesitaba.

Darío

AGRADECIMIENTO

Agradecer primeramente a Dios por la salud prestada y fuerzas para culminar el proyecto; también agradecer a todas aquellas personas que, de alguna forma, son parte de su culminación. Mi sincero agradecimiento está dirigido hacia ingeniera Adriana Cárdenas gerente de la cooperativa de transporte terrestre “Ciudad de Sucua”, quien, con su ayuda desinteresada, brindó información relevante, próxima, pero muy cercana a la realidad para el desarrollo del proyecto. A todas las secretarias que ayudaron con las necesidades en cada oficina, y a todos los que conforman la cooperativa de transporte “Ciudad de Sucua” por el apoyo brindado.

Darío

TABLA DE CONTENIDO

TABLA DE CONTENIDO.....	vii
ÍNDICE DE TABLAS.....	xi
INDICE DE FIGURAS.....	xii
RESUMEN.....	xiv
SUMMARY	xv
INTRODUCCIÓN	1
CAPÍTULO I	
1. MARCO TEÓRICO REFERENCIAL.....	10
1.1 Metodología scrum.....	10
<i>1.1.1 Elementos Scrum.....</i>	<i>10</i>
<i>1.1.1.1 Producto Backlog o Pila del Producto</i>	<i>10</i>
<i>1.1.1.2 Sprint Backlog.....</i>	<i>11</i>
<i>1.1.1.3 Incremento funcional potencialmente entregable</i>	<i>11</i>
<i>1.1.1.4 Historia de Usuario.....</i>	<i>11</i>
<i>1.1.1.5 Ciclo de Vida.....</i>	<i>12</i>
1.2 Postgresql	13
<i>1.2.1 Definición</i>	<i>13</i>
<i>1.2.2 Disparadores (triggers)</i>	<i>13</i>
1.3 Framework	13
<i>1.3.1 Bootstrap.....</i>	<i>13</i>
1.4 Norma iso/iec 9126	14
<i>1.4.1 Definición</i>	<i>14</i>
<i>1.4.1.1 Características de la norma iso/iec 9126</i>	<i>15</i>
<i>1.4.2 iso/iec 9126-2.....</i>	<i>16</i>
<i>1.4.2.1 Métrica de comprensibilidad</i>	<i>17</i>
<i>1.4.2.3 Métrica de aprendizabilidad.....</i>	<i>17</i>

1.4.2.4	<i>Métricas de operabilidad</i>	18
1.4.2.5	<i>Métrica de atractivo</i>	18
1.4.2.6	<i>Métricas de cumplimiento de usabilidad</i>	18
1.5	Usabilidad	18
1.5.1	<i>Comprensibilidad</i>	19
1.5.2	<i>Facilidad de Aprender</i>	19
1.5.3	<i>Operabilidad</i>	19
1.5.4	<i>Manejo de Errores</i>	19
1.5.5	<i>Presentación visual apropiada</i>	19
1.6	Analítica web	20
1.6.1	<i>¿Qué mide la analítica web?</i>	20
1.7	Crazy egg	20
1.8	Sistemas de transporte	21
1.9	Algoritmo de encriptación md5	21
1.10	Java server pages (JSP)	22
1.11	Servlets	22
CAPÍTULO II		
2	MARCO METODOLOGICO	24
2.1	Introducción	24
2.2	Fase de planificación	24
2.2.1	<i>Análisis y recolección de información</i>	25
2.2.2	<i>Módulo Gestión de Boletos</i>	26
2.2.3	<i>Módulo Gestión de encomiendas</i>	26
2.2.4	<i>Roles del proyecto</i>	26
2.2.5	<i>Product Backlog</i>	26
2.2.6	<i>Presupuesto para el desarrollo</i>	27
2.2.6.1	<i>Costo de Software</i>	27
2.2.6.2	<i>Costo de Hardware</i>	27
2.2.6.3	<i>Costo de Varios</i>	28

2.3	Fase de diseño	29
2.3.1	Diagramas de Casos de Uso.....	29
2.3.1.1	Caso de uso venta de boletos	29
2.3.1.2	Caso de uso gestión encomiendas.....	30
2.3.3	Diagramas de Despliegue.....	31
2.3.4	Arquitectura del Sistema.....	31
2.3.5	Estándar de Codificación.....	32
2.3.6	Interfaz de Usuario	32
2.3.6.1	Interfaz Principal	33
2.3.6.2	Interfaz Login.....	33
2.3.6.3	Interfaz Empleado-Administrador	34
2.3.6.4	Interfaz Empleado-Secretario/a.....	34
2.3.6.5	Interfaz Cliente.....	35
2.3.7	Diseño Base de datos.....	35
2.4	Fase de desarrollo.....	38
2.4.1	Sprint Backlog (Lista de sprint).....	38
2.4.2	Historias de Usuarios.....	39
2.4.3	Pruebas de aceptación.....	40
2.4.4	Tareas de ingeniería.....	41
2.4.5	Gráfica de Producto (Burn Up).....	42
2.4.6	Implementación del algoritmo de encriptación md5.....	42
CAPÍTULO III		
3	MARCO DE RESULTADOS.....	45
3.1	Introducción	45
3.2	Determinación de usabilidad bajo la norma iso 9126-2.....	45
3.2.1	Métricas para la usabilidad de software.....	45
3.2.1.1	Muestra	46
3.2.1.2	Datos Obtenidos.....	47
3.2.1.3	Análisis de Usabilidad mediante atributos propuestos.....	53

3.2.2	<i>Análisis de usabilidad mediante la herramienta de crazy eggs</i>	55
3.2.2.1	<i>Análisis HeatMap</i>	55
3.2.2.2	<i>Análisis Scrollmap</i>	57
3.2.2.3	<i>Análisis Overlay</i>	59
	CONCLUSIONES	62
	RECOMENDACIONES	63
	BIBLIOGRAFIA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-2:	Desarrollo reuniones.....	25
Tabla 2-2:	Roles y personal encargados del proyecto.....	26
Tabla 3-2:	Product backlog.....	26
Tabla 4-2:	Detalle costos en software.....	27
Tabla 5-2:	Detalle costo de hardware.....	27
Tabla 6-2:	Detalle de costos varios.....	28
Tabla 7-2:	Sprint Backlog.....	38
Tabla 8-2:	Detalle del Sprint 1.....	39
Tabla 9-2:	Historias de Usuarios.....	40
Tabla 10-2:	Detalle de Prueba de aceptación.....	40
Tabla 11-2:	Detalle tarea de ingeniería.....	41
Tabla 1-3:	Detalle de preguntas Facilidad de aprendizaje.....	47
Tabla 2-3:	Detalle de preguntas Tiempo de respuesta.....	48
Tabla 3-3:	Detalle de preguntas Entendibilidad.....	50
Tabla 4-3:	Detalle de preguntas Atractivo.....	51
Tabla 5-3:	Detalle de Atributos Usabilidad.....	53

INDICE DE FIGURAS

Figura 1-1: Descripción del modelo MVC	7
Figura 2-1: Características de la norma ISO 9126.....	15
Figura 3-1: Esquema de aplicación usando Servlets y JSP.....	23
Figura 1-2: Diagrama de Actividades	24
Figura 2-2: Diagrama, caso uso Venta de boletos Sistema “SoftSucua”.....	29
Figura 3-2: Diagrama, caso uso gestión de encomiendas Sistema “SoftSucua”	30
Figura 4-2: Diagrama de clases Sistema “SoftSucua”	30
Figura 5-2: Diagrama de despliegue Sistema “SoftSucua”	31
Figura 6-2: Arquitectura del Sistema Sistema “SoftSucua”	32
Figura 7-2: Página principal Sistema “SoftSucua”	33
Figura 8-2: Página login, Sistema “SoftSucua”	34
Figura 9-2: Página Administrador, Sistema “SoftSucua”	34
Figura 10-2: Página Secretario/a, Sistema “SoftSucua”	35
Figura 11-2: Página Cliente, Sistema “SoftSucua”.....	35
Figura 12-2: Modelo DER, Sistema “SoftSucua”.....	37
Figura 13-2: Burn up, Sistema “SoftSucua”	42
Figura 14-2: Algoritmo Encriptación, Sistema “SoftSucua”	43
Figura 15-2: Algoritmo descriptacion datos, Sistema "SoftSucua"	43
Figura 16-2: Tabla postgresql empleados, Sistema "SoftSucua"	44
Figura 1-3: Resultados HeatMap, Sistema “SoftSucua”.....	56
Figura 2-3: Resultados Scrollmap, Sistema “SoftSucua”	58
Figura 3-3: Resultados Overlay, Sistema “SoftSucua”	60

INDICE DE GRÁFICOS

Gráfico 1-3: Facilidad de aprendizaje, Sistema “SoftSucua”	48
Gráfico 2-3: Tiempo de respuesta, Sistema “SoftSucua”	50
Gráfico 3-3: Entendibilidad, Sistema “SoftSucua”	51
Gráfico 4-3: Atractivo, Sistema “SoftSucua”	53
Gráfico 5-3: Usabilidad, Sistema “SoftSucua”	54

RESUMEN

El principal objetivo del proyecto fue desarrollar un sistema de servicio de encomiendas y ventas de boletos para la cooperativa de transporte interprovincial “Ciudad de Sucua” aplicando la norma de calidad ISO/IEC 9126-2. Para la determinar el problema se utilizó como técnica de estudio la entrevista, para el desarrollo del sistema web se utilizó la metodología ágil SCRUM que permitió un adecuado entendimiento entre el cliente y el equipo de desarrollo del sistema, como herramienta para realizar la programación del sistema se utilizó JAVA, como entorno de desarrollo integrado (IDE) se usó NETBEANS, como servidor web GLASSFISH y como framework para crear las interfaces adaptables se utilizó BOOTSTRAP; en el proyecto también se implementaron las librerías iText para generar reportes en PDF y la librería commons-codec el para implementar el algoritmo de encriptación, para los controladores se usó Servlet y adicionalmente se utilizó Ajax y JQUERY. A la culminación del sistema se obtuvo un total de 87580 líneas de código fuente del sistema, con 240 archivos jsp, 20 clases, 23 controladores y 14 archivos de hoja de estilo, la aplicación tiene un tamaño de 45.6 MB en disco. Para determinar si la aplicación cumple la norma de calidad ISO/IEC 9126 se realizó una encuesta con atributos de la norma, se utilizó la herramienta online crazy eggs con la que se obtuvo un porcentaje de 97.48% de aceptación de los encuestados. A partir de los resultados se concluye que el sistema satisface las necesidades del usuario brindando un software listo para producción. Se recomienda la incorporación de un servidor propio para la empresa que implemente el sistema.

PALABRAS CLAVE: <ISO/IEC 9126> <METODOLOGÍA DE DESARROLLO (SCRUM)>
<LENGUAJE DE PROGRAMACION (JAVA)> <CRAZY EGGS> <USABILIDAD>
<SERVICIO DE ENCOMIENDAS Y VENTAS DE BOLETOS>

SUMMARY

The main objective of the project was to develop a system of parcel service and ticket sales for the interprovincial transport cooperative "Ciudad de Sucua" applying the quality standard ISO / IEC 9126-2. To determine the problem, it was used as an interview study technique, for the development of the web system the agile SCRUM methodology was used, which allowed an adequate understanding between the client and the system development team, as a tool to carry out the programming of the system JAVA was used, as an integrated development environment (IDE), NETBEANS was used, as a GLASSFISH web server, and BOOTSTRAP was used as a framework to create the adaptable interfaces; in the project iText libraries were also implemented to generate reports in PDF and the commons-codec library to implement the encryption algorithm, Servlet was used for the controllers and Ajax and JQUERY were used in addition. At the end of the system a total of 87580 lines of system source code was obtained, with 240 jsp files, 20 classes, 23 controllers and 14 style sheet files, the application has a size of 45.6 MB on disk. To determine if the application meets the ISO / IEC 9126 quality standard, a survey was carried out with attributes of the standard, using the online tool crazy eggs with which a percentage of 97.48% of the respondents was obtained. From the results it is concluded that the system meets the needs of the user by providing software ready for production. It is recommended; the incorporation of an own server for the company that implemented the system.

KEY WORDS: <TECHNOLOGY AND SCIENCE OF ENGINEERING>,SOFTWARE ENGINEERING>, <SOFTWARE QUALITY>, <DEVELOPMENT METHODOLOGY (SCRUM)> <JAVA (SOFTWARE)> <CRAZY EGGS (SOFTWARE)> <USABILITY> <PROCESS AUTOMATION>

INTRODUCCIÓN

ANTECEDENTES

Al ver el desarrollo y el crecimiento tecnológico de las TICs se ha incrementado la necesidad de automatizar procesos para ahorrar tiempo, dinero, recursos humanos, al notarse esta evolución en la tecnología las empresas se han visto la necesidad de integrarse al mundo virtual para satisfacer a sus clientes y se adapten a sus necesidades.

La cooperativa “Ciudad de Sucua” es una empresa de transporte público interprovincial creada el año “1990” con permiso de operaciones número “0058-2016” con fecha “06 de abril de 2016” cuyo fin es servir a la provincia de Morona Santiago y sus alrededores sin distinción de edad, raza, cultura o religión. Actualmente posee “25” socios activos, cuatro oficinas en las ciudades de: Macas, Sucua, Gualaquiza y Cuenca.

Con el incremento de nuevas oficinas (Taisha, Zamora y Huaquillas) se ha visto inconformidad por parte de los clientes ya que no pueden tener información sobre sus paquetes (encomiendas) enviados o ubicación de los mismos, además de una sobreventa de boletos en cada una de sus oficinas causando enojo entre pasajeros y mala reputación de la empresa ya que el sistema actual de la cooperativa tiene dificultad para manejo de la información y no es interconectada entre sí.

Para el desarrollo de este proyecto se han tomado en cuenta varias investigaciones científicas, Tesis de grado, Proyectos técnicos que se han llevado a cabo dentro del campo de la Ingeniería de Software entre las que se destacan:

“Sistema informático para la gestión administrativa de las empresas de transporte interprovincial del terminal terrestre de la ciudad de Tulcán”

Según (Hernández y Andrés 2014) el trabajo está dedicado a la elaboración de un sistema informático para la gestión administrativa del terminal terrestre de la ciudad de Tulcán en el campo de venta de boletos, utilizando el estándar de calidad 9126-2 la cual hacen hincapié en la usabilidad.

“Automatización del Control de los Procesos de registro – Turno de Buses, venta de Boletos y encomiendas de la Cooperativa Panamericana Internacional de la Provincia de el Oro.”

Según (Coronel y Julissa 2005) hacen hincapié en el desarrollo de servicios de encomiendas y venta de boletos donde se puede tomar de ejemplo para generar los módulos de la tesis.

“Desarrollo De Un Sistema De Gestión De Encomiendas, Reservación Y Venta De Tickets Para La Cooperativa De Transporte “Fbi”. En La Ciudad De Babahoyo”

Según (Díaz et al. 2012) desarrollan una aplicación exclusivamente para el servicio de encomiendas y al parecido que tiene con el problema planteado donde se tomarán varios aspectos del desarrollo del proyecto técnico.

“Calidad de software e Ingeniería de Usabilidad”

Según (Mascheroni et al. 2012) en este trabajo se describen los principales conceptos sobre usabilidad y los enfoques actuales que proponen la integración de la Ingeniería de Usabilidad a la Ingeniería del Software, así como también la metodología seguida para recabar información acerca de la importancia que las pymes de software confieren a este tema.

Para el desarrollo del trabajo de titulación se ha planteado dividir este documento en tres capítulos que se detallan:

CAPÍTULO I: Se describe todo los conceptos básicos y definiciones de técnicas, herramientas y tecnologías utilizadas en el trabajo.

CAPÍTULO II: En este capítulo se desarrollará en si el sistema que se refiere a procesos de venta de boletos, servicio de encomiendas para el transporte terrestre bajo la metodología de desarrollo (SCRUM) e implementación del algoritmo de encriptación MD5.

CAPÍTULO III: Este es el capítulo más crítico del proyecto ya que es donde se realiza el análisis de los resultados obtenidos y verificar si se cumplen los objetivos planteados en el trabajo de titulación.

FORMULACIÓN DEL PROBLEMA

¿El desarrollo de una aplicación para el servicio de encomiendas y ventas de boletos, de la cooperativa de transporte “CIUDAD DE SUCUA” bajo la norma de calidad ISO/IEC 9126-2 permitirá la mejora de los procesos de envío y entrega de encomiendas y generación de boletos de la cooperativa y prevenir inconvenientes con los clientes?

SISTEMATIZACIÓN DEL PROBLEMA

- ¿Cómo se realiza actualmente el proceso de venta de boletos y servicio de encomiendas en la cooperativa de transporte “Ciudad de Sucua”?
- ¿Cuáles son los requerimientos necesarios para el desarrollo del sistema de encomiendas y venta de boletos?
- ¿Cuáles son las subcaracterísticas para determinar la usabilidad del software de acuerdo a la norma ISO/IEC 9126-2?
- ¿Qué nivel de usabilidad alcanzará el proceso de venta de boletos y servicio de encomiendas de la cooperativa con el sistema propuesto?

JUSTIFICACIÓN TEÓRICA

El sistema de encomiendas y venta de boletos “SoftSucua” que se desarrollara para la cooperativa de transporte interprovincial “Ciudad de Sucua” que se encuentra ubicada en la ciudad del mismo nombre; utilizando la norma ISO/IEC 9126-2 que está orientado a la optimización de la usabilidad externa del software, debe cumplir con algunos parámetros como son: comprensibilidad, facilidad de aprender, operabilidad.

De esta manera satisfacer las necesidades de la empresa, para el desarrollo de esta aplicación se va a utilizar las siguientes tecnologías:

Como gestor de base de datos (DBMS) postgresQL 9.3, como lenguajes de programación JAVA, JSP (Java Server Page), JavaScrip, como entorno integrado de desarrollo Netbeans versión 8.2, como framework de maquetación bootstrap, como metodología de desarrollo de software SCRUM, para determinar la usabilidad de la aplicación la norma ISO/IEC 9126-2 con la

herramienta Crazy Egg, también se incluirá un algoritmo de encriptación MD5 para asegurar contraseñas que se encuentren en el sistema.

Se seleccionaron las herramientas tomando en cuenta que:

Scrum. Para la ejecución del proyecto se tomó la decisión de optar por la metodología de desarrollo ágil Scrum ya que con ella se puede desarrollar a partir de una sola persona además de permitir al desarrollador como la persona que requiere el software ver avances cada quince días (depende la forma en que se ponga de acuerdo el scrum master y el desarrollador) ya que esta metodología trabaja en iteraciones (Sprint). (schwaber, k. y sutherland, j, 2017)

Para iniciar una nueva iteración se debe haber obtenido un producto funcional y haber revisados tareas pendientes. (schwaber, k. y sutherland, j, 2017)

NetBeans IDE. En si es un entorno de desarrollo integrado que se enfoca al desarrollo de aplicaciones con el lenguaje de programación java. Netbenas también permite desarrollar aplicaciones en lenguajes como php, Android entre otros lenguajes de código libre. (netbeans.org, 2016).

PostgreSQL. El Gestor de bases de datos PostgreSQL está orientado a objetos, muy usado en el mundo informatico por ser de código libre y cumplir con estándares como SQL92 y SQL99, además de brindar un conjunto de funcionalidades que permiten aligerar el peso de una base de datos siendo así robusta y confiable. (Fernando y Nora 2012)

Bootstrap. Framework que tuvo sus inicios en Twitter, ayuda a realizar interfaces que sean adaptables con cualquier tipo de dispositivos que tengan un display y un navegador, permite crear interfaces web con CSS y JavaScript, a esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo. (Johanny Solis 2014).

Patrón de diseño MVC. El patrón Modelo-Vista-Controlador es un mapa para diseñar arquitecturas de sistemas que permitan una comunicación eficiente con usuarios. (Aguirre & Moncayo, 2003)

Este patrón de diseño está integrado por tres capas o modelos, los cuales sus características son las siguientes:

Modelo. - También conocido como como acceso datos donde se realizan acciones básicas del sistema. (Aguirre & Moncayo, 2003)

Vista. - Se lo conoce también como interfaz de usuario, en esta capa donde se visualiza todo lo referente a la interacción con el cliente. (Aguirre & Moncayo, 2003)

Controlador. - Son las reglas de negocios también conocido como capa lógica lugar donde se realizan modificaciones sin alterar a las dos capas anteriores. (Aguirre & Moncayo, 2003)

Crazy Egg. Es una herramienta de mapa de calor y analizar el comportamiento de los usuarios para poder evaluar por parte de un equipo de desarrollo y determinar cómo hacer más usable una página identificando los problemas más críticos por parte del cliente. (Casado 2016)

MD5. Es un algoritmo que permite cifrar una cadena de texto usando una clave publica para poder transmitirlo por medio de la web, para comprobar si un password es correcto se compara las dos cadenas de texto que están cifradas. (medina 2015)

JUSTIFICACIÓN APLICATIVA

Para solucionar el problema que presenta la cooperativa de transporte interprovincial “Ciudad de Sucua” el sistema contará con 2 módulos principales que son: **Módulo de Gestión de encomiendas y Módulo gestión de boletos.**

Para el desarrollo de estos módulos se desarrollará una base de datos en la cual estará toda la información requerida correspondiente a la cooperativa, sus servicios, oficinas y clientes; además esto garantiza que los datos estén durante toda la vida del software. Los módulos de desarrollo estarán subdivididos en:

Módulo de Gestión de Boletos

- Ingreso de empleado
- Eliminar de empleados
- Modificar de empleados
- Búsqueda de empleados
- Reporte de empleados general

- Reporte de empleados por oficina
- Ingreso de cliente
- Eliminar de cliente
- Modificar de cliente
- Búsqueda de cliente
- Reporte de cliente general
- Ingreso de socio
- Eliminar de socio
- Modificar de socio
- Búsqueda de socio
- Reporte de socio general
- Ingreso de una oficina
- Eliminar una oficina
- Modificar una oficina
- Buscar una oficina
- Reporte de ingresos por ventas de boletos diario en cada oficina
- Reporte de ingresos por ventas de boletos mensual en cada oficina
- Reporte de ingresos por envío de encomiendas diarias en cada oficina
- Reporte de ingresos por envío de encomiendas mensuales en cada oficina
- Ingreso de unidad de transporté
- Eliminar de unidad de transporté
- Modificar de unidad de transporté
- Búsqueda de unidad de transporté
- Reporte de unidad de transporté general
- Ingreso de ruta
- Eliminación de ruta
- Modificación de ruta
- Búsqueda de ruta
- Reporte de rutas general
- Generación de boletos
- Anulación de un boleto
- Ingreso de reservación por parte del cliente
- Anular una reservación por parte del empleado
- Generación de la hoja de ruta

Módulo de Gestión de encomiendas

- Generación de recibo por encomienda enviada
- Anulación de recibo por encomienda enviada
- Ingreso de paquete para envío
- Modificar un paquete para envío
- Modificar el estado de un paquete
- Eliminar un paquete para envío
- Reporte de paquetes para envío
- Reporte de paquetes en bodega (salida)
- Reporte de paquetes entregados (llegados)
- Reporte de estado de paquetes por de numero de cedula
- Generación de guía de remisión

Arquitectura del Sistema

La arquitectura del sistema consiste en un conjunto de patrones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software. En la Figura.1-1 podemos detallar como estará desplegado el sistema bajo el modelo MVC (Modelo, Vista, Controlador).

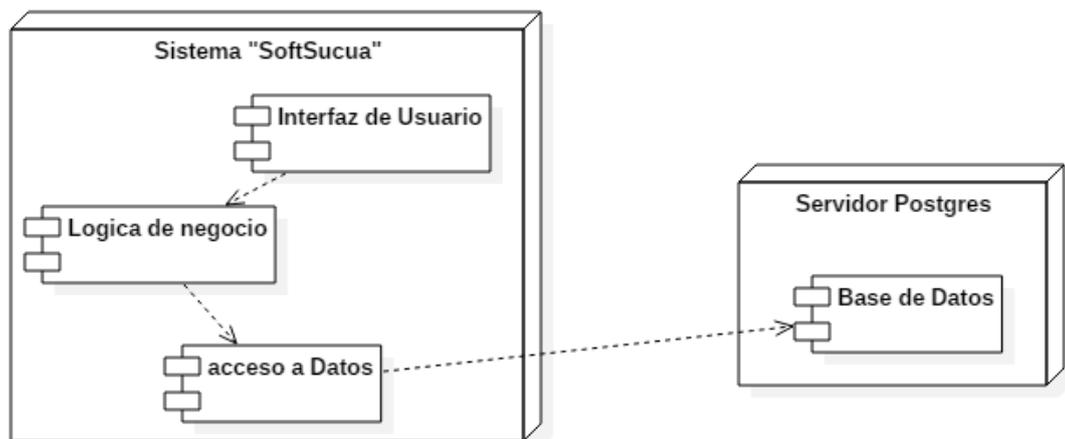


Figura 1-1: Descripción del modelo MVC

Realizado Por: Darío Jiménez, 2017

Reservación de Boletos Online

La reservación de boletos se realizará de la siguiente forma:

- **Reservación realizada por el cliente**
 - Esta reservación se genera a partir del portal web, en el cual el cliente tendrá que registrarse y podrá generar una reservación, vale destacar que el valor de la reservación debe ser cancelado en las oficinas de la empresa con un tiempo máximo de una hora antes de la salida de la unidad de transporté caso contrario el empleado encargado podrá desechar la reservación.

El proyecto a realizar tiene como línea de Investigación de la EIS, Proceso de desarrollo de Software basado en el ámbito de Construcción del Software.

Se ajusta a la línea y programa de Investigación de la ESPOCH, teniendo como línea V. Tecnologías de la Información, Comunicación, Procesos Industriales y Biotecnológicos y como programa para el desarrollo de aplicaciones de software para procesos de gestión y administración pública y privada. Educación.

A su vez que se adapta al PNBV que tiene como objetivo 11. Asegurar la soberanía y eficiencia de los sectores estratégicos para la transformación industrial y tecnológica y como Política 11.3.C Impulsar la calidad, la seguridad y la cobertura en la prestación de servicios públicos, a través del uso de las telecomunicaciones y de las TIC; especialmente para promover el acceso a servicios financieros, asistencia técnica para la producción, educación y salud.

OBJETIVOS

OBJETIVOS GENERALES

Desarrollar un sistema de encomiendas y ventas de boletos para la cooperativa de transporte interprovincial “Ciudad de Sucua” aplicando la norma de calidad ISO/IEC 9126-2.

OBJETIVOS ESPECIFICOS

- Identificar los requerimientos Funcionales y no funcionales del sistema de encomiendas y venta de boletos.
- Planificar el desarrollo del sistema de encomiendas y venta de boletos.
- Diseñar e implementar el sistema de encomiendas y ventas de boletos para la cooperativa de transporte interprovincial “Ciudad de Sucua” aplicando la norma de calidad ISO/IEC 9126-2.
- Determinar la usabilidad del sistema de encomiendas y venta de boletos mediante métricas establecidas en la norma de calidad ISO/IEC 9126-2 y la herramienta Crazy Egg.

CAPÍTULO I

1. MARCO TEÓRICO REFERENCIAL

1.1 Metodología scrum

Según (Agile manifestó, 2017) Scrum se basa en la teoría de control de procesos empírica o empirismo. Lo que quiere decir que se maneja de una forma donde la experiencia ayuda a tomar decisiones correctas basado el conocimiento anterior. (Schwaber y Sutherland, 2013)

Para el desarrollo del proyecto bajo la metodología SCRUM según (Schwaber & Sutherland, 2013) *“existen dos roles indispensables que son: el ScrumMaster quien es el responsable de que el proyecto se esté generando de acuerdo a la metodología, en si es el jefe de proyecto y el Product Owner es aquel que representa al cliente o en si el cliente mismo, es el encargado de verificar si las funciones del sistema son las que se determinan en los requerimientos”*. (Schwaber y Sutherland, 2013)

El avance de un proyecto que está bajo la metodología scrum se determina por medio de iteraciones también llamadas Sprints. (Schwaber y Sutherland, 2013)

1.1.1 Elementos Scrum

El proceso de Scrum posee una mínima cantidad necesaria de elementos formales para poder desarrollar el proyecto, las cuales son:

1.1.1.1 Producto Backlog o Pila del Producto

En si el product backlog es el conjunto de requerimientos que también se los puede llamar PBIs (Product Backlog Items) que se los van obtenidos durante las reuniones mantenidas con el cliente el cual se lo puede organizar mediante una lista de necesidades. (Alaimo, 2013)

Estos listados de necesidades serán realizados por el cliente, el Product Owner y el Scrum Master, el mismo que ayudará a estimar el costo de cada requisito para el desarrollo del sistema, y el valor del producto final software. (Alaimo, 2013)

1.1.1.2 Sprint Backlog

El sprint backlog describen un conjunto de PBIs que se van a realizar en un determinado tiempo (sprint), en el sprint backlog también se generan tareas que cada uno de los desarrolladores debe entregar para generar un producto software funcional que tendrá el software. (Alaimo, 2013)

1.1.1.3 Incremento funcional potencialmente entregable

El resultado de cada Sprint debe ser un incremento funcional y potencialmente entregable.

- **Incremento funcional:** Es una característica funcional nueva (o modificada) de un producto que está siendo construido de manera evolutiva. (Alaimo, 2013)
- **Potencialmente entregable:** Cada una de estas características se encuentra lo suficientemente validada y verificada como para poder ser desplegada en producción o entregada a usuarios finales si así el negocio lo permite o el cliente lo desea. (Alaimo, 2013)

1.1.1.4 Historia de Usuario

La historia de usuario apareció por primera vez dentro de la metodología XP “programación extrema” para poder tener una comunicación más fácil de los especialistas de desarrollo con el cliente y no tener que utilizar documentos muy largos. En si una historia de usuario especifica un determinado PBI para generar una especificación funcional. (Alaimo, 2013)

Los componentes de una historia de usuario se dividen en tres elementos o componentes también se los conoce como “las tres Cs”.

- **Card (Ficha)** – En si una card es un papel pequeño donde se describe toda la información de historia de usuario. (Scrum Manager, 2013)
- **Conversación** – Para realizar la historia de usuario se debe realizar una charla entre el gestor del proyecto (Scrum Master) y el cliente (Product Owner). Esta

conversación debe ser personal y en donde se intercambian información, pensamientos, opiniones y sentimientos. (Scrum Manager, 2013)

- **Confirmación** – Para terminar una historia de usuario se debe tener una confirmación o criterio de aceptación para que el o los desarrolladores (developer team) sepan que es lo que se debe desarrollar. (Scrum Manager, 2013)

1.1.1.5 Ciclo de Vida

El ciclo de vida son las fases que cumple el desarrollo de la metodología

- **Fase de Planificación:** Se define como la fase en donde se abarca el análisis y la visión del proyecto, también se define los límites, el alcance, la arquitectura y diseños generales. (Scrum Manager, 2013)
- **Fase de Diseño:** Esta fase es opcional ya que se la puede incluir en la planificación, pero para mejor entendimiento de cómo se diseñó el proyecto se lo ha implementado al trabajo de titulación. (Scrum Manager, 2013)
- **Fase de Desarrollo de sprints:** En si esta fase se define como el desarrollo del proyecto durante un tiempo determinado para generar una nueva versión del producto tomando en cuenta las variables de tiempo, nuevos requisitos entre otros, también se debe tener en cuenta la cantidad de días disponibles para generar un entregable así el proyecto va evolucionando de acuerdo con cada iteración. (Scrum Manager, 2013)
- **Fase de Cierre:** Esta fase puede ser incluida en la fase de desarrollo y la cual describe el lanzamiento de una versión incluyendo los documentos del mismo. Entre las tareas de cierre se encuentran: integración, pruebas del sistema, documentación de usuario, preparación del material de formación y marketing. (Scrum Manager, 2013)

1.2 Postgresql

1.2.1 Definición

Según (Kasián & Reyes, 2012) postgresql es un motor de bases de datos relacional la cual tiene características como integración de datos, confiabilidad y desempeño, al ser una base de datos libre de licencia (código abierto) puede ser usados y modificado a conveniencia del usuario y se lo puede usar en diferentes sistemas operativos. (Obando, 2013)

1.2.2 Disparadores (triggers)

Una de las funcionalidades disponibles en PostgreSQL son los denominados disparadores (triggers). (Obando, 2013)

En si un disparador o (triggers) es una función que se ejecuta automáticamente al realizar algún **INSERT**, un **UPDATE** o un **DELETE** en una tabla especifica.

Un disparador se puede definir de las siguientes maneras:

- Para que ocurra ANTES de cualquier INSERT,UPDATE ó DELETE. (Obando, 2013)
- Para que ocurra DESPUES de cualquier INSERT,UPDATE ó DELETE. (Obando, 2013)
- Para que se ejecute una sola vez por comando SQL (statement-level trigger). (Obando, 2013)
- Para que se ejecute por cada línea afectada por un comando SQL (row-level trigger). (Obando, 2013)

1.3 Framework

1.3.1 Bootstrap

Es un framework desarrollado por Twitter, el cual trabaja con CSS y JavaScript lo cual ayuda a crear interfaces que se adapten de acuerdo al tamaño del dispositivo que se esté utilizando, es decir que la pagina creada se vea de una forma correcta en una PC, en una Tablet o en cualquier

dispositivo que tenga un navegador web, La técnica de desarrollo se llama “responsive desing”. (Philibert, 2011)

El funcionamiento de Bootstrap se genera a partir del módulo de CSS3 que utiliza jQuery con ellas se puede trabajar en el tamaño de las pantallas (porcentaje, dimensiones) para tener fluidez en los navegadores web, esto ayuda también a consumir menos recursos en cada uno de los dispositivos volviendo más rápidos las transferencias de datos. (Philibert, 2011)

En si Bootstrap posee algunos elementos con estilos predefinidos como: Botones, Menús desplegables, Formularios los cuales se pueden modificar al gusto del desarrollador y decir que cada uno de estos componentes están integrados con jQuery lo cual ofrece que las paginas también puedan ser dinámicas. (Philibert, 2011)

El framework tiene unas pequeñas inconsistencias ya que depende del navegador para ser visto de una mejor manera esto se debe a que no es completo para HTML5, esto no genera que la pagina desarrollada no sea compatible con algunos navegadores web ya que la información básica de Bootstrap es compatible con todos los navegadores. (Johanny Solis 2014)

1.4 Norma iso/iec 9126

1.4.1 Definición

La ISO, bajo la norma IEC-9126, es un estándar internacional utilizado para medir la calidad de un sistema informático que fue desarrollado en el año de 1991, la cual establece características o métricas para medir la calidad del software, su nombre “Information technology–Software product evaluation: Quality characteristics and guidelines for their use” fue publicado en el año de 1992. (Espinoza y Alberto 2014).

El estándar ISO-9126 cuenta con métricas para medir la calidad del software las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; esto permite saber si cumple con la norma ISO-9126, cada métrica se puede subdividir con sus atributos propios y ser verificados para analizar la calidad del producto final. (Espinoza y Alberto, 2014).

En la Figura. 2-1 muestra la pregunta central que atiende cada una de estas características.

<i>Características</i>	<i>Preguntas centrales</i>
<i>Funcionalidad</i>	¿Las funciones y propiedades satisfacen las necesidades explícitas e implícitas; esto es, el qué . . . ?
<i>Confiabilidad</i>	¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?
<i>Usabilidad</i>	¿El software es fácil de usar y de aprender?
<i>Eficiencia</i>	¿Es rápido y minimalista en cuanto al uso de recursos?
<i>Mantenebilidad</i>	¿Es fácil de modificar y verificar?
<i>Portabilidad</i>	¿Es fácil de transferir de un ambiente a otro?

Figura 2-1: Características de la norma ISO 9126

Fuente: (Espinoza y Alberto 2014).

1.4.1.1 Características de la norma iso/iec 9126

- **Funcionalidad.** – Es la capacidad del software para cumplir con los requerimientos establecidos por el cliente. La funcionalidad tiene atributos que ayudan a verificar esto, los cuales son:
 - **Idoneidad.** - Se refiere si el software se desempeña de una manera aceptable en la tarea para el que fue desarrollado.
 - **Exactitud.** - se verifica si los resultados obtenidos del sistema son los esperados.
 - **Interoperabilidad.** - Se analiza si el sistema puede interactuar con otro sistema que se encuentre aparte
 - **Seguridad.** - Verifica si el software no permite acceso a una persona ajena a la empresa.

- **Fiabilidad.** – Es la capacidad del software para brindar un servicio optimo dada las condiciones y un tiempo sin tener inconvenientes. La fiabilidad tiene atributos que ayudan a verificar esto, los cuales son:
 - **Madurez.** - Se refiere a la capacidad del software ante errores ya solucionados en la fase de pruebas.
 - **Recuperabilidad.** – La capacidad del sistema para no perder datos si existe algún error interno o externo.
 - **Tolerancia a fallos.** – La capacidad del sistema ante algún fallo de sí mismo.

- **Eficiencia.** – Es la capacidad del software en atender los requisitos de una forma rápida y confiable. La eficiencia tiene atributos que ayudan a verificar esto, los cuales son:

- **Comportamiento en el tiempo.** - Se refiere al tiempo establecido para realizar alguna tarea en el sistema.
- **Comportamiento de recursos.** – Se refiere si utiliza de manera óptima los recursos empleados para las tareas a realizar.
- **Mantenibilidad.** – La capacidad del sistema para adaptarse a cambios de requerimientos y nuevas especificaciones sin afectar a los ya desarrollados. La mantenibilidad tiene atributos que ayudan a verificar esto, los cuales son:
 - **Estabilidad.** – capacidad del sistema para mantenerse activo cuando haya cambios drásticos Se refiere al tiempo establecido para realizar alguna tarea en el sistema.
 - **Facilidad de análisis.** – El sistema debe tener facilidad en su código para que cualquier persona que esté a cargo pueda entenderlo y analizarlo.
 - **Facilidad de cambio.** – El sistema puede realizar cambios internos sin afectar procesos que se estén ejecutando.
 - **Facilidad de pruebas.** – El sistema no requiera demasiados recursos para realizar pruebas.
- **Portabilidad.** – La capacidad del sistema para poder ser implementado en cualquier sistema operativo (en nuestro caso en cualquier navegador). La portabilidad tiene atributos que ayudan a verificar esto, los cuales son:
 - **Capacidad de instalación.** – capacidad del sistema para ser instalado fácilmente.
 - **Capacidad de reemplazamiento.** – Se refiere a la capacidad que tiene el software para ser reemplazado por otro sistema que realice las mismas tareas.
 - **Co-Existencia.** – El sistema puede convivir con otros sistemas.

1.4.2 iso/iec 9126-2

En si la ISO/IEC 9126-2 permite determinar las métricas externas de software para la medición de las características o las subcaracterísticas de un sistema informático, en si miden el comportamiento del software ante el cliente. (iso.org, 2013)

Las personas o gestores de proyectos que utilicen la norma ISO/IEC 9126-2 pueden tomar o modificar las métricas para aplicar en su proyecto, o se pueden definir las métricas que los usuarios necesitan siempre y cuando estén bajo la especificación de la norma. (iso.org, 2013)

La ISO/IEC 9126-2 explica el cómo y dónde aplicar las métricas de calidad de software, también explica cómo implementar estas métricas durante el ciclo de vida del producto software.

Según (iso.org, 2013) *“la ISO/IEC 9126-2 no asigna rangos de valores de estas métricas a niveles nominales o grados de cumplimiento, porque estos valores están definidos para cada producto de software o una parte del producto de software, por su naturaleza, dependiendo de factores tales como la categoría del software, el nivel de integridad y las necesidades de los usuarios.”* (iso.org, 2013)

Para entender la iso/iec 9126-2 basado a la usabilidad se debe tener en cuenta las siguientes métricas que ayudan a medir la usabilidad del software:

1.4.2.1 Métrica de comprensibilidad

Los usuarios deberían poder seleccionar un producto de software adecuado para su uso previsto. Una métrica de comprensibilidad externa debería poder evaluar si los nuevos usuarios pueden entender (iso.org, 2013):

- Si el software es adecuado
- Cómo se puede usar para tareas particulares.

1.4.2.3 Métrica de aprendizabilidad

Una métrica de capacidad de aprendizaje externa debería poder evaluar cuánto tiempo tardan los usuarios en aprender a usar funciones particulares y la efectividad de los sistemas de ayuda y la documentación. (iso.org, 2013)

La capacidad de aprendizaje está fuertemente relacionada con la comprensibilidad, y las mediciones de comprensibilidad pueden ser indicadores del potencial de aprendizaje del software. (iso.org, 2013)

1.4.2.4 Métricas de operabilidad

Una métrica de operatividad externa debería poder evaluar si los usuarios pueden operar y controlar el software. Las métricas de operatividad se pueden categorizar según los principios de diálogo en ISO 9241-10 (iso.org, 2013):

- Idoneidad del software para la tarea. (iso.org, 2013)
- Autodescriptivo del software. (iso.org, 2013)
- Controlabilidad del software. (iso.org, 2013)
- Conformidad del software con las expectativas del usuario. (iso.org, 2013)
- Tolerancia de error del software. (iso.org, 2013)
- Idoneidad del software para la individualización. (iso.org, 2013)

La elección de las funciones a evaluar dependerá de la frecuencia esperada de uso de las funciones, la criticidad de las funciones y cualquier problema de usabilidad previsto. (iso.org, 2013)

1.4.2.5 Métrica de atractivo

Una métrica de atractivo externo debería poder evaluar la apariencia del software y se verá influenciada por factores como el diseño de la pantalla y el color. Esto es particularmente importante para los productos de consumo. (iso.org, 2013)

1.4.2.6 Métricas de cumplimiento de usabilidad

Una métrica de cumplimiento de usabilidad externa debería poder evaluar el cumplimiento de estándares, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad. (iso.org, 2013)

1.5 Usabilidad

La usabilidad de software se la puede definir como la capacidad que tiene un sistema para ser entendido, de fácil aprendizaje, sin dificultades de uso además de tener una apariencia que satisfaga al usuario final. (Mascheroni, 2012)

Para el estándar ISO 9126-2, el cual se encarga de verificar los requerimientos ergonómicos del sistema esto quiere decir que la usabilidad a partir del estándar ISO 9126-2 es “*el grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso*”. (Mascheroni, 2012)

1.5.1 Comprensibilidad

En si es la capacidad que tiene el sistema para ser entendido por parte del usuario además de verificar el esfuerzo que tuvo que hacer para familiarizarse con el sistema. (Mascheroni, 2012)

1.5.2 Facilidad de Aprender

Se refiere a la capacidad del sistema para que el usuario no sienta complejidad al usar el sistema y se sienta cómodo con el mismo. (Mascheroni, 2012)

1.5.3 Operabilidad

El sistema no debe contar con tareas complejas las cuales hagan lento al sistema y dificultad al usuario, verifica el control que se tiene sobre el sistema. (Mascheroni, 2012)

1.5.4 Manejo de Errores

Este atributo se refiere a la manera en que el sistema puede manejar los errores cometidos por el usuario mientras está realizando una tarea. (Mascheroni, 2012)

1.5.5 Presentación visual apropiada

En si trata que el usuario se siente conforme con el uso de iconos, colores entre otros. (Mascheroni et al. 2012).

1.6 Analítica web

La analítica web consiste en utilizar datos de los usuarios mediante un navegador web y grabarlos en sus primeros pasos en la web se utilizaba en los logs de los servidores, pero en la actualidad las herramientas que se dedican a medir o analizar una página web se la realiza mediante un código que generado por la herramienta en un lenguaje javascript que se inserta en el head o pie de página dependiendo del desarrollador. (analitica web, 2015)

El javascript generado recoge información como por ejemplo los clics dados en un lugar o los scrolls entre otros para ser enviados a la base de datos de la herramienta donde pueden ser consultados por el usuario de la herramienta. (analitica web, 2015)

- Herramientas profesionales más conocidas para realizar la analítica web son:
 - **Crazy Egg.** (analitica web, 2015)
 - **Omniure SiteCatalyst.** (analitica web, 2015)
 - **XiTi.** (analitica web, 2015)
 - **Nedstat.** (analitica web, 2015)
 - **Visual Sciences.** (analitica web, 2015)
 - **WebTrends.** (analitica web, 2015)
 - **ClickTracks.** (analitica web, 2015)
 - **Clickdensity.** (analitica web, 2015)

1.6.1 ¿Qué mide la analítica web?

Las principales medidas o atributos de cualquier herramienta de analítica web son:

- **Páginas vistas.** (analitica web, 2015)
- **Usuarios Únicos.** (analitica web, 2015)
- **Visitas.** (analitica web, 2015)
- **Tiempo de navegación.** (analitica web, 2015)

1.7 Crazy egg

Herramienta la cual permite medir la usabilidad y optimización de una página web mediante la monitorización de los usuarios de dicha página. Permite obtener información para ver como se ha navegado en la página ya que muestra en qué secciones de la web se detienen los usuarios, hasta

qué punto hacen scroll, dónde hacen clic, etc. Con estos datos se pueden tomar decisiones cruciales tanto de diseño como de usabilidad (Beraldo 2016).

1.8 Sistemas de transporte

Un sistema de transporte de carga es aquel que permite movilizar un paquete o mercadería de un lugar a otro teniendo una remuneración por ello. El servicio brindado se rige bajo una cadena de distribución la cual consta de los siguientes integrantes:

- **Paquete.** – El artículo, mercadería, dinero o producto a transportar. (Sanmartin 2011)
- **Remitente.** - La persona que envía el paquete. (Sanmartin 2011)
- **Receptor.** – Persona que recibe el paquete. (Sanmartin 2011)
- **Oficina remitente.** – La oficina que se encarga de enviar los paquetes. (Sanmartin 2011)
- **Oficina receptora.** – la que posee los paquetes enviados y se encarga de despachar. (Sanmartin 2011)
- **Medio de transporte.** – Es la forma de transporta el paquete la cual puede ser terrestre, marítima, aérea. (Sanmartin 2011).

1.9 Algoritmo de encriptación md5

En criptografía, MD5 (Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado. El código MD5 fue diseñado por Ronald Rivest en 1991.

El algoritmo MD5 es una función de cifrado tipo hash que acepta una cadena de texto como entrada, y devuelve un número de 128 bits. Las ventajas de este tipo de algoritmos son la imposibilidad (computacional) de reconstruir la cadena original a partir del resultado, y también la imposibilidad de encontrar dos cadenas de texto que generen el mismo resultado. (medina 2015)

Esto nos permite usar el algoritmo para transmitir contraseñas a través de un medio inseguro, simplemente se cifra la contraseña, y se envía de forma cifrada. En el punto de destino, para comprobar si el password es correcto, se cifra de la misma manera y se comparan las formas cifradas.(medina 2015)

1.10 Java server pages (JSP)

“La tecnología JavaServer Pages (JSP) es una extensión de la tecnología de los servlets. El contenedor de JSPs traduce cada JSP y la convierte en un servlet. A diferencia de los servlets, las JSPs nos ayudan a separar la presentación del contenido”. (DEITEL, 2008)

“Las JavaServer Pages permiten a los programadores de aplicaciones Web crear contenido dinámico mediante la reutilización de componentes predefinidos, y mediante la interacción con componentes que utilizan secuencias de comandos del lado servidor”. (DEITEL, 2008)

“Los programadores de JSPs pueden utilizar componentes especiales de software llamados JavaBeans, y bibliotecas de etiquetas personalizadas que encapsulan una funcionalidad dinámica y compleja”. (DEITEL, 2008)

1.11 Servlets

“Los servlets extienden la funcionalidad de un servidor, al permitir que éste genere contenido dinámico. Por ejemplo, los servlets pueden generar en forma dinámica documentos XHTML personalizados, ayudar a proporcionar un acceso seguro a un sitio Web, interactuar con bases de datos a beneficio de un cliente y mantener la información de sesión única para cada cliente. Un componente del servidor Web, conocido como contenedor de servlets, ejecuta los servlets e interactúa con ellos”. (DEITEL, 2008)

“Los paquetes javax.servlet y javax.servlet.http proporcionan las clases e interfaces para definir servlets. El contenedor de servlets recibe peticiones HTTP de un cliente y dirige cada petición al servlet apropiado”. (DEITEL, 2008)

“El servlet procesa la petición y devuelve una respuesta apropiada al cliente; por lo general en forma de un documento XHTML o XML (Lenguaje de marcado extensible) para mostrarlo en el navegador. XML es un lenguaje que se utiliza para intercambiar datos estructurados en la Web”. (DEITEL, 2008)

En la Figura 3-1 podemos observar cómo es la estructura de una aplicación que se encuentre desarrollada con tecnología jsp y servlets.

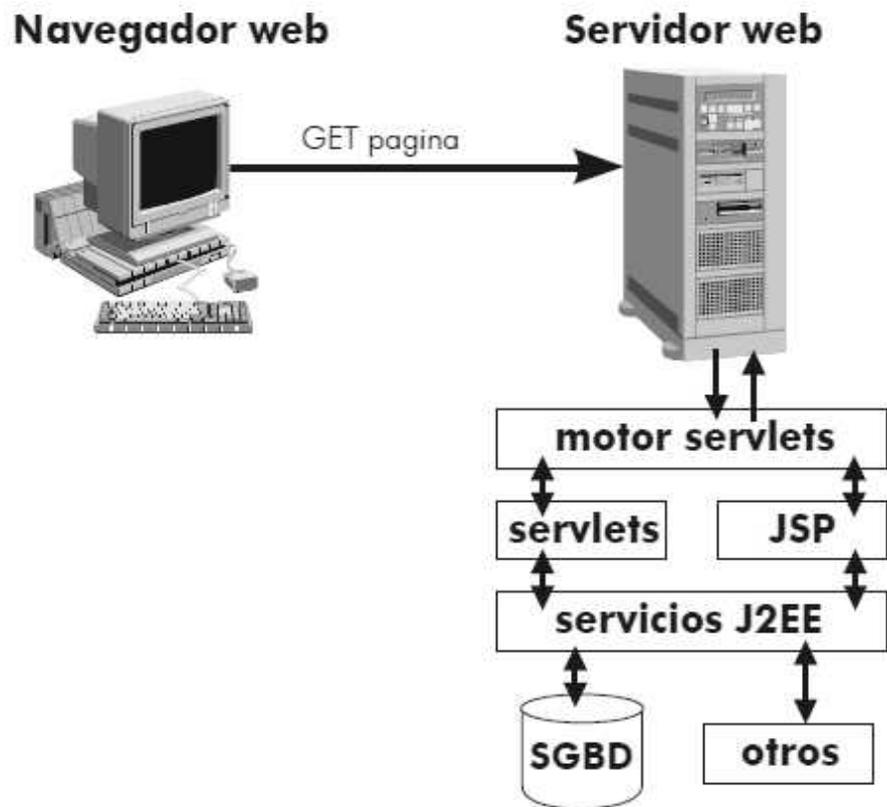


Figura 3-1: Esquema de aplicación usando Servlets y JSP

Fuente: (Sánchez, 2004).

CAPÍTULO II

2 MARCO METODOLOGICO

2.1 Introducción

En este capítulo se tratará sobre la metodología SCRUM la cual esta dividida en la fase de planificación, fase de diseño y fase de desarrollo la cual van a ser detalladas a continuación.

2.2 Fase de planificación

En esta fase se planifica las reuniones con el cliente para determinar cómo se va a desarrollo la aplicación para el servicio de encomiendas y venta de boletos, de la cooperativa de transporte “CIUDAD DE SUCUA”. El proyecto se ejecuta bajo la siguiente planificación que se detalla en el diagrama Gantt, Figura.1-2

★	▲ Fase de análisis y recolección de información	6 días	lun 13/03/17	lun 20/03/17
★	Recolección de la información general	3 días	lun 13/03/17	mié 15/03/17
★	Estudio de la tecnología a utilizar	3 días	jue 16/03/17	lun 20/03/17
★	▲ Fase de Especificación de requerimientos	11 días	mar 21/03/17	mar 04/04/17
★	Estudio preliminar	3 días	mar 21/03/17	jue 23/03/17
★	Toma de requerimientos	4 días	vie 24/03/17	mié 29/03/17
★	Análisis de requerimientos	4 días	jue 30/03/17	mar 04/04/17
★	▲ Fase de desarrollo del sistema de gestión de encomiendas y generación de boletos	60 días	mié 05/04/17	mar 27/06/17
★	Planificación de actividades	6 días	mié 05/04/17	mié 12/04/17
★	Desarrollo de la metodología (Sprint)	50 días	jue 13/04/17	mié 21/06/17
★	Implementación del algoritmo de encriptación MD5	1 día	lun 22/05/17	lun 22/05/17
★	Cierre	3 días	vie 23/06/17	mar 27/06/17

Figura 1-2: Diagrama de Actividades

Realizado Por: Darío Jiménez. 2017.

2.2.1 Análisis y recolección de información

Esta etapa nace con las reuniones realizadas entre el Scrum Master, product owner y/o clientes, en la tabla 1-2 se puede observar el detalle de las reuniones realizadas las fechas y las actividades.

Mediante cuatro reuniones de trabajo se obtuvo un total de cincuenta requerimientos funcionales y cuatro requerimientos no funcionales.

Tabla 1-2: Desarrollo reuniones

Encargados	Actividad	Fecha	Resultados
Ing. Adriana Cárdenas gerente de la cooperativa de transporte “Ciudad de Sucua”	Recolección de información correspondiente a la cooperativa (necesidades y posibles soluciones)	Desde:13/03/2017 Hasta: 16/03/2017	Información de cómo se maneja la información correspondiente a la venta de boletos y gestión de paquetes.
Sra. Verónica Urquiza secretaria de la oficina ubicada en ciudad de macas de la cooperativa de transporte “Ciudad de Sucua”	Recolección de información correspondiente a la cooperativa (necesidades y posibles soluciones)	Desde:13/03/2017 Hasta: 16/03/2017	Información de cómo se maneja la información correspondiente a la venta de boletos y gestión de paquetes en las oficinas.
Ing. Adriana Cárdenas gerente de la cooperativa de transporte “Ciudad de Sucua”	Definición preliminar de requerimientos funcionales y no funcionales del sistema	Desde:21/03/2017 Hasta: 26/03/2017	Bosquejo de la información para comenzar a definir requerimientos funcionales y no funcionales del sistema.
Ing. Adriana Cárdenas gerente de la cooperativa de transporte “Ciudad de Sucua”	Presentación de requerimientos funcionales y división en módulos del sistema	Desde:26/03/2017 Hasta: 30/03/2017	Aceptación de las partes para el desarrollo del sistema

Realizado Por: Darío Jiménez. 2017

2.2.2 Módulo Gestión de Boletos

El módulo de gestión de boletos se enfoca a la generación de boletos (tickets), en este módulo también se genera el ingreso de información de empleados, clientes, socios, unidades de transporte, rutas, reportes correspondientes a todo lo mencionado anteriormente.

2.2.3 Módulo Gestión de encomiendas

El módulo de gestión de encomiendas se enfoca a la administración de paquetes (ingresados, en viaje, no enviados) además de poder genera una hoja de ruta para las encomiendas enviadas.

2.2.4 Roles del proyecto

Para el desarrollo del sistema se debe nombrar y definir el personal encargado en cada área, en la tabla 2-2 se detallan a las personas y su rol en el proyecto.

Tabla 2-2: Roles y personal encargados del proyecto

Rol	Nombre Encargado
Product Owner	Ing. Adriana Cárdenas
Scrum master	Ing. Jonny Guaiña
Developer team	Darío Jiménez

Realizado Por: Darío Jiménez. 2017

2.2.5 Product Backlog

El product backlog (pila de producto) en si es la lista de todas las historias de usuario determinadas por su importancia para el negocio la cual se detallan las 10 primeras historias de usuario en la tabla 3-2, para el revisar todo el product backlog revisar el **anexo A**.

Tabla 3-2: Product backlog

Código	Historia de Usuario
HU_01	Generación de boletos
HU_02	Anulación de un boleto

HU_03	Ingreso de reservación por parte del cliente
HU_04	Anular una reservación por parte del empleado
HU_05	Generación de la hoja de ruta
HU_06	Generación de recibo por encomienda enviada
HU_07	Anulación de recibo por encomienda enviada
HU_08	Ingreso de paquete para envío
HU_09	Modificar un paquete para envío
HU_10	Modificar el estado de un paquete

Realizado Por: Darío Jiménez

2.2.6 Presupuesto para el desarrollo

2.2.6.1 Costo de Software

El costo a lo que se refiere a software se detalla en la tabla 4-2.

Tabla 4-2: Detalle costos en software

NOMBRE	DESCRIPCIÓN	CANTIDAD	VALOR/U	TOTAL
Centos	Versión 8.2	1	-	-
Postgresql	Versión 9.3	1	-	-
GlassFish	Versión v4.0	1	-	-
TOTAL				-

Realizado Por: Darío Jiménez

2.2.6.2 Costo de Hardware

El costo a lo que se refiere a hardware se detalla en la tabla 5-2.

Tabla 5-2: Detalle costo de hardware

NOMBRE	DESCRIPCIÓN	CANTIDAD	VALOR/U	TOTAL
Laptop Dell Xps 15 i7 6700hq.	Desarrollo aplicación	1	1500	2350
Computadora de escritorio core i5, 500 GB	Para uso de las Oficinas	7	500	3500

almacenamiento y 4 GB de memoria ram				
TOTAL				5850

Realizado Por: Darío Jiménez

2.2.6.3 Costo de Varios

El costo a lo que se refiere a varios se detalla en la tabla 6-2.

Tabla 6-2: Detalle de costos varios

NOMBRE	DESCRIPCION	CANTIDAD	VALOR/U	TOTAL
Cuadernos	Apuntes	2	1,10	2,20
Resma de Papel	Impresiones	3	3,50	10,50
Esferos	Usos varios	6	0,40	2,40
Lápices	Usos varios	6	0,25	1,50
Bibliografía	Libros desarrollo sistema	4	50	200
TOTAL				216,60

Realizado Por: Darío Jiménez

Total, de los costos= \$ 6066,60

El costo total del desarrollo nos indica el total de gastos que se utilizarán durante el desarrollo del sistema, costos de software, hardware y gastos de suministros.

A estos costos se debe incluir también los valores correspondientes a la estimación del costo del proyecto con el cual tenemos un total del costo del proyecto es de aproximadamente de **veinte y cuatro mil ochocientos noventa y cuatro dólares con sesenta centavos**. El costo del proyecto va a estar financiado por gestor del proyecto.

2.3 Fase de diseño

Para una mejor comprensión de cómo está diseñado el sistema se implementó la fase de diseño la cual tendrá diagramas de diseño, estándar de codificación, diseño de la interfaz de usuario y la base de datos que se detallan a continuación.

2.3.1 Diagramas de Casos de Uso

En este apartado se podrá apreciar los diferentes casos de uso que va a constituir el sistema, los más representativos son las de ventas de boletos y la gestión de encomiendas, funciones que serán desarrolladas por los empleados de la cooperativa.

2.3.1.1 Caso de uso venta de boletos

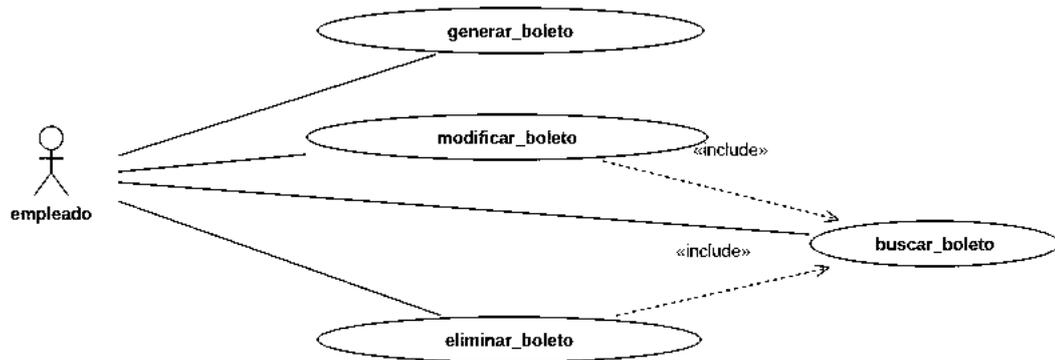


Figura 2-2: Diagrama, caso uso Venta de boletos Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 20117

2.3.1.2 Caso de uso gestión encomiendas

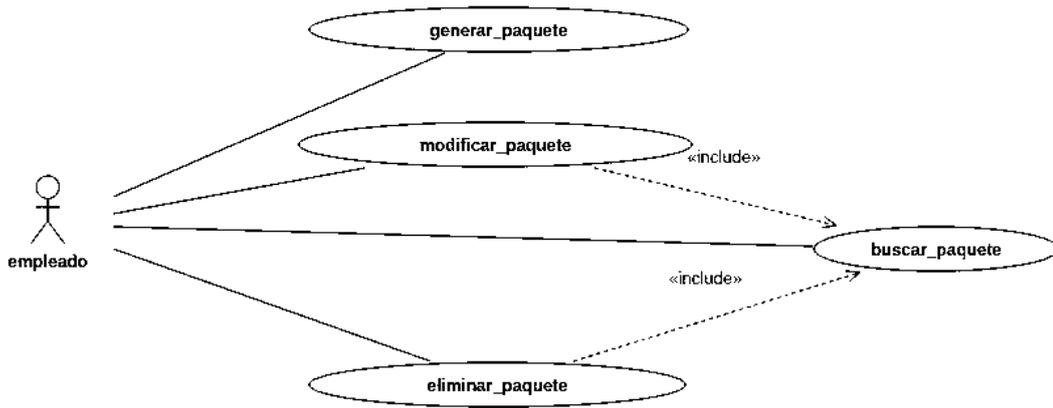


Figura 3-2: Diagrama, caso uso gestión de encomiendas Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

2.3.2 Diagrama de Clases

En este apartado se podrá apreciar las clases que interactuarán en el sistema, cada una de estas se encuentran con sus atributos y métodos como se aprecia en el Figura 6-2, es una de las partes fundamentales para empezar a desarrollar el sistema.

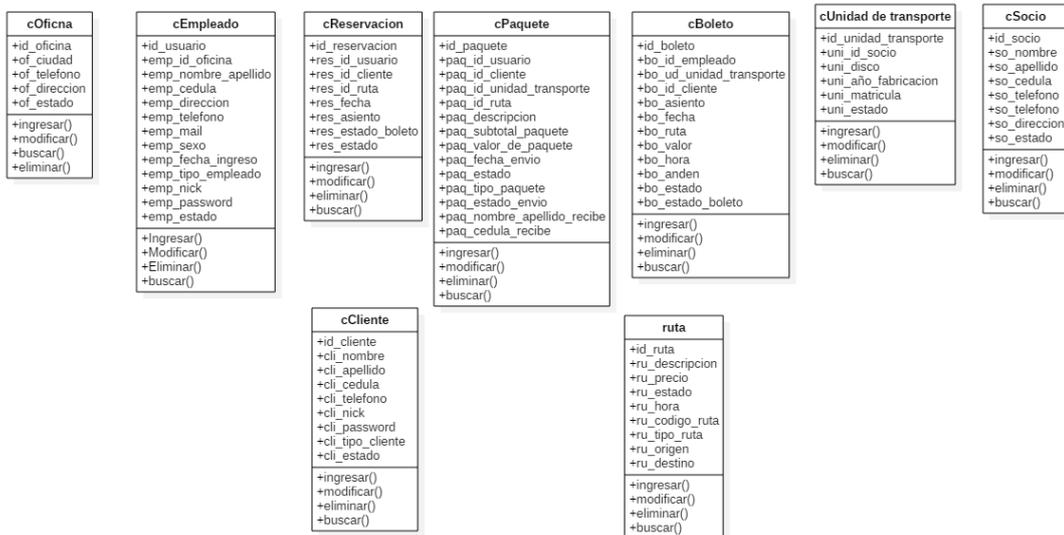


Figura 4-2: Diagrama de clases Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

2.3.3 Diagramas de Despliegue

En este diagrama se detalla cómo está desplegado el sistema en cada una de sus capas además tener la información del servidor de base de datos como se puede observar en la Figura. 5-2.

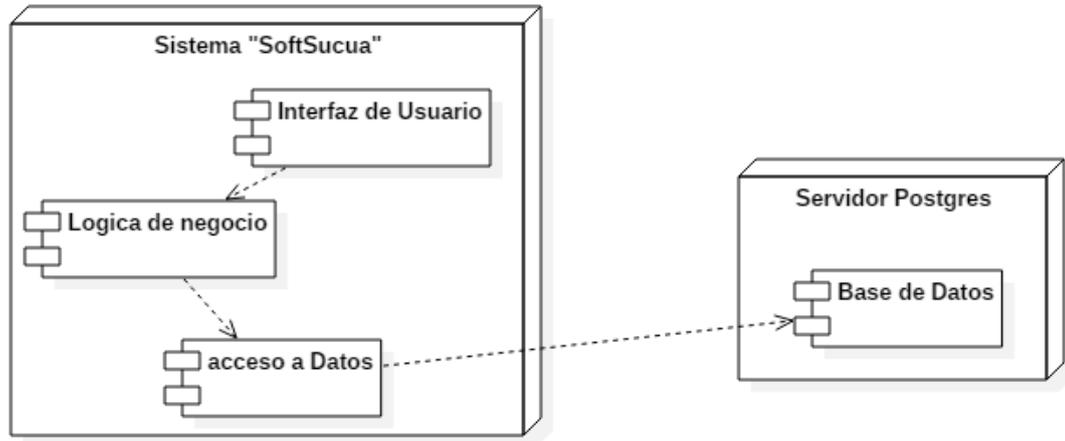


Figura 5-2: Diagrama de despliegue Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

2.3.4 Arquitectura del Sistema

Se ha establecido la correspondiente arquitectura del sistema que representa el esquema que se seguirá en el desarrollo e implementación del sistema y cómo se maneja los datos o peticiones que se realiza por medio de la interfaz de usuario hasta llegar a la base de datos.

En la elaboración del sistema SoftSucua realizaremos la arquitectura MVC, para tener flexibilidad cuando haya cambios que pueda incurrir en cambios para el sistema. Para el proyecto se utilizará 1 computador que trabajan como servidor, con las capas de base de datos, acceso a datos, lógica del negocio y GUI (Graphical User Interface). Permitiendo de esta manera trabajar con la arquitectura MVC (modelo,vista, controlador) como se detalla en la Figura. 6-2.

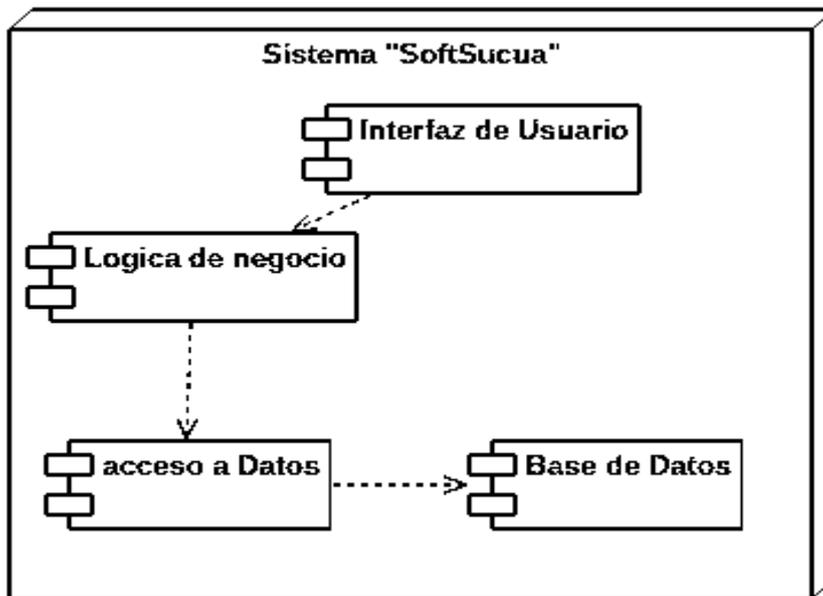


Figura 6-2: Arquitectura del Sistema Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

2.3.5 Estándar de Codificación

En el desarrollo del presente sistema se usará CamelCase como estándar de codificación ya que es la nomenclatura por excelencia en el mundo Java siendo el más sencillo, ya que utiliza reglas simples y fáciles de comprender.

En CamelCase su nombre se debe a que las mayúsculas a lo largo de una palabra en CamelCase se asemejan a las jorobas de un camello; que se podría traducir como Mayúsculas/Minúsculas Camello para un mayor detalle del estándar de codificación revisar **anexo B**.

2.3.6 Interfaz de Usuario

Para el desarrollo de la interfaz se tomaron en cuenta la teoría de colores y los colores representativos de la cooperativa, tomando en cuenta estos aspectos se determinó los siguientes colores como fundamentales para el desarrollo de la interfaz los cuales son verde, negro, blanco.

En cuanto al tipo de letra se utilizó Arial con un tamaño de doce para los textos como son ingresos listados, etc. con para los títulos se utilizó un tipo de letra Arial de 16 para la mejor visibilidad

del cliente, al utilizarse un fondo de color negro las letras fueron color verde en los menús y letras de color negro en fondos de color blanco con lo cual el usuario se sintió satisfecho.

Esto permitirá que el usuario final del sistema se familiarice muy fácilmente en un corto tiempo sin perjudicar su trabajo y por el contrario hacerlo más ligero. Las interfaces están divididas en seis interfaces principales que son:

2.3.6.1 Interfaz Principal

Es la página donde se encuentra información principal de la empresa como misión, visión, número telefónicos entre otros como describe la Figura 7-2.



Figura 7-2: Página principal Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

2.3.6.2 Interfaz Login

Página donde el cliente, administrados o empleado de la empresa podrá iniciar sesión, donde será necesario un usuario y contraseña y en caso de ser cliente nuevo registrarse como nuestra la figura 8-2.



Figura 8-2: Página login, Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

2.3.6.3 Interfaz Empleado-Administrador

Esta interfaz principal donde el administrador podrá realizar cualquier operación que necesite como muestra la figura 9-2.



Figura 9-2: Página Administrador, Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

2.3.6.4 Interfaz Empleado-Secretario/a

Interfaz donde se encuentran las funciones del empleado en este caso una secretaria como muestra la figura 10-2.

DÍA	FRECUENCIA	HORA
1	MACAS-LIMON-CUENCA	05H30
1	CUENCA-LIMON-MACAS	18H30
2	MACAS-GUALAGUZA-ZAMORA	09H30
2	ZAMORA-GUALAGUZA-MACAS	22H30
3	MACAS-LIMON-CUENCA-HUAGULLAS	20H30
4	HUAGULLAS-CUENCA-LIMON-MACAS	21H30
5	MACAS-GUALAGUZA-ZAMORA	12H30
6	ZAMORA-GUALAGUZA-MACAS	02H30
6	MACAS-LOGROÑO	13H00
6	LOGROÑO-MACAS	15H00 (SUCUA 15H45)

Figura 10-2: Página Secretario/a, Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

2.3.6.5 Interfaz Cliente

Página donde el cliente podrá revisar el estado de su paquete y realizar reservaciones de un turno antes previsto como se visualiza en la figura 11-2

DÍA	FRECUENCIA	HORA
1	MACAS-LIMON-CUENCA	05H30
1	CUENCA-LIMON-MACAS	18H30
2	MACAS-GUALAGUZA-ZAMORA	09H30
2	ZAMORA-GUALAGUZA-MACAS	22H30
3	MACAS-LIMON-CUENCA-HUAGULLAS	20H30
4	HUAGULLAS-CUENCA-LIMON-MACAS	21H30
5	MACAS-GUALAGUZA-ZAMORA	12H30
6	ZAMORA-GUALAGUZA-MACAS	02H30
6	MACAS-LOGROÑO	13H00
6	LOGROÑO-MACAS	15H00 (SUCUA 15H45)

Figura 11-2: Página Cliente, Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

2.3.7 Diseño Base de datos

La base de datos es uno de los principales componentes de nuestro sistema por lo cual para garantizar su correcta implementación y funcionamiento se realizó el diseño preliminar de la

misma que consta del Diagrama Entidad-Relación, los modelos lógicos, físico, el diccionario de datos y el correspondiente script.

Como se puede observar en la Figura. 12-2 que corresponde al diagrama entidad-relación, en la cual existen 9 entidades la primera es la entidad oficina, el cual va contiene información correspondiente a cada una de las oficinas de la cooperativa, también se implementó la entidad empleado el cual representa a los y las trabajadores de la cooperativa.

La entidad paquete es donde estará la información de todo el proceso de encomiendas, la entidad reservación se encargará de albergar toda la información de las reservaciones realizadas por los clientes, la entidad cliente contiene datos de clientes, la entidad boleto es una de las entidades más importantes ya que de aquí esta información muy valiosa para la cooperativa.

Por otro lado, también tenemos las entidades ruta, socio y unidad de transporte que es información correspondiente a la administración de la cooperativa. Para revisara de forma más detallada la información correspondiente a los demás modelos revise en el **anexo C**.

Realizado Por: Darío Jiménez. 2017

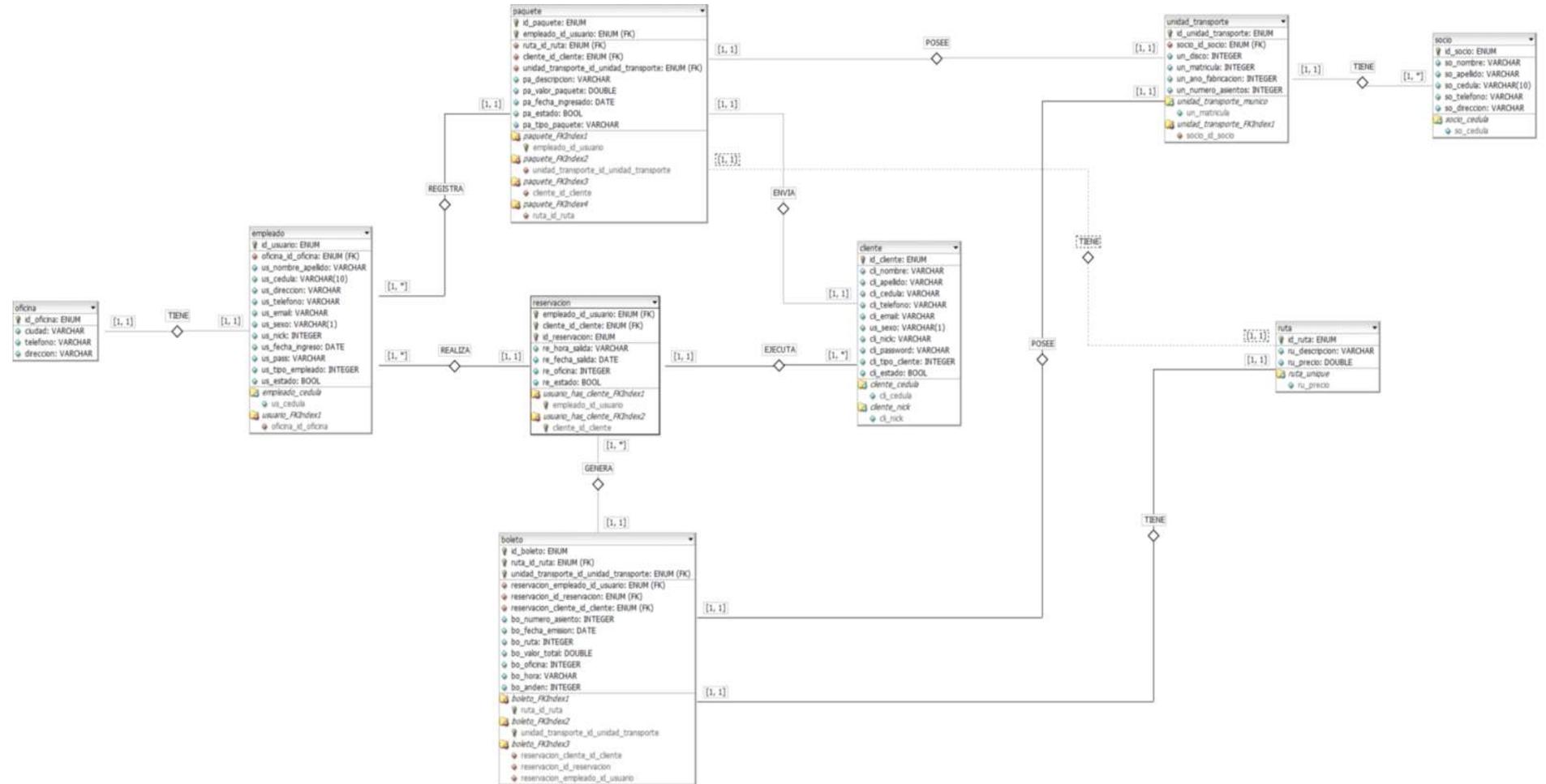


Figura 12-2: Modelo DER, Sistema “SoftSucua”

2.4 Fase de desarrollo

Para la implementación de esta etapa se debe proceder a ejecutar cada uno de los sprint planificados anteriormente de una forma organizada y cumpliendo con las actividades, las funcionalidades del sistema se realizan paulatinamente cumpliendo con las especificaciones de la metodología SCRUM, tomando en cuenta la documentación en cuanto a tareas de ingeniería y pruebas de aceptación.

2.4.1 Sprint Backlog (Lista de sprint)

El sprint backlog es el conjunto de historias de usuario que se van a realizar en un tiempo determinado, el proyecto fue diseñado para ser trabajado en cinco sprints y un sprint adicional donde se realizaron las metáforas del sistema, el cual se detalla a continuación en la tabla 7-2.

Tabla 7-2: Sprint Backlog

Sprint	Detalle	F. Inicio	F. Final
Sprint 0: metáforas del sistema	El sprint 0 nace con la planificación del proyecto incluyendo en este las metáforas del sistema	lun 13/03/17	vie 07/04/17
Sprint 1: Gestión de clientes y empleados	El sprint 1 contiene todo lo relacionado a lo que es los empleados y clientes también se implementan reportes	lun 10/04/17	vie 21/04/17
Sprint 2: Gestión de oficinas y socios	El sprint 2 contiene todo lo relacionado con oficinas y socios también se implementan reportes de socios	lun 24/04/17	vie 05/05/17
Sprint 3: Gestión de reportes de oficinas, gestión de unidad de transporte y gestión de rutas.	El sprint 3 contiene todo lo relacionado con la gestión de una unidad de transporte, de rutas y reportes de oficinas detalladas	lun 08/05/17	vie 19/05/17

Sprint 4: Gestión de boletos	En este sprint se ha desarrollado todo lo correspondiente a boletos	lun 22/05/17	vie 02/06/17
Sprint 5: Gestión de encomiendas	En este sprint se ha desarrollado todo lo correspondiente a paquetes y encomiendas	lun 05/06/17	vie 16/06/17

Realizado Por: Darío Jiménez. 2017

En la tabla 8-2 se puede observar el detalle del sprint 1 que contiene información de clientes y empleados

Tabla 8-2: Detalle del Sprint 1

SPRINT 1	10 días	lun 10/04/17	vie 21/04/17	
Autenticación de usuarios		lun 10/04/17	lun 10/04/17	3 pts
Ingresar un empleado		lun 10/04/17	lun 10/04/17	3 pts
Modificar un empleado		mar 11/04/17	mar 11/04/17	6 pts
Eliminar un empleado		mie 12/04/17	mie 12/04/17	6 pts
Buscar un empleado		jue 13/04/17	jue 13/04/17	6 pts
Ingresar un cliente		vie 14/04/17	vie 14/04/17	6 pts
Modificar un cliente		lun 17/04/17	lun 17/04/17	6 pts
Eliminar un cliente		mar 18/04/17	mar 18/04/17	6 pts
Buscar un cliente		mie 19/04/17	mie 19/04/17	6 pts
Reporte de empleados general		jue 20/04/17	jue 20/04/17	3 pts
Reporte de cliente general		jue 20/04/17	jue 20/04/17	3 pts
Reporte de empleados por oficina		vie 21/04/17	vie 21/04/17	6 pts

Realizado Por: Darío Jiménez. 2017

Para más detalles revisar el **anexo D** que contiene cada uno de los sprint detallados.

2.4.2 Historias de Usuarios

En este ítem se analizará el médelo de las historias como están conformadas sus atributos para mejor entendimiento se detalla una historia de usuario en la tabla 9-2.

Tabla 9-2: Historias de Usuarios

HISTORIA DE USUARIO	
Numero: HU_01	Nombre: Login de usuarios
Modificación historia de usuario:	
Usuario: Usuario	Sprint Asignado: 1
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en desarrollo: Alta	Puntos Reales: 5
Descripción: Como Administrador quiero que tanto los clientes como mis secretarias o secretarios no tengan acceso a toda mi información, sino a cierta parte de ella.	
Observaciones: Cada usuario se diferenciará por un rol.	

(anverso de la historia de usuario)

Pruebas de Aceptación
<ul style="list-style-type: none"> • Si se autentica el Administrador tendrá acceso a toda la información. • Si se Autentifica el secretario/a podrá realizar ciertas funcionalidades. • Si se Autentifica el Cliente podrá realizar ciertas funcionalidades.

Realizado Por: Darío Jiménez. 2017

El resto de conjunto de historias de usuario se puede obtener en el manual de usuario pag. 75.

2.4.3 Pruebas de aceptación

En este ítem se analizará el médelo de las pruebas de aceptación que se tiene para cada historia de usuario para mejor entendimiento se detalla una prueba de aceptación en la tabla 10-2

Tabla 10-2: Detalle de Prueba de aceptación

PRUEBA DE ACEPTACIÓN	
Código: PA_01	Metáfora del sistema: HU_01 Logue de usuarios.
Nombre: Si se autentica el Administrador tendrá acceso a toda la información.	
Responsable: Darío Jiménez	Fecha: 10/04/17
Descripción: Si la información ingresada tanto en el campo usuario y contraseña es correcta podrá tener acceso a las funcionalidades del Administrador.	

<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Debe existir datos en la tabla Administrador.
<p>Pasos de ejecución:</p> <ul style="list-style-type: none"> • Ejecutar la página de Login. • Ingresar datos en el campo usuario. • Ingresar datos en el campo contraseña. • Verificar los datos de la base de datos. • Enviar la información del formulario. • Verificar que accede sin ningún problema a la página principal después de hacer presionar en el botón Ingresar.
<p>Resultado esperado: Se ingresa sin ningún problema y se cargan las funcionalidades según el tipo de usuario que sea.</p>
<p>Evaluación de la prueba: Exitosa.</p>

Realizado Por: Darío Jiménez. 2017

El conjunto de pruebas de aceptación se puede obtener en el manual de usuario desde la pag. 75.

2.4.4 Tareas de ingeniería

Las tareas de ingeniería son aquellas que detallan lo que se hará durante un tiempo determinado para cumplir una historia de usuario, para mejor entendimiento se detalla una prueba de aceptación en la tabla 11-2.

Tabla 11-2: Detalle tarea de ingeniería

TAREA DE INGENIERÍA	
Historia de Usuario: HU_01 Logue de usuarios.	
Número de Tarea: TI_1	Nombre de Tarea: lógica de negocio.
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha de inicio: 10/04/17	Fecha Fin: 10/04/17
Programador responsable: Darío Jiménez	
Descripción:	
Crear un cliente que consuma el servicio web del método login e implementar un servicio web que reciba un objeto tipo Usuario y mande sus datos a su cliente web para los realizar la búsqueda en la base.	
Pruebas de aceptación:	
<ul style="list-style-type: none"> • Verificar que el servicio web de la capa de lógica de negocio realiza la acción de Búsqueda de un Usuario, Cliente 	

Realizado Por: Darío Jiménez. 2017

El conjunto de tareas de ingeniería se puede obtener en el manual de usuario desde la pag. 75.

2.4.5 Gráfica de Producto (Burn Up)

En este apartado se trata de tener una visión más clara de la evolución del sistema tomando en cuenta la planificación estimada y el desarrollo real que se obtuvo en el desarrollo del proyecto como se detalla en la Figura 15-2.

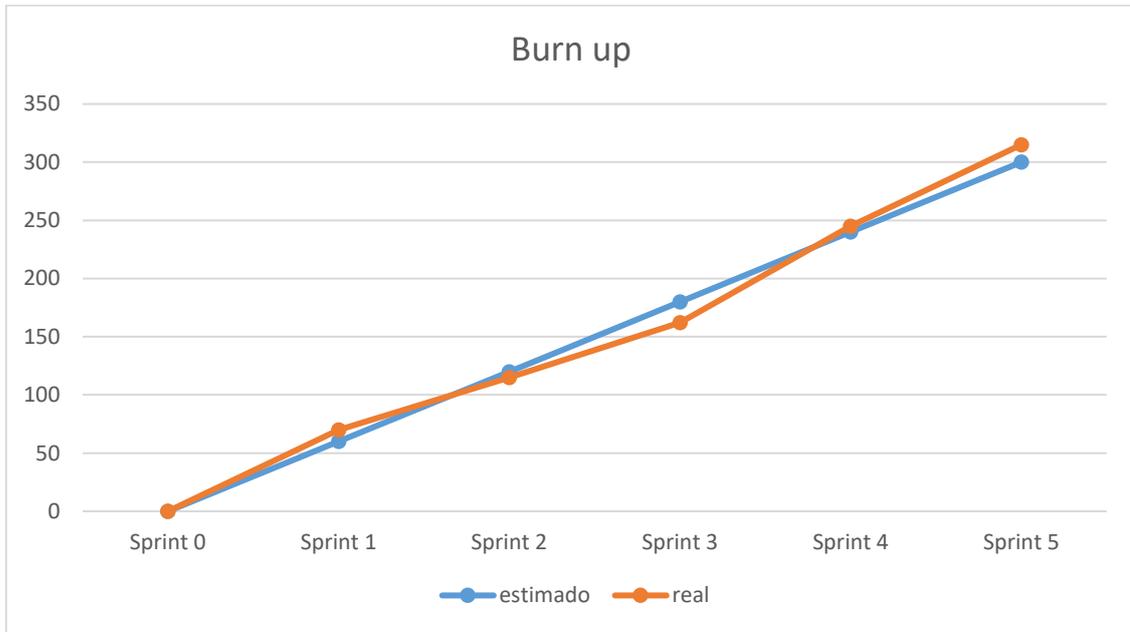


Figura 13-2: Burn up, Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

2.4.6 Implementación del algoritmo de encriptación md5

Para la implementación del cifrado md5 se tomó en cuenta la teoría descrita en el marco teórico y con el algoritmo previamente desarrollado se implementó en el proyecto, tomando en cuenta que el algoritmo solo va a ser implementado para las contraseñas de los empleados como de los clientes de la cooperativa. La Figura. 14-2 y la Figura.15-2 se detalla cómo esta implementado el algoritmo en el sistema.

```

public static String Encriptar(String texto) {
    String secretKey = "cooperativasucua2017"; //llave para encriptar datos cooperativasucua2017
    String base64EncryptedString = "";

    try {
        MessageDigest md = MessageDigest.getInstance("MD5");
        byte[] digestOfPassword = md.digest(secretKey.getBytes("utf-8"));
        byte[] keyBytes = Arrays.copyOf(digestOfPassword, 24);

        SecretKey key = new SecretKeySpec(keyBytes, "DESede");
        Cipher cipher = Cipher.getInstance("DESede");
        cipher.init(Cipher.ENCRYPT_MODE, key);

        byte[] plainTextBytes = texto.getBytes("utf-8");
        byte[] buf = cipher.doFinal(plainTextBytes);
        byte[] base64Bytes = Base64.encodeBase64(buf);
        base64EncryptedString = new String(base64Bytes);
    } catch (Exception ex) {
    }
    return base64EncryptedString;
}

```

Figura 14-2: Algoritmo Encriptación, Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

```

public static String Desencriptar(String textoEncriptado) throws Exception {
    String secretKey = "cooperativasucua2017"; //llave para encriptar datos
    String base64EncryptedString = "";

    try {
        byte[] message = Base64.decodeBase64(textoEncriptado.getBytes("utf-8"));
        MessageDigest md = MessageDigest.getInstance("MD5");
        byte[] digestOfPassword = md.digest(secretKey.getBytes("utf-8"));
        byte[] keyBytes = Arrays.copyOf(digestOfPassword, 24);
        SecretKey key = new SecretKeySpec(keyBytes, "DESede");

        Cipher decipher = Cipher.getInstance("DESede");
        decipher.init(Cipher.DECRYPT_MODE, key);

        byte[] plainText = decipher.doFinal(message);

        base64EncryptedString = new String(plainText, "UTF-8");
    } catch (Exception ex) {
    }
    return base64EncryptedString;
}

```

Realizado Por: Darío Jiménez. 2017

Figura 15-2: Algoritmo descriptacion datos, Sistema "SoftSucua"

En la Figura. 16-2 se puede apreciar cómo están encriptado las contraseñas(passwords) en la tabla empleados.

	id_cliente [PK] serial	di_nombre character va	di_apellido character va	di_cedula character va	di_telefono character va	di_nick character va	di_password character va	di_tipo_cien integer	di_estado boolean
1	1	consumidor	final	9999999999	00000000000	0105103907	yA2CW0FDDNg	1	TRUE
2	5	Daniela	Flores	1400223432	0990556715	1400223432	zn2KAPBOGPR	1	FALSE
3	6	Estefania	Jimenez	0105130907	0990556715	0105130907	PAeif9igjYY	1	FALSE
4	7	Adriana	Tapia	0105130905	0990556715	0105130905	PAeif9igjYZ	1	TRUE
5	17	Daniela	Arevalo	0914899562	0990556715	0914899562	EBkRgBOPQB9	1	TRUE
6	19	Celso	Jimenez	0102223575	0990556715	martincito	yA2CW0FDDNg	1	TRUE
*									

Figura 16-2: Tabla postgresql empleados, Sistema "SoftSucua"

Realizado Por: Darío Jiménez. 2017

CAPÍTULO III

3 MARCO DE RESULTADOS

3.1 Introducción

Para el desarrollo de este apartado se tomó en cuenta los resultados obtenidos al implementar el software en la cooperativa de transporte “Ciudad de Sucua”, estos mismos datos serán detallados a lo largo del este capítulo.

3.2 Determinación de usabilidad bajo la norma iso 9126-2

Para esta etapa del proyecto se tomó en cuenta el uso del estándar de calidad ISO 9126-2 en cual se trata específicamente en la usabilidad del software teniendo en si varios parámetros para verificar si un software es usable, estos parámetros también son llamadas métricas de calidad las cuales están detalladas en el marco teórico del proyecto. Para la determinación de la usabilidad se determinaron dos formas las cuales son:

3.2.1 Métricas para la usabilidad de software

En este parte se realizó una evaluación empírica la cual consta de realizar preguntas a los usuarios finales sobre el funcionamiento del software tomando en cuenta las métricas establecidas en el estándar (esto no quiere decir que el software es totalmente usable ya que para unos va a estar bien y para otros usuarios no), a continuación, se detallan las métricas que se tomaron en cuenta para esta evaluación.

- Facilidad de aprendizaje
- Tiempo de respuesta
- Entendibilidad
- Atractivo

Para determinar a la usabilidad del software desde la vista de los usuarios finales se realizó el test que se describe en el **anexo E**, en donde las calificaciones son calificativas y no cuantitativas ya que la usabilidad no se la puede medir en números.

3.2.1.1 Muestra

Se tiene una población de 25 empleados los cuales trabajan en la empresa y los cuales se dividen en cinco empleados por oficina ya que son las estas el número de oficinas habilitas de la empresa que cuentan con servicio de encomiendas y venta de boletos (dos que están por ser implementada).

Para obtener la muestra de esta población tomamos en cuenta los siguientes datos:

$$N = 25$$

$$Z = 95\% = 1.96$$

$$P = 0.5$$

$$Q = 0.5$$

$$E = 2\% = 0.2$$

Ecuación para encontrar la muestra

$$n = \frac{Z^2 \times N \times P \times Q}{[E^2 \times (N - 1)] + [Z^2 \times P \times Q]}$$

Remplazando los datos:

$$n = \frac{1.96^2 \times 25 \times 0.5 \times 0.5}{[0.2^2 \times (25 - 1)] + [1.96^2 \times 0.5 \times 0.5]}$$

$$n = \frac{24.01}{0.97}$$

$$n = 24.75$$

El resultado que se obtuvo es muy parecido a la población, esto se debe a que la población estudiada es pequeña así que se tomara toda la población para realizar el estudio.

3.2.1.2 Datos Obtenidos

Los datos obtenidos al realizar el test a los empleados y clientes que tuvieron una capacitación anticipada para manejar el sistema y poder contestar el test, en el cual se evaluó facilidad de aprendizaje, tiempo de respuesta, entendibilidad, atractivo.

Tabla 1-3: Detalle de preguntas Facilidad de aprendizaje

Pregunta	Si	No
¿Es fácil entender los cambios producidos en las operaciones?	22	3
¿Es fácil ingresar datos y ver sus resultados?	25	0
¿Se puede cerrar sesión de cualquier parte de la aplicación?	25	0
¿La terminología es constante en toda la aplicación?	23	2
¿Un mismo elemento aparece igual en toda la aplicación?	25	0
¿Se dan indicaciones para completar campos problemáticos?	24	1
¿Se identifican fácilmente las Figuras, las tablas, las zonas activas y el tipo de acción que se debe ejecutar?	24	1
Si una tarea tiene opciones por defecto, ¿están a disposición del usuario?	20	5
¿La aplicación no requiere volver a escribir la información solicitada anteriormente?	25	0
¿La herramienta soporta el aprendizaje colaborativo?	25	0
TOTAL:	238	12

Realizado Por: Darío Jiménez. 2017

Media Aritmética

Para poder analizar los datos obtenidos en el test con el apartado de facilidad de aprendizaje se debe realizar una media aritmética para encontrar el punto medio de las respuestas con sí y con no.

Ecuación media aritmética

$$\bar{X} = \sum_{i=1}^n \frac{x_i}{n}$$

Datos con respuestas si

$$\bar{X} = \frac{238}{10}$$

$$\bar{X} = 23.8$$

Datos con respuestas no

$$\bar{X} = \frac{12}{10}$$

$$\bar{X} = 1.2$$

Análisis: Después de haber obtenidos los datos de la media aritmética se puede decir que en cuanto a al aprendizaje tiende a ser fácil para los usuarios ya que solo se obtuvo un 1.2 repuestas negativas en general que apenas alcanza un 4.8% en las preguntas, como se puede observar en el gráfico 1-3 más detalladamente.



Gráfico 1-3: Facilidad de aprendizaje, Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

Tabla 2-3: Detalle de preguntas Tiempo de respuesta

Pregunta	Si	No
¿Es fácil de recordar como ejecutar una tarea en la herramienta?	24	1
¿Se ejecuta la página en un tiempo promedio de 3-7 segundos?	25	0

TOTAL:	49	1
---------------	----	---

Realizado Por: Darío Jiménez. 2017

Media Aritmética

Para poder analizar los datos obtenidos en el test con el apartado de tiempo de respuesta se debe realizar una media aritmética para encontrar el punto medio de las respuestas con sí y con no.

Ecuación media aritmética

$$\bar{X} = \sum_{i=1}^n \frac{x_i}{n}$$

Datos con respuestas si

$$\bar{X} = \frac{49}{2}$$

$$\bar{X} = 24.5$$

Datos con respuestas no

$$\bar{X} = \frac{1}{10}$$

$$\bar{X} = 0.5$$

Análisis: Después de haber obtenidos los datos de la media aritmética se puede decir que, en cuanto al tiempo de respuesta, tiende a tener respuestas que conforman al usuario y lo podemos comprobar con los resultados obtenidos donde solo un 0.5 de respuestas fueron negativas y que representa un 2% en las preguntas, como se puede observar en el gráfico 2-3 más detalladamente.



Gráfico 2-3: Tiempo de respuesta, Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

Tabla 3-3: Detalle de preguntas Entendibilidad

Pregunta	Si	No
¿Los mensajes emitidos son fáciles de entender?	25	0
¿El mensaje de error propone una solución?	23	2
¿Los tipos y tamaños de letra son legibles y distinguibles?	25	0
¿El tamaño de las imágenes e iconos es adecuado para saber qué representan?	25	0
¿El lenguaje está más cercano al utilizado por el usuario que al informático o técnico?	25	0
TOTAL:	123	2

Realizado Por: Darío Jiménez. 2017

Media Aritmética

Para poder analizar los datos obtenidos en el test con el apartado de entendibilidad se debe realizar una media aritmética para encontrar el punto medio de las respuestas con sí y con no.

Ecuación media aritmética

$$\bar{X} = \sum_{i=1}^n \frac{x_i}{n}$$

Datos con respuestas si

$$\bar{X} = \frac{123}{5}$$

$$\bar{X} = 24.6$$

Datos con respuestas no

$$\bar{X} = \frac{2}{10}$$

$$\bar{X} = 0.4$$

Análisis: Después de haber obtenidos los datos de la media aritmética se puede decir que en cuanto a la entendibilidad, el sistema tiende a ser fácil de entender para los usuarios ya que solo se obtuvo un 0.4 repuestas negativas en general que apenas alcanza un 0.32% en las preguntas, como se puede observar en el gráfico 3-3 más detalladamente.

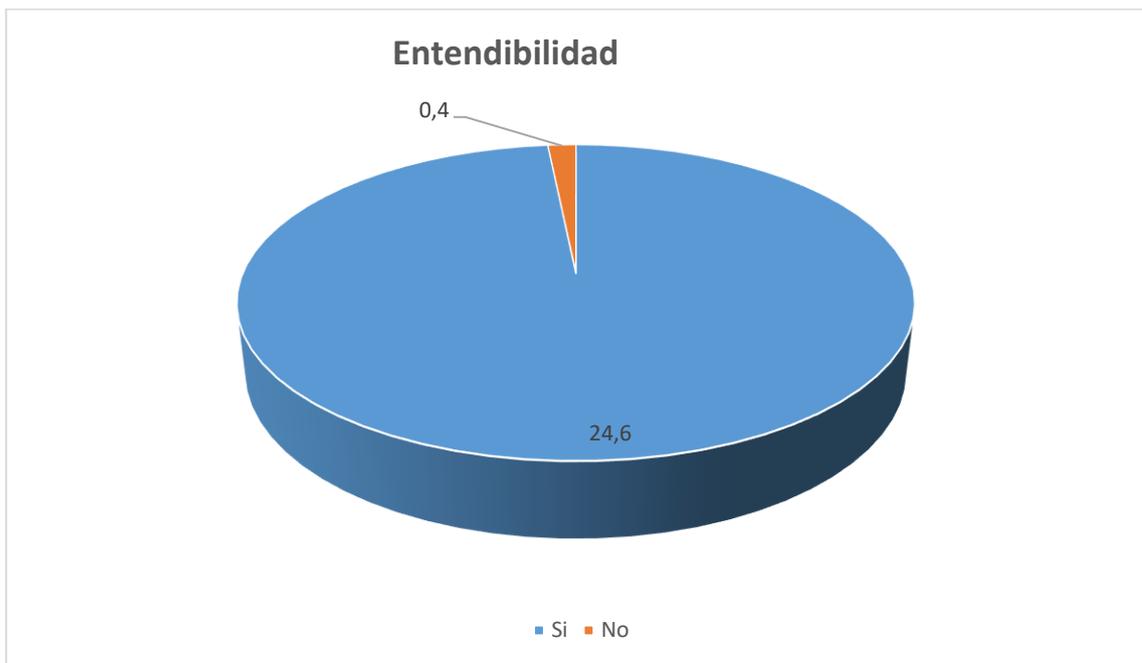


Gráfico 3-3: Entendibilidad, Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

Tabla 4-3: Detalle de preguntas Atractivo

Pregunta	Si	No
¿Los títulos son descriptivos y distintivos?	25	0
¿No despliega excesiva información en la interfaz de la aplicación?	25	0

¿Se utilizan las ventanas emergentes para mostrar información importante?	25	0
¿En los menús desplegables, los elementos de las listas son suficientemente explicativos y se visualizan todos a la vez?	24	1
¿La apariencia de la herramienta es agradable y sencilla?	24	1
TOTAL:	123	2

Realizado Por: Darío Jiménez. 2017

Media Aritmética

Para poder analizar los datos obtenidos en el test con el apartado de atractivo se debe realizar una media aritmética para encontrar el punto medio de las respuestas con sí y con no.

Ecuación media aritmética

$$\bar{X} = \sum_{i=1}^n \frac{x_i}{n}$$

Datos con respuestas si

$$\bar{X} = \frac{123}{5}$$

$$\bar{X} = 24.6$$

Datos con respuestas no

$$\bar{X} = \frac{2}{10}$$

$$\bar{X} = 0.4$$

Análisis: Después de haber obtenidos los datos de la media aritmética se puede decir que, en cuanto a la atraktividad, el sistema tiende a ser fácil y simple de entender para los usuarios ya que solo se obtuvo un 0.4 repuestas negativas en general que apenas alcanza un 0.32% en las preguntas, como se puede observar en la Figura 4-3 más detalladamente.

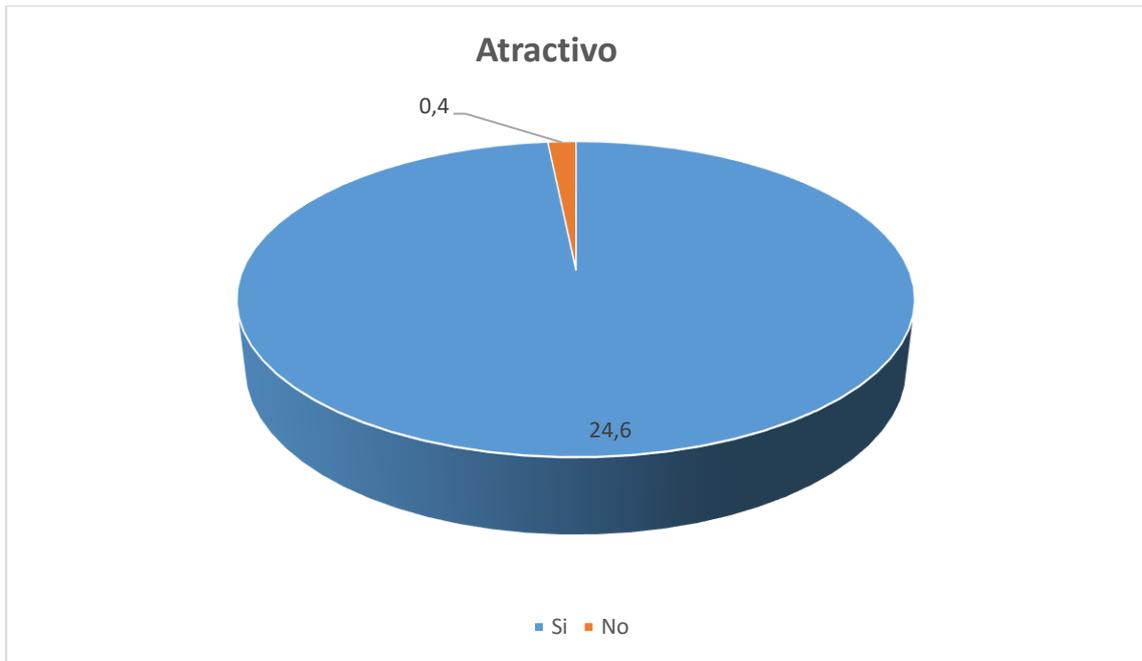


Gráfico 4-3: Atractivo, Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

3.2.1.3 Análisis de Usabilidad mediante atributos propuestos

Para poder analizar la usabilidad de acuerdo con los datos obtenidos en el análisis de cada uno de los atributos, se puede observar en la tabla 5-3 el detalle de los resultados.

Tabla 5-3: Detalle de Atributos Usabilidad

Atributo	Si	No
Facilidad de aprendizaje	23.8	1.2
Tiempo de respuesta	24.5	0.5
Entendibilidad	24.6	0.4
Atractivo	24.6	0.4
Total:	97.5	2.5

Realizado Por: Darío Jiménez. 2017

Media Aritmética

Para poder analizar los datos obtenidos de cada uno de los atributos se debe realizar una media aritmética para encontrar el punto medio de las respuestas con sí y con no.

Ecuación media aritmética

$$\bar{X} = \sum_{i=1}^n \frac{x_i}{n}$$

Datos con respuestas si

$$\bar{X} = \frac{97.5}{4}$$

$$\bar{X} = 24.37$$

Datos con respuestas no

$$\bar{X} = \frac{2.5}{4}$$

$$\bar{X} = 0.63$$

Análisis: Después de haber obtenidos los datos de la media aritmética se puede decir que la usabilidad de la aplicación está en un nivel muy alto, con el cual podemos decir que el sistema tiende a ser fácil y simple y llamativo para el usuario final ya que solo se obtuvo un 0.63 de respuestas negativas que apenas alcanza un 2.52% en las preguntas, como se puede observar en el gráfico 5-3 más detalladamente.

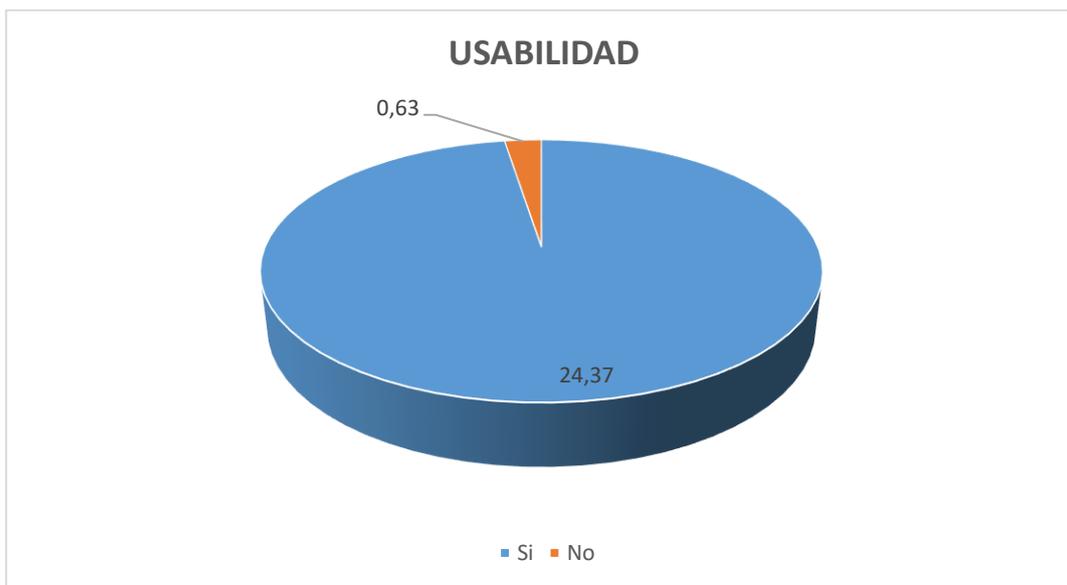


Gráfico 5-3: Usabilidad, Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

3.2.2 Análisis de usabilidad mediante la herramienta de crazy eggs

Para el análisis de la usabilidad mediante la herramienta crazy eggs se debe tener en cuenta que la herramienta analiza los siguientes puntos:

- **Heatmap.** – Mapa de calor que se construye en base a los clics.
- **Scrollmap.** – Mapa de scroll que se construye en base a los desplazamientos que se realizan en la página, mostrando las partes que más se ven.
- **Overlay:** Muestra el porcentaje de clics que representa un elemento sobre el total, aportando información adicional en función de distintas métricas.

3.2.2.1 Análisis HeatMap

Para el análisis del HeatMap se tomaron en cuenta a cuarenta y cinco personas quienes utilizaron el sistema durante un día, dos horas y cuarenta y cinco minutos. En la Figura 1-3 podemos observar los resultados obtenidos con la herramienta crazy eggs.

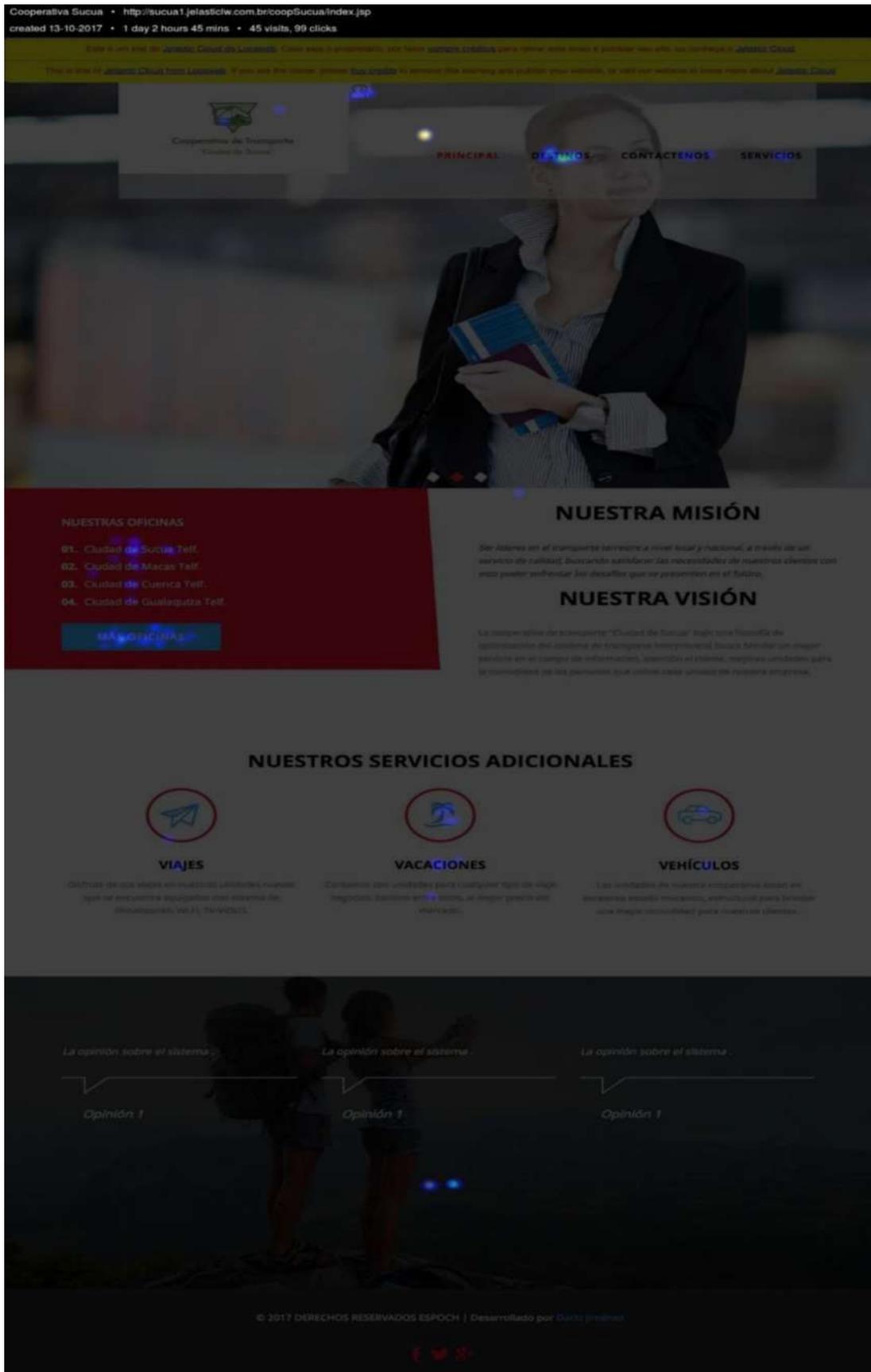


Figura 1-3: Resultados HeatMap, Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

Análisis. – Al observar los datos obtenidos en la Figura. 1-3 podemos decir que el mapa de calor (headMap) se ubican sobre todo en el encabezado, el grafico para el login y la información de las oficinas, siendo satisfactorio el uso ya que son los puntos en donde se implementó la funcionalidad del sistema.

3.2.2.2 Análisis Scrollmap

Para el análisis del Scrollmap se tomaron en cuenta a cuarenta y cinco personas quienes utilizaron le sistema durante un día, dos horas y cuarenta y cinco minutos. En la Figura 2-3 podemos observar los resultados obtenidos con la herramienta crazy eggs.



Figura 2-3: Resultados Scrollmap, Sistema “SoftSucua”

Realizado Por: Darío Jiménez. 2017

Análisis. – Al observar la Figura. 2-3 podemos decir que el sistema es de fácil desplazamiento con lo que no se dificulta a usuario aprenderlo a manejar.

3.2.2.3 Análisis Overlay

Para el análisis del overlay se tomaron en cuenta a cuarenta y cinco personas quienes utilizaron le sistema durante un día, dos horas y cuarenta y cinco minutos. En la Figura 3-3 podemos observar los resultados obtenidos con la herramienta crazy eggs.



Figura 3-3: Resultados Overlay, Sistema "SoftSucua"

Realizado Por: Darío Jiménez. 2017

Análisis. – La Figura. 3-3 nos presenta los clicks dados en la pantalla y donde se enfocaron los usuarios en pensar que hará algo, resultando muy cómodo para el usuario ya que no es de complejo aprendizaje.

CONCLUSIONES

- En la fase de planificación de la metodología de desarrollo de software se obtuvieron los requerimientos funcionales y no funcionales del sistema con un éxito total.
- El desarrollo del sistema se terminó con éxito teniendo un retraso mínimo para entregar el sistema.
- Al culminar el desarrollo del sistema y aplicando los atributos para la norma ISO/IEC 9126 se tuvo una aceptación del software del 97.48% por parte de los usuarios finales.
- Gracias a la herramienta crazy egg se verificó que la aplicación sí cumple con la norma de usabilidad y dando un visto bueno por parte del equipo de desarrollo.

RECOMENDACIONES

- Al terminar el sistema podemos recomendar al cliente que el sistema se lo ponga en producción.
- Se recomienda la implementación de un servidor físico propio para que la aplicación no de ningún problema y tener almacenada la información de la empresa.
- Para el futuro se recomienda generar facturación electrónica para mejor manejo de la parte contable de la empresa.
- Incentivar a los estudiantes a implementar los sistemas y que salgan a producción después de ser presentados en la tesis para aumentar plazas de trabajo.

BIBLIOGRAFIA

AGILE MANIFESTO. *Desarrollo metodología Ágil* [en línea]. Estados Unidos. [Consulta: 15 mayo 2017]. Disponible en <http://agilemanifesto.org>.

AGUIRRE BUENAÑO & MONCAYO ALVAREZ. *Análisis de frameworks MVC de Java para el desarrollo de aplicaciones web empresariales. Caso práctico: Sistema de Bienestar Politécnico* (tesis). (Ingeniería). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería en Sistemas. Riobamba-Ecuador. 2003. pp. 45-60.

ALAIMO, DIEGO MARTÍN. *Proyectos ágiles con Scrum: flexibilidad, aprendizaje, innovación y colaboración en contextos complejos*. Buenos Aires: EBook, 2013.

ANALÍTICA WEB. [en línea]. Mexico-Mexico D.F [Consulta: 17 enero 2017]. Disponible en: http://www.alzado.org/articulo.php?id_art=684.

ARAYA P. *Proyecto fortalecimiento de las capacidades tic en pymes y gobiernos locales mediante el uso de software libre*. (Material apoyo). Universidad nacional. Lima-Perú 2009 pp. 24-47

BERALDO, F. *¿qué es Crazy Egg y cómo funciona?* Brasil-Sao Paulo [en línea]. [Consulta: 30 enero 2017]. Disponible en: <http://blog.cicloagenciadigital.com.br/crazy-egg-o-que-e-e-como-funciona/>.

CASADO, A. *Crazy Egg: cómo funciona y qué ofrece la herramienta* [en línea] México. [Consulta: 30 enero 2017]. Disponible en: <http://www.kanlli.com/analisis-web-y-optimizacion/crazy-egg/>.

CORONEL, A. & JULISSA, L. Automatización del Control de los Procesos de registro – Turno de Buses, venta de Boletos y encomiendas de la Cooperativa Panamericana Internacional de la Provincia de el Oro [en línea] (tesis). (Pregrado), Universidad de Machala-Machala [Consulta: 12 diciembre 2016]. Disponible en: <http://repositorio.utmachala.edu.ec/handle/48000/1154>.

DEITEL, PAUL J. & HARVEY M. DEITEL. *Cómo programar en java. Séptima edición.* Mexico : PEARS ON EDUCACIÓN, 2008.

DÍAZ, O; et al. Desarrollo De Un Sistema De Gestión De Encomiendas, Reservación Y Venta De Tickets Para La Cooperativa De Transporte «Fbi». En La Ciudad De Babahoyo [en línea] (tesis). (Pregrado), [Consulta: 12 diciembre 2016]. Disponible en: <http://dspace.utb.edu.ec/handle/49000/923>.

ESPINOZA, P. & ALBERTO, F. Heurísticas de usabilidad para sitios web transaccionales. Pontificia Universidad Católica del Perú [en línea], [Consulta: 19 diciembre 2016]. Disponible en: <http://tesis.pucp.edu.pe/repositorio/handle/123456789/5399>.

HERNÁNDEZ, J. & ANDRÉS, J. Sistema informático para la gestión administrativa de las empresas de transporte interprovincial del terminal terrestre de la ciudad de Tulcán. [en línea] (tesis). (Pregrado), [Consulta: 12 diciembre 2016]. Disponible en: <http://localhost:8080/xmlui/handle/123456789/2157>.

ISO/IEC TR 9126-2:2003 *Software engineer—product quality—part 2: external metrics.*

MASCHERONI, M.A; et al. ESTAYNO, M.G. Calidad de software e ingeniería de usabilidad. *XIV Workshop de Investigadores en Ciencias de la Computación* [en línea]. [Consulta: 6 diciembre 2016]. ISBN 978-950-766-082-5. Disponible en: <http://hdl.handle.net/10915/19202>.

MEDINA L. *seguridad en redes* [en línea]. [Consulta: 19 diciembre 2016]. Disponible en: <http://seguridadredesmedina.blogspot.com/2009/10/md5-definicion-y-aplicaciones.html?m=1>.

NETBEANS.ORG. *Información NetBeans IDE 8.1. netbeans* [en línea]. [Consulta: 19 diciembre 2016]. Disponible en: https://netbeans.org/community/releases/61/index_es.html.

OBANDO, PATRICIA. *Curso básico de administración del sgbd postgresql*. [En línea]. [Consulta: 10 de diciembre de 2016.] Disponible en: <http://linuxsilo.net/articles/postgresql-pgpool.html>.

PHILIBERT, B. *Bootstrap 3: Le framework 100%* [en línea]. [Consulta: 2 enero 2017]. Disponible en: <http://www.eyrolles.com/Chapitres/9782212141320/9782212141320.pdf>.

SÁNCHEZ, JORGE. *Java2*. Mexico : PEARS ON EDUCACIÓN, 2004.

SANMARTIN, M. *DEFINICIÓN DEL SERVICIO DE TRANSPORTE DE CARGA* [en línea]. [Consulta: 6 diciembre 2016]. Disponible en: <https://transportedecargadepits.wordpress.com/2011/09/29/definicion-del-servicio-de-transporte-de-carga/>.

SCHWABER, K. & SUTHERLAND, J. *Scrum Basics. scrumguides* [en línea]. [Consulta: 2 enero 2017]. Disponible en: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>.

SOLIS JOHANNY. *¿Qué es Bootstrap y cómo funciona en el diseño web?* [en línea]. [Consulta: 6 diciembre 2016]. Disponible en: <http://www.arweb.com/chucherias/editorial/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web.htm>.

ANEXOS

Anexo A: Product backlog

Código	Historia de Usuario
HU_01	Generación de boletos
HU_02	Anulación de un boleto
HU_03	Ingreso de reservación por parte del cliente
HU_04	Anular una reservación por parte del empleado
HU_05	Generación de la hoja de ruta
HU_06	Generación de recibo por encomienda enviada
HU_07	Anulación de recibo por encomienda enviada
HU_08	Ingreso de paquete para envío
HU_09	Modificar un paquete para envío
HU_10	Modificar el estado de un paquete
HU_11	Ingreso de empleado
HU_12	Eliminar de empleados
HU_13	Modificar de empleados
HU_14	Búsqueda de empleados
HU_15	Reporte de empleados general
HU_16	Reporte de empleados por oficina
HU_17	Ingreso de cliente
HU_18	Eliminar de cliente
HU_19	Modificar de cliente
HU_20	Búsqueda de cliente
HU_21	Reporte de cliente general
HU_22	Ingreso de socio

HU_23	Eliminar de socio
HU_24	Modificar de socio
HU_25	Búsqueda de socio
HU_26	Reporte de socio general
HU_27	Ingreso de una oficina
HU_28	Eliminar una oficina
HU_29	Modificar una oficina
HU_30	Buscar una oficina
HU_31	Reporte de ingresos por ventas de boletos diario en cada oficina
HU_32	Reporte de ingresos por ventas de boletos mensual en cada oficina
HU_33	Reporte de ingresos por envío de encomiendas diarias en cada oficina
HU_34	Reporte de ingresos por envío de encomiendas mensuales en cada oficina
HU_35	Ingreso de unidad de transporté
HU_36	Eliminar de unidad de transporté
HU_37	Modificar de unidad de transporté
HU_38	Búsqueda de unidad de transporté
HU_39	Reporte de unidad de transporté general
HU_40	Ingreso de ruta
HU_41	Eliminación de ruta
HU_42	Modificación de ruta
HU_43	Búsqueda de ruta
HU_44	Reporte de rutas general
HU_45	Eliminar un paquete para envío
HU_46	Reporte de paquetes para envío
HU_47	Reporte de paquetes en bodega (salida)

HU_48	Reporte de paquetes entregados (llegados)
HU_49	Reporte de estado de paquetes por de numero de cedula
HU_50	Generación de guía de remisión

Realizado Por: Darío Jiménez. 2017

Anexo B: Estándar de codificación

ESTÁNDAR JAVA	
ACCIONES Y ELEMENTOS	REGLA
Sentencias de importación	Tras la declaración del paquete se incluirán las sentencias de importación de los paquetes necesarios. Esta importación de paquetes obligatorios seguirá el siguiente orden: <ol style="list-style-type: none"> 1. Paquetes del JDK de java. 2. Paquetes de utilidades no pertenecientes al JDK de Java. 3. Paquetes de la aplicación.
Declaraciones de clases e interfaces	<p>Variables de clase (estáticas) - En primer lugar, las variables de clase públicas (public), después las protegidas (protected), posteriormente las de nivel de paquete (sin modificador), y por último las privadas (private).</p> <p>Variables de instancia - Primero las públicas (public), después las protegidas (protected), luego las de nivel de paquete (sin modificador), y finalmente las privadas (private).</p> <p>Métodos - Deben agruparse por funcionalidad en lugar de agruparse por ámbito o accesibilidad. Por ejemplo, un método privado puede estar situado entre dos métodos públicos. El objetivo es desarrollar código fácil de leer y comprender.</p>
Sangría	Como norma general se establecen 4 caracteres como unidad de sangría. Los entornos de desarrollo integrado (IDE) más populares, tales como Eclipse o NetBeans, incluyen facilidades para formatear código Java.
Longitud de línea	La longitud de línea no debe superar los 80 caracteres por motivos de visualización e impresión
Espacios en blanco	Las líneas y espacios en blanco mejoran la legibilidad del código permitiendo identificar las secciones de código relacionadas lógicamente. Se utilizarán espacios en blanco en los siguientes casos:

	<p>1. Entre una palabra clave y un paréntesis. Esto permite que se distingan las llamadas a métodos de las palabras clave. Por ejemplo:</p> <pre>while (true) { ... }</pre> <p>2. Tras cada coma en un listado de argumentos. Por ejemplo:</p> <pre>Objeto.UnMetodo(a, b, c);</pre> <p>3. Para separar un operador binario de sus operandos, excepto en el caso del operador ("."). Nunca se utilizarán espacios entre los operadores unarios ("++" o "--") y sus operandos. Por ejemplo:</p> <pre>a += b + c; a = (a + b) / (c + d); contador++;</pre> <p>4. Para separar las expresiones incluidas en la sentencia "for". Por ejemplo:</p> <pre>for (expresion1; expresion2; expresion3)</pre> <p>5. Al realizar el moldeo o "casting" de clases. Ejemplo:</p> <pre>Unidad Unidad = (Unidad) Objeto;</pre>
Paquetes	<p>Se escribirán siempre en letras minúsculas para evitar que entren en conflicto con los nombres de clases o interfaces.</p> <p>El prefijo del paquete siempre corresponderá a un nombre de dominio de primer nivel el cual será com. El resto de componentes del paquete se nombrarán de acuerdo a nombre de la empresa, nombre del proyecto y nivel de capa de la aplicación o utilidad.</p> <p>Ejemplos:</p> <pre>com.shakeg.logicanegocio com.shakeg.datos com.shakeg.datos.nombredelatabla</pre>

	<pre>java.util.ArrayList; java.util.Date; javax.servlet.http.HttpServletRequest</pre>
Clases e interfaces	<p>Los nombres de clases deben ser sustantivos seguida de su nombre designado con la primera letra en mayúscula. Si el nombre es compuesto, cada palabra componente deberá comenzar con mayúsculas.</p> <ul style="list-style-type: none"> • Los nombres serán simples y descriptivos. • El nombre comienza con C de clase. • Debe evitarse el uso de acrónimos o abreviaturas. • Las interfaces se nombrarán siguiendo los mismos criterios que los indicados para las clases. Como norma general toda interfaz se nombrará con el prefijo "I" para diferenciarla de la clase que la implementa. <p>Ejemplo:</p> <pre>cCiudadano cOrganigramaDAO cAgendaService cAgendaService</pre>
Métodos	<p>Los métodos deben ser verbos escritos en minúsculas y la primera con mayúscula.</p> <p>Ejemplos:</p> <pre>capa1=accesoDatos public (tipo de dato) insertaUnidadCapa1(Unidad unidad); capa2=logicaNegocio Public (tipo de dato) insertaUnidadCapa2(Unidad unidad); public (tipo de dato) eliminaAgendaCapa1(Agenda agenda);</pre>
Variables	<p>Se definirán usando minúsculas que corresponde al nombre asignado.</p> <p>Las variables nunca podrán comenzar con caracteres (\$, &, %, etc.) o signos de puntuación. Los nombres de variables deben ser cortos y sus significados tienen que expresar con suficiente claridad la función que desempeñan en el código. Debe evitarse el uso de nombres de variables con un sólo carácter, excepto para variables temporales.</p>

	<p>Ejemplos:</p> <p>Unidad unidad;</p> <p>Integer contador;</p> <p>String nombre;</p> <p>Connection conexion;</p>
Objeto	<p>Todos los nombres de los objetos tendrán que escribirse Obj seguido del nombre de dicho objeto con mayúscula la primera letra. Cuando los nombres de los objetos sean compuestos, las palabras se escribirán unidas.</p> <p>Ejemplos:</p> <p>objUnidad;</p>
Visibilidad de atributos de instancia y de clase	<p>Los atributos de instancia y de clase serán siempre privados, excepto cuando tengan que ser visibles en subclases herederas, en tales casos serán declarados como protegidos.</p> <p>El acceso a los atributos de una clase se realizará por medio de los métodos "get" y "set" correspondientes, incluso cuando el acceso a dichos atributos se realice en los métodos miembros de la clase.</p> <p>Ejemplo:</p> <pre>public class cUnidad { private int id; private String nombre; ... public void actualizaUnidad(Unidad unidad) { this.setId(unidad.getId()); this.setNombre(unidad.getNombre()); } ... }</pre>
Asignación sobre variables	<p>Se deben evitar las asignaciones de un mismo valor sobre múltiples variables en una misma sentencia, ya que dichas sentencias suelen ser difíciles de leer.</p> <pre>int a = b = c = 2; // Evitar</pre>

	<p>No utilizar el operador de asignación en aquellos lugares donde sea susceptible de confusión con el operador de igualdad. Por ejemplo:</p> <pre>// INCORRECTO if ((c = d++) == 0) { }</pre> <p>// CORRECTO</p> <pre>c = d++; if (c == 0) { }</pre> <p>No utilizar asignaciones embebidas o anidadas. Ejemplo:</p> <pre>(c = 3) + 4 + d;</pre> <p>debería escribirse</p> <pre>c = 3; c = c + 4 + d;</pre>
Paréntesis	<p>Es una buena práctica el uso de paréntesis en expresiones que incluyan distintos tipos de operadores para evitar problemas de precedencia de operadores. Aunque la precedencia de operadores nos pueda parecer clara, debemos asumir que otros programadores no tengan un conocimiento exhaustivo sobre las reglas de precedencia.</p> <p>Ejemplo:</p> <pre>if (w == x && y == z) // INCORRECTO</pre> <pre>if ((w == x) && (y == z)) // CORRECTO</pre>
Valores de retorno	<p>Los valores de retorno tendrán que ser simples y comprensibles, de acuerdo al propósito y comportamiento del objeto en el que se utilicen.</p> <pre>public boolean esProgramador(Empleado emp) { NoProgramador = false; if (emp.getRol().equals(ROL_PROGRAMADOR)) { EsProgramador = true; } return EsProgramador; }</pre>

<p>Expresiones en el operador condicional ternario</p>	<p>Toda expresión compuesta, por uno o más operadores binarios, situada en la parte condicional del operador ternario deberán ir entre paréntesis.</p> <p>Ejemplo: $(x \geq y) ? x : y;$</p>
--	--

Realizado Por: Darío Jiménez. 2017

Anexo C: Modelo lógico

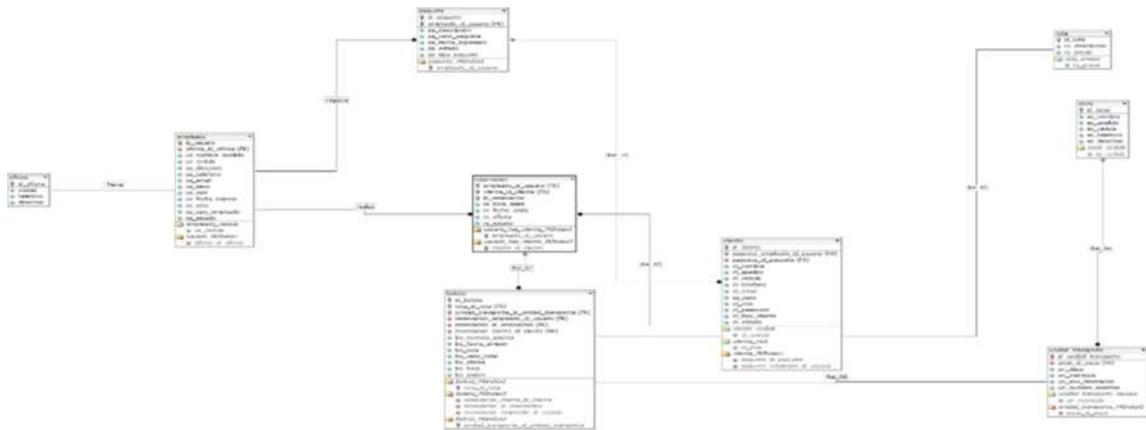
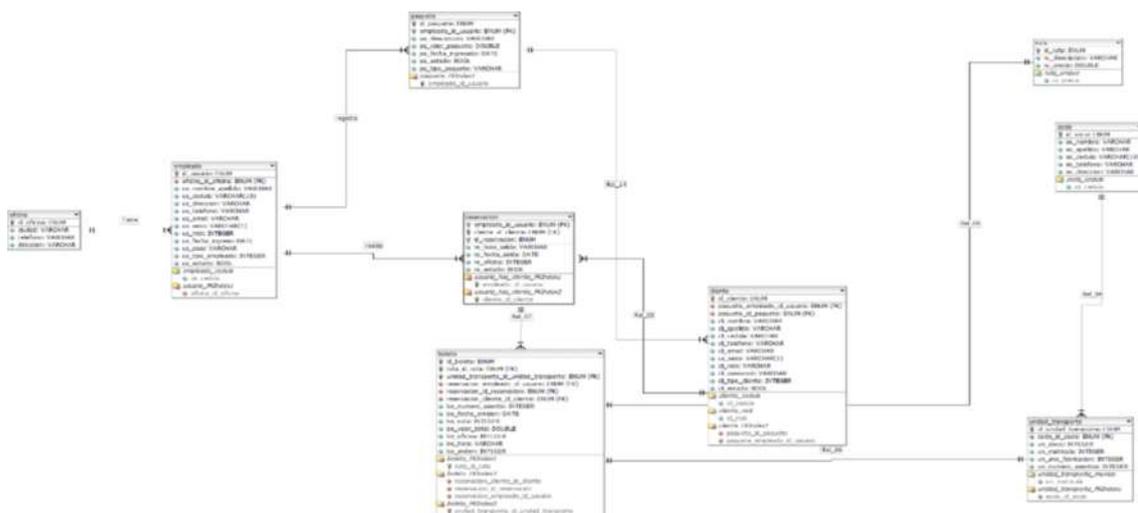


Gráfico 7-13. Modelo lógico, Sistema "SoftSucua"

Realizado Por: Dario Jimenez. 2017

Anexo C: Modelo físico



Modelo entidad-relacion, Sistema "SoftSucua"

Realizado Por: Darío Jiménez. 2017

Anexo C: Diccionario de datos

Anexo C: Datos Oficina

Nombre de Archivo: oficina		Fecha: 05/04/2017	
Descripción: Archivo Principal de la oficina, contendrá información de cada uno de ellos.			
Campo	Tipo	Tamaño	Descripción
<u>Id_oficina</u>	Serial	-	Clave
Ciudad	Carácter	-	ciudad de la oficina
Teléfono	Carácter	10	telefono del oficina
Dirección	Carácter	50	dirección del oficina
Estado	boolean	-	Estado del oficina
Relaciones:		Campos Clave: id_oficina (PK)	

Anexo C: Datos Empleado

Nombre de Archivo: empleado		Fecha: 05/04/2017	
Descripción: Archivo Principal de empleado, contendrá información detallada.			
Campo	Tipo	Tamaño	Descripción
<u>Id_usuario</u>	Serial	-	Identificador del empleado
us_nombre_apellido	Carácter	50	Nombre del empleado
us_cedula	Carácter	10	Cedula del empleado
us_telefono	Carácter	30	Teléfono del empleado
us_email	Carácter	30	Email del empleado
ussexo	Carácter	-	Pass del empleado
us_fecha_ingreso	Date	-	Fecha ingreso empleado
us_pass	Carácter	-	Contraseña empleado
us_tipo_empleado	integer	-	Tipo de estado del empleado
us_estado	boolean	-	Estado del empleado
us_nick	Carácter	50	Sobrenombre del empleado
oficina_id	Integer	-	Identificador oficina
Relaciones: oficina_id		Campos Clave: id_usuario	

Anexo C: Datos Unidad de transporte

Nombre de Archivo: unidad_transporte		Fecha: 05/04/2017	
Descripción: Archivo Principal de unidad_transporte, contendrá información del tipo_usuario			
Campo	Tipo	Tamaño	Descripción
<u>id_unidad_transporte</u>	Serial	-	Identificador de la unidad de transporte
un_disco	Integer	30	Disco de la unidad de transporte
un_ano_fabricacion	Integer	-	Año de fabricación de la unidad de transporte
un_numero_asientos	Integer	-	Numero de asientos de la unidad de transporte
un_matricula	Carácter	10	Matricula de la unidad de transporte
un_estado	Boolean	-	Estado de la unidad de transporte
socio_id	Integer	-	Identifiacador socio
Relaciones: socio_id		Campos Clave: id_unidad_transporte	

Anexo C: Datos Socio

Nombre de Archivo: socio		Fecha: 09/11/2016	
Descripción: Archivo Principal del detalle de la socio, contendrá información detallada.			
Campo	Tipo	Tamaño	Descripción
<u>Id_socio</u>	serial	-	Identificador del socio
so_nombre	Carácter	50	Nombre del socio
so_apellido	Carácter	50	Apellido del socio
so_cedula	Carácter	10	Cedula del socio
so_telefono	Carácter	10	Teléfono del socio
so_direccion	Carácter	50	Dirección del socio
so_estado	boolean	.-	Estado del socio
Relaciones:		Campos Clave: id_socio	

Anexo C: Datos Ruta

Nombre de Archivo: Ruta		Fecha: 05/04/2017	
Descripción: Archivo Principal de la Ruta, contendrá información de la ruta.			
Campo	Tipo	Tamaño	Descripción
<u>Id_ruta</u>	Serial	-	Identificador ruta
ru_descripcion	Carácter	50	Descripción de la ruta
ru_precio	Money	-	Precio de la ruta
ru_estado	Boolean	-	Estado de la ruta
ru_hora	Carácter	20	Hora de la ruta
codigo_ruta	Caarácter	50	Codigo de la ruta
tipo_ruta	Integer	-	Tipo de ruta
ru_origen	Carácter	50	Origen de la ruta
ru_destino	Carácter	50	Destino de la ruta
Relaciones:		Campos Clave: id_ruta	

Anexo C: Datos de la reservación

Nombre de Archivo: reservacion		Fecha: 05/04/2017	
Descripción: Archivo Principal de reservacion, contendrá información de reservacion.			
Campo	Tipo	Tamaño	Descripción
<u>Id_reservacion</u>	Serial	-	Identificador de reservación
empleado_id	Integer	-	Indentificador del empleado
cliente_id	Integer	-	Indentificador del cliente
ruta_id	Integer	-	Indentificador de la ruta
re_fecha_salida	Date	-	Fecha de reservación
re_numero_asiento	Integer	-	Numero de asiento de la reservación
re_estado_boleto	Carácter	1	Estado de reservación
re_estado	Boolean	-	Estado de la reservación
Relaciones: empleado_id, cliente_id, ruta_id		Campos Clave: id_reservacion	

Anexo C: Datos cliente

Nombre de Archivo: cliente	Fecha: 05/04/2017
-----------------------------------	--------------------------

Descripción: Archivo Principal de cliente, contendrá información detallada.			
Campo	Tipo	Tamaño	Descripción
<u>Id_cliente</u>	Serial	-	Identificador del cliente
cli_nombre	Carácter	50	Nombre del cliente
cli_apellido	Carácter	50	Apellido del cliente
cli_cedula	Carácter	10	Cedula del cliente
cli_telefono	Carácter	50	Telefono del cliente
cli_nick	Carácter	50	nick del cliente
cli_password	Carácter	50	Contraseña del cliente
cli_tipo_cliente	Integer	-	Tipo de Cliente
cli_estado	boolean	-	Estado del empleado
Relaciones:		Campos Clave: id_cliente	

Anexo C: Datos paquetes

Nombre de Archivo: paquete		Fecha: 05/04/2017	
Descripción: Archivo Principal de paquete, contendrá información detallada.			
Campo	Tipo	Tamaño	Descripción
<u>id_paquete</u>	Serial	-	Identificador del paquete
empleado_id	Integer	-	Identificador del empleado
cliente_id	Integer	-	Identificador del cliente
unidad_transporte_id	Integer	-	Identificador de la unidad
ruta_id	Integer	-	Identificador de la ruta
pa_descripcion	Carácter	50	Descripción del paquete
pa_subtotal_paquete	Money	50	Subtotal del paquete
pa_valor_paquete	Money	-	Valor de paquete
pa_estado	Boolean	-	Estado del paquete
pa_fecha_envio	Date	-	Fecha envio del paquete
pa_tipo_paquete	Carácter	-	Tipo de paquete
pa_estado_envio	Carácter	1	Estado envio paquete

pa_nombre_apellido_recibe	Caractér	50	Nombre y apellido del que recibe
pa_cedula_recibe	Caractér	10	Cedula del que recibe
Relaciones: empleado_id, cliente_id, unidad_transporte_id, ruta_id			
Campos Clave: id_paquete			

Anexo C: Datos boletos

Nombre de Archivo: boletos		Fecha: 05/04/2017	
Descripción: Archivo Principal de boletos, contendrá información detallada.			
Campo	Tipo	Tamaño	Descripción
<u>id_paquete</u>	Serial	-	Identificador del cliente
empleado_id	Integer	-	Identificador del empleado
cliente_id	Integer	-	Identificador del cliente
unidad_transporte_id	Integer	-	Identificador de la unidad
ruta_id	Integer	-	Identificador de la ruta
bo_numero_asiento	Integer	-	Numero del boleto
bo_valor_total	Money	-	Valor de boleto
bo_estado	Boolean	-	Estado del boleto
bo_fecha	Date	-	Fecha emisión boleto
bo_hora	Caractér	-	Hora del boleto
bo_anden	Integer	-	Anden del boleto
bo_estado_boleto	Caractér	1	Estado que se encuentra el boleto
Relaciones: empleado_id, cliente_id, unidad_transporte_id, ruta_id			
Campos Clave: id_boleto			

Anexo D: Sprint Backlog

Nombre de la tarea	Duración	Comienzo	Fin	Puntos estimados
SPRINT 0	20 días	lun 13/03/17	vie 07/04/17	
Metaforas del sistema		lun 13/03/17	vie 07/04/17	-

SPRINT 1	10 días	lun 10/04/17	vie 21/04/17	
Autenticación de usuarios		lun 10/04/17	lun 10/04/17	3 pts
Ingresar un empleado		lun 10/04/17	lun 10/04/17	3 pts
Modificar un empleado		mar 11/04/17	mar 11/04/17	6 pts
Eliminar un empleado		mie 12/04/17	mie 12/04/17	6 pts
Buscar un empleado		jue 13/04/17	jue 13/04/17	6 pts
Ingresar un cliente		vie 14/04/17	vie 14/04/17	6 pts
Modificar un cliente		lun 17/04/17	lun 17/04/17	6 pts
Eliminar un cliente		mar 18/04/17	mar 18/04/17	6 pts
Buscar un cliente		mie 19/04/17	mie 19/04/17	6 pts
Reporte de empleados general		jue 20/04/17	jue 20/04/17	3 pts
Reporte de cliente general		jue 20/04/17	jue 20/04/17	3 pts
Reporte de empleados por oficina		vie 21/04/17	vie 21/04/17	6 pts
SPRINT 2	10 días	lun 24/04/17	vie 05/05/17	
Ingresar un socio		lun 24/04/17	lun 24/04/17	6 pts
Modificar un socio		mar 25/04/17	mar 25/04/17	6 pts
Eliminar un socio		mie 26/04/17	mie 26/04/17	6 pts
Buscar un socio		jue 27/04/17	jue 27/04/17	6 pts
Ingresar una oficina		vie 28/04/17	vie 28/04/17	6 pts
Modificar una oficina		lun 01/05/17	lun 01/05/17	6 pts
Eliminar una oficina		mar 02/05/17	mar 02/05/17	6 pts
Buscar una oficina		mie 03/05/17	mie 03/05/17	6 pts
Reporte de socio general		jue 04/05/17	jue 04/05/17	3 pts
Reporte de ingresos por ventas de boletos diario en cada oficina		jue 04/05/17	vie 05/05/17	9pts

SPRINT 3	10 días	lun 08/05/17	vie 19/05/17	
Reporte de ingresos por ventas de boletos mensual en cada oficina		lun 08/05/17	lun 08/05/17	6 pts
Reporte de ingresos por envío de encomiendas diarias en cada oficina		mar 09/05/17	mar 09/05/17	6 pts
Reporte de ingresos por envío de encomiendas mensuales en cada oficina		mie 10/05/17	mie 10/05/17	6 pts
Ingresar una unidad de transporté		jue 11/05/17	jue 11/05/17	6 pts
Modificar una unidad de transporté		vie 12/05/17	vie 12/05/17	6 pts
Eliminar una unidad de transporté		lun 15/05/17	lun 15/05/17	6 pts
Buscar una unidad de transporté		mar 16/04/17	mar 16/05/17	6 pts
Ingresar una ruta		mie 17/05/17	mie 17/05/17	6 pts
Modificar una ruta		jue 18/05/17	jue 18/05/17	6 pts
Reporte de unidad de transporté general		vie 19/05/17	vie 19/05/17	3pts
Reporte de rutas general		vie 19/05/17	vie 19/05/17	3pts
SPRINT 4	10 días	lun 22/05/17	vie 02/06/17	
Eliminar una ruta		lun 22/05/17	lun 22/05/17	6 pts
Buscar una ruta		mar 23/05/17	mar 23/05/17	6 pts
Generación de boletos		mie 24/05/17	mie 31/05/17	30 pts
Anulación de un boleto		jue 01/06/17	jue 01/06/17	3 pts
Ingresar una reservación		jue 01/06/17	jue 01/06/17	3 pts
Anular una reservación		vie 02/06/17	vie 02/06/17	3 pts
Generar hoja de ruta		vie 02/06/17	vie 02/06/17	3 pts
SPRINT 5	10 días	lun 05/06/17	vie 16/06/17	
Generación de recibo por encomienda enviada		lun 05/06/17	lun 05/06/17	6 pts

Anulación de recibo por encomienda enviada		mar 06/06/17	mar 06/06/17	6 pts
Ingreso de paquete para envoi		mie 07/04/17	mie 26/04/17	6 pts
Modificar un paquete para envió		jue 08/04/17	jue 27/04/17	6 pts
Eliminar un paquete para envió		vie 09/04/17	vie 28/04/17	6 pts
Reporte de paquetes para envió		lun 01/05/17	lun 01/05/17	6 pts
Reporte de paquetes en bodega (salida)		mar 02/05/17	mar 02/05/17	6 pts
Reporte de paquetes entregados (llegados)		mie 03/05/17	mie 03/05/17	6 pts
Reporte de estado de paquetes por de numero de cedula		jue 03/05/17	jue 03/05/17	6 pts
Generación de guía de remisión		jue 04/05/17	vie 05/05/17	6 pts

anexo E: Encuesta sobre la usabilidad del software.

Test para medir la usabilidad

Nombre:

Ciudad:

Fecha:

Telefono:

	SI	No
Facilidad para entender los cambios producidos en las operaciones		
Facilidad para entender que datos ingresar y los resultados proporcionados		
Desde cualquier punto un usuario puede salir de la herramienta		
La terminología es constante en toda la herramienta		
Un mismo elemento aparece igual en toda la herramienta		
Se dan indicaciones para completar campos problemáticos		

Se identifican fácilmente las Figuras, las tablas, las zonas activas y el tipo de acción que se debe ejecutar		
Si una tarea tiene opciones por defecto, están a disposición del usuario		
La herramienta no requiere volver a escribir la información solicitada anteriormente		
La herramienta soporta el aprendizaje colaborativo		
Es fácil de recordar como ejecutar una tarea en la herramienta		
Se ejecuta la página en un tiempo promedio de 3-7 segundos		
Los mensajes emitidos son fáciles de entender		
El mensaje de error propone una solución		
Los tipos y tamaños de letra son legibles y distinguibles		
El tamaño de las imágenes e iconos es adecuado para saber qué representan		
El lenguaje está más cercano al utilizado por el usuario que al informático o técnico		
Los títulos son descriptivos y distintivos		
No despliega excesiva información en la interfaz de la herramienta.		
Se utilizan las ventanas emergentes para mostrar información importante		
En los menús desplegados, los elementos de las listas son suficientemente explicativos y se visualizan todos a la vez		
La apariencia de la herramienta es agradable y sencilla		

Realizado Por: Darío Jiménez. 2017

Firma