



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E  
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E  
INFORMÁTICOS**

**Tema:**

---

**Aplicación Web para la gestión documental de la Delegación Electoral de Bolívar.**

---

Trabajo de Graduación. Modalidad: TEMI. Trabajo Estructurado de Manera Independiente, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

Sublinea: Aplicaciones web

AUTOR: Richard Santiago Quinatoa Alvarez

TUTOR: Ing. Hernando Buenaño Mg.

Ambato - Ecuador

Enero/2015

## APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo de investigación sobre el tema: “**Aplicación Web para la gestión documental de la Delegación Electoral de Bolívar**”, del señor Richard Santiago Quinatoa Alvarez, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el Art. 16 del Capítulo II, del Reglamento de Graduación para obtener el título terminal de tercer nivel de la Universidad Técnica de Ambato.

Ambato Enero 07, 2015

EL TUTOR

---

Ing. Hernando Buenaño Mg.

## AUTORÍA

El presente trabajo de investigación titulado: **“Aplicación Web para la gestión documental de la Delegación Electoral de Bolívar”**. Es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato Enero 07, 2015

---

Richard Santiago Quinatoa Alvarez

CC: 0201928066

## **DEDICATORIA**

*EL presente trabajo de investigación le dedico  
A mi Madre por todo el esfuerzo, sacrificio, amor, comprensión  
y el apoyo incondicional q siempre me lo brindo.  
A mi amada esposa que ha sido un apoyo y confianza incondicional  
en este camino para alcanzar las metas que se veían como inalcanzables.*

*Richard Santiago Quinatoa Alvarez.*

## **AGRADECIMIENTO**

*Quiero agradecer a Dios por haberme dado la sabiduría  
para terminar este laborioso trabajo de investigación.*

*También quiero expresar un sincero agradecimiento a*

*Mi profesor Tutor de tesis Ing. Hernando Buenaño*

*Quien me ha guiado corregido y apoyado sin interés alguno.*

*Y como no agradecer a mi querida*

*Facultad de Facultad de Ingeniería en Sistemas, Electrónica e Industrial*

*Que me abrió las puertas para formarme académica y profesionalmente.*

*Richard Santiago Quinatoa Alvarez.*

## Índice

APROBACIÓN DEL TUTOR .....	ii
AUTORÍA .....	iii
DEDICATORIA.....	iv
AGRADECIMIENTO .....	v
RESUMEN EJECUTIVO .....	xii
SUMMARY.....	xiii
INTRODUCCIÓN .....	xiv
CAPÍTULO I.....	1
EL PROBLEMA .....	1
1.1. Tema:.....	1
1.2. Planteamiento del problema.....	1
1.3. Delimitación del Problema.....	2
1.4. Justificación.....	3
1.5. Objetivos .....	3
1.5.1. Objetivo General:.....	3
1.5.2. Objetivos Específicos: .....	4
CAPÍTULO II.....	1
MARCO TEÓRICO.....	1
2.1. Marco Teórico.....	1
2.1.1. Sistema de Gestión Documental.....	1
2.1.2. Aplicación Web.....	3
2.1.3. Servidor Web .....	3
2.1.4. Cliente Web.....	4
2.1.5. Base de Datos .....	5
• Características .....	5
2.1.6. Lenguaje de Programación Web.....	6
2.2. Propuesta de Solución .....	6
CAPÍTULO III .....	7
METODOLOGÍA .....	7
3.1. Modalidad de la investigación .....	7
3.2. Población y Muestra .....	7

3.3.	Recolección de Información .....	8
3.4.	Procesamiento y análisis de datos.....	8
3.5.	Desarrollo del Proyecto .....	8
CAPÍTULO IV .....		10
DESARROLLO DE LA PROPUESTA .....		10
4.1.	Recolección y Análisis de información en la Delegación Electoral de Bolívar. .	10
4.2.	Metodología .....	11
4.3.	Análisis de requisitos.....	12
4.3.1.	Requisitos Funcionales .....	12
4.3.2.	Requisitos de Hardware.....	13
4.4.	Diseño del Sistema .....	13
4.4.1.	Diagramas UML .....	14
•	Diagramas de Casos de uso.....	14
4.5.	Servidor Web .....	29
4.6.	Servidor HTTP Apache .....	29
4.7.	Diseño del Prototipo de la Aplicación Web.....	30
4.8.	Interfaz de la aplicación .....	30
•	Interfaz página principal del sistema.....	31
•	Interfaz Página de inicio del sistema.....	32
•	Interfaz Pagina de Cargos .....	33
•	Interfaz de página de departamentos .....	34
4.9.1.	Interfaz Pagina de usuario .....	35
•	Interfaz página creación de usuario .....	36
4.9.	Diseño de la Base de datos de la Aplicación .....	38
4.10.	Codificación .....	39
4.11.	Conceptualización.....	39
4.11.1.	Sistema de gestión de bases de datos.....	39
4.11.2.	MySQL .....	39
4.11.3.	Lenguaje de Programación .....	40
4.11.4.	PHP .....	40
•	Características .....	40
4.11.5.	Inconvenientes.....	42

4.12.	Diccionario de Datos .....	42
4.13.	Implantación del sistema .....	49
4.14.	Implementación de la programación .....	50
4.14.1.	La capa del Modelo .....	50
4.14.2.	La capa de la Vista .....	51
4.14.3.	La capa del Controlador.....	51
4.15.	Controlador de Ingreso a Departamento.....	52
4.16.	Controlador Departamento.....	53
4.17.	Controlador Cargo .....	55
4.18.	Controlador Oficio .....	57
4.19.	Controlador Recorrido.....	60
4.20.	Modelo Login.....	63
4.21.	Modelo Departamento .....	64
4.22.	Modelo Cargo.....	65
4.23.	Modelo Oficio .....	66
4.24.	Modelo Recorrido .....	67
4.25.	Vistas Departamento.....	68
4.26.	Vista Oficio .....	69
4.27.	Vista Recorrido .....	70
4.28.	Manual de usuario.....	72
4.29.	Pruebas de Funcionamiento .....	93
4.30.	Discusión de Resultados.....	97
CAPÍTULO V .....		99
CONCLUSIONES Y RECOMENDACIONES .....		99
5.1.	CONCLUSIONES .....	99
5.2.	RECOMENDACIONES .....	100
Bibliografía .....		101
Anexos .....		103



## Índice de Tablas

Tabla 4. 1 Casos de Uso Autenticarse .....	16
Tabla 4. 2 Casos de Uso: Agregar Usuario .....	17
Tabla 4. 3 Casos de Uso: Modificar Usuario .....	18
Tabla 4. 4 Casos de Uso: Eliminar Usuario.....	19
Tabla 4. 5 Casos de Uso: Acceder a Registros.....	20
Tabla 4. 6 Casos de Uso: Exportar Reporte.....	21
Tabla 4. 7 Casos de Uso: Acceder a Documentación .....	22
Tabla 4. 8 Casos de Uso: Imprimir Reportes .....	23
Tabla 4. 9 Casos de Uso: Agregar Usuario .....	25
Tabla 4. 10 Casos de Uso: Validar Contraseña .....	26
Tabla 4. 11 Cuadro comparativo de Lenguajes de Programación .....	27
Tabla 4. 12 Tabla Sistemas Administradores de Base de Datos .....	28
Tabla 4. 13 Descripción de la tabla Cargo .....	43
Tabla 4. 14 Descripción de la tabla Departamento .....	43
Tabla 4. 15 Descripción de la tabla Instancia .....	44
Tabla 4. 16 Descripción de la tabla Roles .....	44
Tabla 4. 17 Descripción de la tabla Movilización.....	46
Tabla 4. 18 Descripción de la tabla Oficio .....	46
Tabla 4. 19 Descripción de la tabla Permiso .....	47
Tabla 4. 20 Descripción de Permiso_Role .....	47
Tabla 4. 21 Descripción de la tabla Permiso_Usuario .....	48
Tabla 4. 22 Descripción de la tabla Recorido_Oficio .....	48
Tabla 4. 23 Descripción de la tabla Usuarios .....	49
Tabla 4. 24. Submenú del módulo registros .....	73

## Índice de Imágenes

Imagen 4. 1 Metodología Cascada .....	12
Imagen 4. 2 Sistema .....	15
Imagen 4. 3 Diagrama Agregar usuario al Sistema.....	24
Imagen 4. 4Diagrama Acceder al Sistema.....	26
Imagen 4. 5 Diseño del Prototipo de la interfaz .....	30
Imagen 4. 6 Página principal del sistema .....	31
Imagen 4. 7 Página de inicio del sistema.....	32
Imagen 4. 8 Página de cargos del sistema .....	33
Imagen 4. 9 Página de departamento del sistema.....	34
Imagen 4. 10 Página de Usuarios .....	35
Imagen 4. 11 Página de Roles .....	36
Imagen 4. 12 Diagrama Acceder en el Sistema.....	38
Imagen 4. 13 Diagrama de Modelo-Controlador-Vista .....	50
Imagen 4. 14 Árbol de Navegación.....	73
Imagen 4. 15 Interfaz de la Página de Inicio .....	74
Imagen 4. 16 Interfaz de la Pantalla de inicio de sesión.....	75
Imagen 4. 17 Interfaz de la Página Menú principal .....	75
Imagen 4. 18 Interfaz de la Página Departamentos .....	76
Imagen 4. 19 Interfaz de la página de Agregar Departamento.....	77
Imagen 4. 20 Editar departamento .....	78
Imagen 4. 21 Interfaz de la Pantalla de Cargo.....	79
Imagen 4. 22 Página Editar Cargo .....	80
Imagen 4. 23 Página Roles .....	81
Imagen 4. 24 Página Editar Rol.....	82
Imagen 4. 25 Página de usuarios .....	83
Imagen 4. 26 Interfaz de la Pantalla creación de usuario .....	84
Imagen 4. 27 Página editar usuario .....	85
Imagen 4. 28 Página de Oficios.....	86
Imagen 4. 29 Pagina Nuevo Oficio .....	87
Imagen 4. 30 Página Editar Oficio .....	88
Imagen 4. 31 Página Recorrido .....	89
Imagen 4. 32 Página Nuevo Recorrido .....	90
Imagen 4. 33 Página Editar recorrido .....	91
Imagen 4. 34 Página Reporte.....	92
Imagen 4. 35 Reporte Generado.....	92
Imagen 4. 36 Prueba de Caja Negra.....	93
Imagen 4. 37 Prueba Ingreso Usuario .....	94
Imagen 4. 38 Comprobación del Ingreso de Usuario.....	94
Imagen 4. 39 Prueba ingreso Rol .....	95
Imagen 4. 40 Comprobación ingreso de rol .....	95

Imagen 4. 41 Prueba Ingreso Nuevo Departamento .....	96
Imagen 4. 42 Comprobación Nuevo Departamento.....	96
Imagen 4. 43 Prueba Editar Departamento.....	97
Imagen 4. 44 Comprobación Editar Departamento.....	97

## **RESUMEN EJECUTIVO**

La Delegación Electoral de Bolívar es una institución pública designada para llevar de una manera transparente todos los procesos electorales, además de ayudar al ciudadano en cualquier trámite de carácter electoral.

La documentación generada dentro de la institución ha ido aumentando con el pasar del tiempo con la afluencia del ciudadano a realizar consultas de procesos electorales actuales y anteriores, además de realizar trámites personales como la obtención de la papeleta de votación y consulta de afinidad política.

El presente trabajo de investigación propone el desarrollo de una aplicación web la misma que permite gestionar la documentación de manera eficiente y con total transparencia.

Mediante un determinado proceso de levantamiento de requerimientos se ha podido realizar un diagrama de flujos donde se muestra los pasos que se llevaran a cabo en el desarrollo de la aplicación web para la gestión documental dentro de la institución.

Se definió las herramientas de desarrollo para poder crear una interfaz clara e intuitiva y se analizó los requerimientos específicos para la implementación y funcionamiento de dicha aplicación.

Una vez implementado el software necesario se podrá llevar a cabo pruebas de funcionamiento para el correcto desempeño de la Aplicación Web para la gestión documental de la Delegación Electoral de Bolívar.

## SUMMARY

The Electoral Delegation of Bolívar is designated to manage in a transparent mode all electoral processes, and help citizens in any proceeding of electoral public institution.

The documentation generated within the institution has been increasing over time with the influx of citizens to consult current and previous elections, in addition to personal matters such as obtaining the ballot and consultation of political affinity.

This paper proposes the development of a web application for managing the same documentation efficiently and transparently.

Through a particular process requirements gathering has been able to make a flow chart where the steps were carried out in developing web application for document management within the institution shown.

Development tools to create a clear and intuitive interface and the specific requirements for the implementation and operation of the application are defined analyzed.

Once implemented the necessary software to carry out tests may be operating for the proper performance of the Web application for document management Electoral Delegation of Bolívar.

## **INTRODUCCIÓN**

El presente trabajo de investigación cuyo tema es, “APLICACIÓN WEB PARA LA GESTIÓN DOCUMENTAL DE LA DELEGACIÓN ELECTORAL DE BOLÍVAR”, consta de cinco capítulos que se detallan a continuación:

Capítulo I denominado “EL PROBLEMA”, se identifica el problema a investigar, la justificación respectiva de la investigación y el planteamiento de los objetivos a obtener tras la culminación del trabajo de investigación.

Capítulo II contiene el “MARCO TEÓRICO”, en donde se establece el conjunto de conocimientos en los cuales se sustenta la investigación, además de presentar investigaciones previas que sirven de soporte a la investigación y se establece la propuesta de solución del problema.

Capítulo III comprende la “METODOLOGÍA”, en la cual se especifica la metodología de investigación a utilizar, el proceso de recolección de la información, el procesamiento y análisis de la información recabada y define las etapas para el desarrollo del proyecto.

Capítulo IV consta del “DESARROLLO DE LA PROPUESTA”, que detalla los requerimientos los casos de uso, diseño de la interfaz gráfica, diccionario de datos y la base de datos que ayudan al desarrollo de la aplicación.

Capítulo V Conclusiones y Recomendaciones”.- en este capítulo se detalla las conclusiones a las que llega el investigador y las recomendaciones que se definieron en el transcurso del desarrollo de la Aplicación Web.

Y por último se encuentran los anexos correspondientes al trabajo de investigación.

## **CAPÍTULO I**

### **EL PROBLEMA**

#### **1.1. Tema:**

Aplicación Web para la gestión documental de la Delegación Electoral de Bolívar.

#### **1.2. Planteamiento del problema**

El avance tecnológico ha llegado a todos los rincones del mundo generando así nuevas aplicaciones que son utilizadas en diferentes áreas profesionales, para cumplir objetivos según requerimientos o circunstancias, las mismas que se pueden llevar a cabo en varios dispositivos electrónicos ayudando a tener la información de manera más ordenada, más accesible y con gran facilidad de transferir y transportar.

Desde hace muchos años atrás la política y la democracia han sido temas que van de la mano y que necesitan ser tratadas con transparencia. Hoy en día el manejo de esta información debe ir de acuerdo a los avances tecnológicos que son de mucha ayuda para tener un manejo de información eficaz, rápida y segura para así mantener la credibilidad respecto a estos temas.

En Ecuador la entidad a cargo de gestionar esta información es el Consejo Nacional Electoral en el cual se deposita la confianza de los ecuatorianos para que se haga cumplir el derecho de democracia por lo cual se apoya con instituciones designadas en todas las provincias del Ecuador llamadas Delegaciones que tienen la información de interés; llevando un control adecuado de la información, las Delegaciones brindaran un servicio de calidad cumpliendo con los requerimientos de los usuarios de la mejor manera.

En Bolívar las instituciones públicas tiene conciencia tanto de la realidad de la organización como de la identidad corporativa interna, por lo que puede ser definida como el repertorio de procesos, mensajes y medios involucrados en la transmisión de información por parte de la organización; por tanto, no se refiere sólo a los mensajes, sino a los actos, al comportamiento mediante el cual todas las empresas transmiten información sobre su identidad, su misión, su forma de hacer las cosas y hasta sobre sus clientes.

La falta de interés de las autoridades en la gestión de información provoca pérdidas de tiempo y económicas las cuales afectan directamente a las funciones de los colaboradores de la Delegación Electoral de Bolívar.

El no tener personal capacitado provoca que éste lleve una inadecuada administración de la información interna que es de gran importancia para el avance del mismo, constituyendo una de las principales problemáticas a las que se enfrenta la entidad.

### **1.3. Delimitación del Problema**

De contenidos:

Área Académica: Software

Línea de Investigación: Desarrollo de software

Sublínea de Investigación: Aplicaciones Web



Espacial: Delegación Electoral de Bolívar

Temporal: 6 meses a partir de la aprobación del Proyecto

#### **1.4. Justificación**

El uso de las nuevas tecnologías de la informática y las comunicaciones, conocidas como TIC ha revolucionado en los últimos años la gestión y procesamiento de información, convirtiéndose así en un factor importante para todas las empresas y porque no decir de los países, omitiendo el procesamiento de información de forma manual. La automatización de procesos ocupa mayor relevancia en la actualidad, accediendo a una gestión documental más simple y dinámica. Las aplicaciones web tienen gran acogida por los entendidos en la materia, permitiendo una interacción entre el usuario y la aplicación más amigable. Con la gestión documental automatizada se puede llevar un control eficaz ahorrando tiempo y dinero, cabe recalcar que es una solución factible que no tiene impacto ambiental por lo cual es una solución ecológicamente aceptable.

Es factible el desarrollo de la presente investigación porque el personal que labora en la Delegación Electoral de Bolívar, está presta a colaborar con la información necesaria para llevar a cabo este proyecto que beneficiará en la agilización de los procesos de Gestión Documental que en ella se desarrollan.

El proyecto a realizar es de gran utilidad porque permitirá mejorar y agilizar el proceso de los documentos que se generan en delegación, además mejorará el desempeño diario del personal que labora en la institución.

#### **1.5. Objetivos**

##### **1.5.1. Objetivo General:**

- Desarrollar una Aplicación Web para la Gestión Documental en la Delegación Electoral de Bolívar.

#### **1.5.2. Objetivos Específicos:**

- Revisar los canales de comunicación existentes entre los departamentos internos de la Delegación Electoral de Bolívar.
- Analizar las deficiencias y errores de los canales de comunicación entre los departamentos internos de la Delegación Electoral de Bolívar.
- Diseñar la aplicación web para la gestión documental de la Delegación Electoral de Bolívar.
- Desarrollar de la aplicación web para la gestión documental de la Delegación Electoral de Bolívar.
- Realizar pruebas en la aplicación web para la gestión documental de la Delegación Electoral de Bolívar.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **2.1. Marco Teórico**

Después de haber indagado en la Biblioteca de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial se ha encontrado dos tesis con temas relacionados a la gestión documental con los siguientes temas:

La primera titulada: "Módulo de Gestión de clientes y seguridades bajo arquitectura SOA para el sistema administrativo contable de la empresa BESIXPLUS CIA LTDA". Autor: Luis Damián Laura Paucar. Año: 2012 y la segunda: "Gestión Documental y su influencia en los procesos de formación ciudadana del centro de Formación Ciudadana de Tungurahua" Autor: Adolfo Xavier Calle Gómez. Año: 2014

Además en el COBUEG (Consortio de Bibliotecas Universitarias del Ecuador) se encontró una tesis en la Universidad Politécnica del Ejército con el tema: Sistema cero papeles para la revisión de correspondencia del Banco COFIEC S.A. Autor: Raúl Fernando Garcés García. Año: 2012

##### **2.1.1. Sistema de Gestión Documental**

"Durante siglos, la gestión documental en las organizaciones fue el dominio exclusivo de administradores, archiveros y bibliotecarios, cuyas herramientas manuales básicas eran los libros de registro, las carpetas, archivadores, cajas y estanterías en que se guardan los documentos de papel (y más tarde los

audiovisuales y los documentos en soportes magnéticos u ópticos), los ficheros o kárdex que permiten hacer referencias cruzadas y una larga lista de técnicas de recuperación de información mediante sistemas de codificación y clasificación. Más recientemente se fueron sumando a ellos los informáticos, que son cada vez más necesarios debido a la complejidad y nivel de sofisticación que van alcanzando los sistemas computacionales de apoyo de la actividad administrativa. Aunque los informáticos benefician sustancialmente la gestión documental, aun los profesionales en sistemas de información son los expertos en los flujos de documentos y los procesos de cada documento de soporte papel o electrónico.

El uso del ordenador en la gestión documental se inicia en la práctica a partir de la experiencia de las grandes bibliotecas nacionales anglófonas, la Biblioteca del Congreso de los Estados Unidos de América y la British Library, que en los años 60 del siglo XX crean el formato bibliográfico MARC (Machine Readable Cataloguing) o catalogación legible por máquina, para sus bases de datos. Unos años más tarde el uso de las tecnologías de información y comunicación se hizo común en la administración pública y privada, con el uso generalizado de bases de datos y la aparición de los procesadores de textos y otras aplicaciones ofimáticas.

En la actualidad, existen en el mundo los más diversos sistemas de gestión documental: desde el simple registro manual de la correspondencia que entra y sale, hasta los más sofisticados sistemas informáticos que manejan no sólo la documentación administrativa propiamente tal, venga ella en papel o en formato electrónico, sino que además controlan los flujos de trabajo del proceso de tramitación de los expedientes, capturan información desde bases de datos de producción, contabilidad y otros, enlazan con el contenido de archivos, bibliotecas, centros de documentación y permiten realizar búsquedas sofisticadas y recuperar información de cualquier lugar. [1]

Los sistemas de gestión documental son aplicaciones que facilitan el tratamiento de documentos que maneja una empresa o una institución, con la finalidad de brindar eficiencia teniendo fácil acceso a la información, reduciendo el tiempo

de procesamiento tanto de almacenamiento como recuperación y manipulación, además que proporciona gran confiabilidad.

Otra gran ventaja es la disminución del consumo de recursos para la conservación de los documentos pudiendo ser compartida y reutilizada sin tener que gastar material físico adicional; optimiza el espacio físico pero asegura preservación histórica.

### **2.1.2. Aplicación Web**

Una aplicación web es un conjunto de páginas que interactúan unas con otras y con diversos recursos en un servidor web, incluidas bases de datos. Esta interacción permite implementar características en su sitio como catálogos de productos virtuales y administradores de noticias y contenidos. Adicionalmente podrá realizar consultas a bases de datos, registrar e ingresar información, solicitudes, pedidos y múltiples tipos de información en línea en tiempo real.

Los administradores de contenidos vía web almacenan los datos en Base de Datos. [2]

Hoy en día las Aplicaciones Web son las principales soluciones en software para las empresas en su gestión de información.

Una de las grandes ventajas de las aplicaciones Web son las actualizaciones debido a que estas se realizan en el servidor web y se encontrarían listas para ser consumida por los usuarios finales de la misma, además de casi no tener problemas de incompatibilidad con los sistemas operativos debido a que todos acceden desde un navegador.

### **2.1.3. Servidor Web**

El servidor web es un programa que está esperando permanentemente las solicitudes de conexión mediante el protocolo HTTP por parte de los clientes web.

La parte servidor de las aplicaciones web está formada por:

- Páginas estáticas (documentos HTML) que siempre muestran el mismo contenido.
- Recursos adicionales (multimedia, documentos adicionales, etc.) que se pueden emplear dentro de las páginas o estar disponibles para ser descargados y ejecutados (visualizados) en el cliente.
- Programas o scripts que son ejecutados por el servidor web cuando el navegador del cliente solicita algunas páginas. La salida de este script suele ser una página HTML estándar que se envía al navegador del cliente. Tradicionalmente este programa o script que es ejecutado por el servidor web se basa en la tecnología CGI. En algunos casos pueden acceder a bases de datos.[3]

Como su nombre lo dice Servidor Web está diseñado para alojar servicios a los clientes web mediante aplicaciones y una conexión dentro de la misma red.

#### **2.1.4. Cliente Web**

El cliente web es un programa con el que interacciona el usuario para solicitar a un servidor web el envío de los recursos que desea obtener mediante HTTP. Los clientes web también suelen actuar como clientes de transferencia de archivos (FTP), lectores de correo (SMTP y POP) y grupos de noticias (NNTP), etc.

Por tanto, la misión del cliente web es interpretar las páginas HTML y los diferentes recursos que contienen (imágenes, sonidos, etc.). Las tecnologías que se suelen emplear para programar el cliente web son:

- HTML.
- CSS.
- DHTML.
- Lenguajes de script: JavaScript, VBScript, etc.
- ActiveX.
- Applets programados en Java.[3]

El cliente web es la aplicación encargada de conectarse mediante peticiones al servidor web para gestionar la información a través de una red.

### 2.1.5. Base de Datos

La base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más **columnas** y **filas**. Las columnas guardan una parte de la información sobre cada elemento que queremos guardar en la tabla, cada fila de la tabla conforma un registro. [4]

Una base de datos es una serie de datos organizados y relacionados entre sí, los cuales son recolectados y consumidos por los sistemas de información de una empresa o entidad.

Estos datos son administrados mediante el servidor Web y enviados para contestar las peticiones de los clientes web mostrando así la información precisa al usuario final.

- **Características**

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.[4]

### **2.1.6. Lenguaje de Programación Web**

Los lenguajes de programación Web han ido surgiendo según las necesidades de las plataformas, intentando facilitar el trabajo a los desarrolladores de aplicaciones. Se clasifican en lenguajes del lado cliente y lenguajes del lado servidor. [5]

El HTML se ha convertido en uno de los lenguajes de programación web más importantes gracias a que la mayoría de los navegadores de Internet lo toleran bastante bien, es uno de los lenguajes más usados para la creación de documentos y es un lenguaje muy fácil de aprender.

De todas formas existen otros lenguajes de programación web que también se usan como partes o a veces acompañando o mejorando el contenido de las páginas web, entre ellos tenemos: CSS, hojas de estilo que mejoran la presentación del documento. JavaScript, lenguaje de programación web que permite darle efectos dinámicos a las páginas webs. PHP, es el más conocido y usado de los lenguajes de programación web de servidor. ASP y JSP, son dos lenguajes de programación web que actualmente está siendo muy usado. [6]

## **2.2. Propuesta de Solución**

La presente investigación propone desarrollar una aplicación Web para la gestión documental en la Delegación Electoral de Bolívar para solucionar el problema de los ineficientes canales de comunicación, permitiendo así optimizar los procesos y el tiempo de ejecución de los mismos.



## **CAPÍTULO III**

### **METODOLOGÍA**

#### **3.1. Modalidad de la investigación**

La investigación fue aplicable porque se dio solución y apoyo a la difusión de información de manera segura, inmediata y eficiente.

Para la realización de la investigación se utilizó las siguientes modalidades:

Modalidad de Investigación de Campo: la investigación tiene una orientación de estudio de procesos de trámites en el lugar donde existe la necesidad de la investigación, por esta razón el lugar de investigación se los realizó en el campo de la necesidad, que para este proyecto fue la Delegación Electoral de Bolívar.

Modalidad de Investigación Bibliográfica: la investigación demanda búsqueda de nuevos conceptos, tecnologías para cumplir con las necesidades de ampliar y profundizar e incluso actualizar los conceptos adquiridos para poder llegar a contrastar diferentes criterios, teorías de varios autores sobre el tema propuesto en la presente investigación.

#### **3.2. Población y Muestra**

La población o muestra no es requerido por qué no se tiene planteado una hipótesis para el desarrollo de esta investigación.

### **3.3. Recolección de Información**

Durante la recolección de información se realizó entrevistas tanto al personal administrativo como operativo de la Delegación Electoral de Bolívar, enfatizando los puntos más importantes que ayuden en el desarrollo de la aplicación, por medio de la misma se logró conocer más a fondo las actividades que se realizan internamente en la institución en cuanto a Gestión Documental se refiere.

Teniendo acceso también a un expediente donde se encuentran almacenados documentos, oficios e informes que han sido gestionados manualmente, por lo que se pudo identificar algunos procesos que deberían ser automatizados y optimizados.

### **3.4. Procesamiento y análisis de datos**

Para cumplir los objetivos planteados, se aplicará los siguientes métodos, procesos y/o actividades:

- Analizar la información recolectada en las fichas de información sobre los canales de comunicación existentes entre los departamentos internos de la Delegación Electoral de Bolívar.
- Clasificar la información obtenida mediante las fichas de observación en diferentes elementos: individuos, actividades, documentos, etc.
- Analizar la relación entre los diferentes elementos.

### **3.5. Desarrollo del Proyecto**

A continuación se detallan las Actividades que se realizaron, para cumplir los objetivos que llevaron a la obtención del producto final, teniendo en cuenta que la metodología utilizada es el Modelo en Cascada.

- Acudir el tiempo necesario a la Delegación Electoral de Bolívar hasta familiarizarse con las actividades que realiza el personal.

- Elaborar una guía de observación que permita reunir la información necesaria para detallar las actividades que se realizan en la institución.
- Procesar y analizar la información recogida con el fin de plasmar de forma clara mediante diagramas, todas las actividades que se realizan.
- Realizar el modelo de la base de datos para organizar la información, teniendo en cuenta elementos, actividades y documentos que se manejan.
- Realizar el diseño de la interfaz de la aplicación web, teniendo en cuenta la facilidad de manejo e interacción con el usuario.
- Determinar la arquitectura de la aplicación, en este caso una aplicación web cliente servidor.
- Determinar los requisitos hardware y software en los cuales se va a ejecutar la aplicación.
- Seleccionar las herramientas de desarrollo más apropiadas para la implementación de la aplicación web.
- Implementar el diseño de la base de datos, interfaces y código fuente mediante las herramientas de desarrollo seleccionadas.
- Realizar las correspondientes pruebas de funcionamiento de la aplicación para corregir errores en caso de ser necesario.
- Elaborar el informe final del presente proyecto de investigación.

## **CAPÍTULO IV**

### **DESARROLLO DE LA PROPUESTA**

#### **4.1. Recolección y Análisis de información en la Delegación Electoral de Bolívar.**

Utilizando la técnica de observación con su herramienta Ficha de Observación (Anexo 1) y la colaboración de los encargados de los Departamentos de Talento Humano y Administrativo se tuvo acceso a fichas, archivos y registros de los procesos que se realizan internamente obteniendo datos importante para ratificar la necesidad de una aplicación Web para la gestión documental demás apoyan directamente al desarrollo de la misma.

También se realizó una entrevista (Anexo 2) al encargado del departamento Administrativo y a Secretaria General con la finalidad de profundizar la información de los procesos que se llevan en la Delegación Electoral de Bolívar, encontrando así información muy útil para el investigador como apoyo para el desarrollo de la propuesta.

La forma de llevar sus procesos es manual por lo cual se presentan inconveniente con los datos o retrasos en las funciones respectivas. Entre los departamentos existentes hay una dependencia entre ellos según sea la gestión a realizarse, el principal departamento es la de Dirección por este tienen que pasar todos los tramites ya sea de funcionario o de la ciudadanía en general.

Por cada departamento existe un promedio de 2 funcionarios por lo cual se puede determinar así el número de usuarios de la aplicación, conociendo que

son únicamente dos funcionarios; con un usuario por departamento será suficiente para el uso.

Otro aspecto importante es la concurrencia diaria en la Delegación que se genera alrededor de 20 trámites diarios que son archivados de manera manual en carpetas físicas, hojas de cálculo, archivos de texto o en muebles de archivadores, por lo cual la búsqueda de un archivo es demorosa y cansada.

Los documentos que se genera en la Delegación Electoral de Bolívar son:

- Registro de ingreso y egreso de correspondencia
- Oficios con requerimientos de otras instituciones y de la ciudadanía
- Memorandos con requerimientos de las diferentes unidades del Consejo Nacional Electoral
- Documentos Certificados
- Informes de actividades, productos y servicios.

## **4.2. Metodología**

Para el desarrollo la aplicación Web se ha tomado la decisión de trabajar con metodología Cascada ya que esta se basa en una planificación total del proyecto para así poder desarrollar el software con rapidez, teniendo plazos y costos de una manera muy precisa dando satisfacción a los clientes en este caso a los usuarios de la Delegación Electoral de Bolívar.

El proceso en Cascada es más seguro por estar sujeto a la planificación inicial, y por ser un proyecto estático donde no se generan cambios en el desarrollo del software.

- **Etapas de la metodología Cascada**

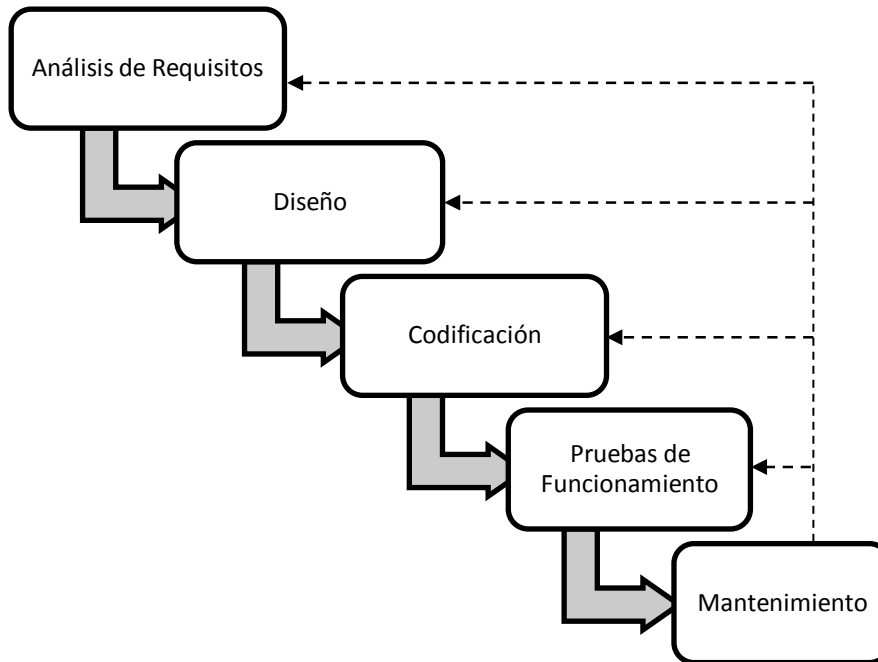


Imagen 4. 1 Metodología Cascada  
Elaborado por: El Investigador

### **4.3. Análisis de requisitos**

En esta fase se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada SRD (documento de especificación de requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos. Es importante señalar que en esta etapa se debe consensuar todo lo que se requiere del sistema y será aquello lo que seguirá en las siguientes etapas, no pudiéndose requerir nuevos resultados a mitad del proceso de elaboración del software

#### **4.3.1. Requisitos Funcionales**

- Ingresar información certera y fidedigna manteniendo la consistencia de los datos.

- Contar con normas de seguridad que den permisos a cada usuario según su nivel de acceso a la información y a los intereses de la administración.
- Disponer de la información del sistema de forma rápida y precisa.
- Permitir exportar reportes según los permisos que tenga cada usuario en el sistema.
- Disponer de una interfaz interactiva a fin de lograr una mayor interacción entre el usuario y el sistema.

### **4.3.2. Requisitos de Hardware**

Para determinar los requisitos de Hardware se realizaron pruebas en equipos informáticos con diferentes características, arrojando como resultado los requerimientos mínimos para el correcto funcionamiento de la Aplicación Web que son los siguientes:

- Sistema Operativo: Windows XP-Vista-7-8.
- Navegador Web: Preferentemente Mozilla Firefox .
- Ordenador: Pentium IV o superior.
- Conexión de red: Conexión de red por cable o wifi.

### **4.4. Diseño del Sistema**

Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras. Es conveniente distinguir entre diseño de alto nivel o arquitectónico y diseño detallado. El primero de ellos tiene como objetivo definir la estructura de la solución (una vez que la fase de análisis ha descrito el problema) identificando grandes módulos (conjuntos de funciones que van a estar asociadas) y sus relaciones. Con ello se define la arquitectura de la solución elegida. El segundo

define los algoritmos empleados y la organización del código para comenzar la implementación.

Es la fase en donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como también los análisis necesarios para saber que herramientas usar en la etapa de Codificación.

#### **4.4.1. Diagramas UML**

- **Diagramas de Casos de uso**

En el Lenguaje de Modelado Unificado, un diagrama de casos de uso es una forma de diagrama de comportamiento UML mejorado. El Lenguaje de Modelado Unificado (UML), define una notación gráfica para representar casos de uso llamada modelo de casos de uso. UML no define estándares para que el formato escrito describa los casos de uso, y así mucha gente no entiende que esta notación gráfica define la naturaleza de un caso de uso; sin embargo una notación gráfica puede solo dar una vista general simple de un caso de uso o un conjunto de casos de uso. Los diagramas de casos de uso son a menudo confundidos con los casos de uso. Mientras los dos conceptos están relacionados, los casos de uso son mucho más detallados que los diagramas de casos de uso. En los conceptos se debe detallar más de un caso de uso para poder identificar qué es lo que hace un caso de uso. En esta práctica es común crear especificaciones suplementarias para capturar detalles de requisitos que caen fuera del ámbito de las descripciones de los casos de uso. Ejemplos de esos temas incluyen restricciones de diseño como: rendimiento, temas de escalabilidad/gestión, o cumplimiento de estándares.

### **Diagrama de Sistema**



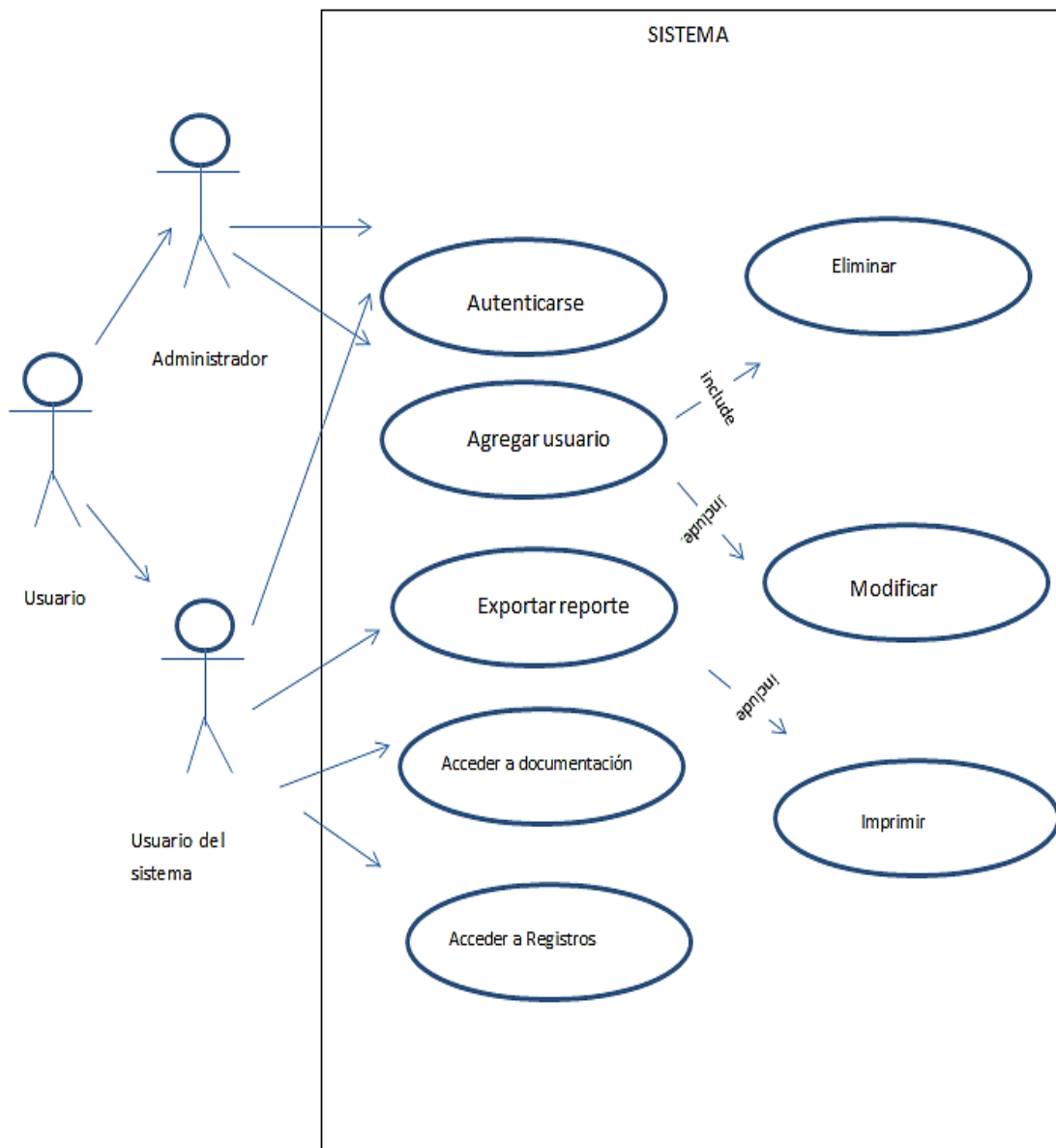


Imagen 4. 2 Sistema  
Elaborado por: El Investigador

- Caso de Uso Autenticarse

<b>Caso de Uso “Autenticarse”</b>
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>✓ Permite ingresar al sistema y visualizar el menú según el tipo de usuario conectado. Para poder acceder al sistema se debe contar con un usuario y una contraseña, de no existir alguno de los anteriores no podrá ingresar a la aplicación.</li> </ul>
<p><b>Actores:</b></p> <p>Usuarios registrados.</p>
<p><b>Precondiciones:</b></p> <p>El usuario debe registrarse en el sistema según su tipo de usuario.</p>
<p><b>Flujo Normal:</b></p> <ul style="list-style-type: none"> <li>✓ El actor escribe su nombre de usuario y contraseña válida.</li> <li>✓ Pulsar el botón de Inicio de sesión.</li> </ul>
<p><b>Flujo Alternativo:</b></p> <ul style="list-style-type: none"> <li>✓ El sistema comprueba la validez de los datos <ul style="list-style-type: none"> <li>○ Si los datos son correctos ingresa el usuario al sistema</li> <li>○ Si los datos son incorrectos envía un mensaje de error al usuario.</li> </ul> </li> </ul>
<p><b>Postcondiciones:</b></p> <p>El usuario ingresa a la página principal con el menú de las páginas que tiene acceso según su rol de usuario.</p>

Tabla 4. 1 Casos de Uso Autenticarse  
Elaborado por: El investigador

- Caso de Uso Agregar Usuario

Caso de Uso: Agregar Usuario
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>✓ Permite agregar un usuario del sistema.</li> </ul>
<p><b>Actores:</b></p> <ul style="list-style-type: none"> <li>✓ Administrador.</li> </ul>
<p><b>Flujo Normal:</b></p> <ul style="list-style-type: none"> <li>✓ Seleccionar el botón de Agregar.</li> <li>✓ Ingresar los datos y permisos del nuevo usuario.</li> <li>✓ Presionar el botón de <i>Guardar</i>.</li> </ul>
<p><b>Flujo Alternativo:</b></p> <ul style="list-style-type: none"> <li>✓ El sistema comprueba la validez de los datos <ul style="list-style-type: none"> <li>○ Si los datos son correctos agrega el nuevo usuario al sistema.</li> <li>○ Si los datos son incorrectos envía un mensaje de error al usuario.</li> </ul> </li> </ul>
<p><b>Postcondiciones:</b></p> <ul style="list-style-type: none"> <li>✓ Se crea un nuevo usuario para uso en el sistema.</li> </ul>

Tabla 4. 2 Casos de Uso: Agregar Usuario  
Elaborado por: El Investigador

- Caso de Uso Modificar Usuario

Casos de Uso: Modificar Usuario
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>✓ Permite modificar un usuario del sistema y agregar o quitar permisos.</li> </ul>
<p><b>Actores:</b></p> <ul style="list-style-type: none"> <li>✓ Administrador.</li> </ul>
<p><b>Flujo Normal:</b></p> <ul style="list-style-type: none"> <li>✓ Autenticarse</li> <li>✓ Seleccionar el botón de agregar usuario</li> <li>✓ Seleccionar el botón de modificar usuario</li> <li>✓ Presionar el botón modificar</li> <li>✓ Presionar el botón guardar</li> </ul>
<p><b>Flujo Alternativo:</b></p> <ul style="list-style-type: none"> <li>✓ El sistema comprueba la validez de los datos               <ul style="list-style-type: none"> <li>○ Si los datos son correctos modifica el usuario del sistema.</li> <li>○ Si los datos son incorrectos envía un mensaje de error al usuario.</li> </ul> </li> </ul>
<p><b>Postcondiciones:</b></p> <ul style="list-style-type: none"> <li>✓ Se modifican los datos de un usuario del sistema.</li> <li>✓ Se modifican los permisos de un usuario del sistema.</li> </ul>

Tabla 4. 3 Casos de Uso: Modificar Usuario  
Elaborado por: El Investigador

- Caso de Uso Eliminar Usuario

Casos de Uso: Eliminar Usuario	
<b>Descripción:</b>	<ul style="list-style-type: none"> <li>✓ Permite eliminar un usuario del sistema.</li> </ul>
<b>Actores:</b>	<ul style="list-style-type: none"> <li>✓ Administrador.</li> </ul>
<b>Flujo Normal:</b>	<ul style="list-style-type: none"> <li>✓ Autenticarse</li> <li>✓ Seleccionar el botón de Eliminar.</li> <li>✓ Presionar el botón de Eliminar.</li> </ul>
<b>Flujo Alternativo:</b>	<ul style="list-style-type: none"> <li>✓ El sistema comprueba la validez de los datos               <ul style="list-style-type: none"> <li>○ Si los datos son correctos elimina el usuario del sistema.</li> <li>○ Si los datos son incorrectos envía un mensaje de error al usuario.</li> </ul> </li> </ul>
<b>Postcondiciones:</b>	<ul style="list-style-type: none"> <li>✓ Se elimina un usuario del sistema.</li> </ul>

Tabla 4. 4 Casos de Uso: Eliminar Usuario  
Elaborado por: El Investigador

- Caso de Uso Acceder a Registros

Casos de Uso: Acceder a registros
<b>Descripción:</b> <ul style="list-style-type: none"> <li>✓ Permite al usuario acceder a los registros del sistema.</li> </ul>
<b>Actores:</b> <ul style="list-style-type: none"> <li>✓ Usuario del sistema.</li> </ul>
<b>Flujo Normal:</b> <ul style="list-style-type: none"> <li>✓ Autenticarse</li> <li>✓ Seleccionar el botón Registros</li> </ul>
<b>Flujo Alternativo:</b> <ul style="list-style-type: none"> <li>✓ El sistema comprueba la validez de los datos               <ul style="list-style-type: none"> <li>○ Si los datos son correctos permite al usuario acceder a los registros que contiene el sistema en sus bases de datos</li> <li>○ Si los datos son incorrectos envía un mensaje de error al usuario.</li> </ul> </li> </ul>
<b>Postcondiciones:</b> <ul style="list-style-type: none"> <li>✓ El usuario accede al registro del sistema</li> </ul>

Tabla 4. 5 Casos de Uso: Acceder a Registros.  
Elaborado por: El Investigador

- Caso de Uso Exportar Reporte

<b>Casos de Uso: Exportar reporte</b>
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>✓ Permite exportar reportes de la base de datos del sistema.</li> </ul>
<p><b>Actores:</b></p> <ul style="list-style-type: none"> <li>✓ Usuario del sistema.</li> </ul>
<p><b>Flujo Normal:</b></p> <ul style="list-style-type: none"> <li>✓ Autenticarse</li> <li>✓ Seleccionar el botón exportar reporte</li> <li>✓ Presionar el botón guardar reporte</li> </ul>
<p><b>Flujo Alternativo:</b></p> <ul style="list-style-type: none"> <li>✓ El sistema comprueba la validez de los datos <ul style="list-style-type: none"> <li>○ Si los datos son correctos el sistema exporta el reporte</li> <li>○ Si los datos son incorrectos envía un mensaje de error al usuario.</li> </ul> </li> </ul>
<p><b>Postcondiciones:</b></p> <ul style="list-style-type: none"> <li>✓ Se exporta un reporte personalizado de los datos que contiene el sistema.</li> <li>✓ Se guardan los datos exportados de la base de datos del sistema.</li> </ul>

Tabla 4. 6 Casos de Uso: Exportar Reporte  
Elaborado por: El Investigador

- Caso de Uso Acceder a Documentación

<b>Casos de Uso: Acceder a documentación</b>	
<b>Descripción:</b>	<ul style="list-style-type: none"> <li>✓ Permite al usuario acceder a la documentación que posee el sistema.</li> </ul>
<b>Actores:</b>	<ul style="list-style-type: none"> <li>✓ Usuario del sistema.</li> </ul>
<b>Flujo Normal:</b>	<ul style="list-style-type: none"> <li>✓ Autenticarse</li> <li>✓ Seleccionar el botón Documentación</li> </ul>
<b>Flujo Alternativo:</b>	<ul style="list-style-type: none"> <li>✓ El sistema comprueba la validez de los datos               <ul style="list-style-type: none"> <li>○ Si los datos son correctos permite al usuario acceder a la documentación que contiene el sistema en sus bases de datos</li> <li>○ Si los datos son incorrectos envía un mensaje de error al usuario.</li> </ul> </li> </ul>
<b>Postcondiciones:</b>	<ul style="list-style-type: none"> <li>✓ El usuario accede a la documentación del sistema</li> </ul>

Tabla 4. 7 Casos de Uso: Acceder a Documentación  
Elaborado por: El Investigador



- Caso de Uso Imprimir Reportes

Casos de Uso: Imprimir Reportes
<b>Descripción:</b> <ul style="list-style-type: none"> <li>✓ Permite imprimir un reporte personalizado del sistema.</li> </ul>
<b>Actores:</b> <ul style="list-style-type: none"> <li>✓ Usuario del sistema.</li> </ul>
<b>Flujo Normal:</b> <ul style="list-style-type: none"> <li>✓ Autenticarse</li> <li>✓ Seleccionar el botón exportar reporte</li> <li>✓ Seleccionar el botón imprimir reporte</li> </ul>
<b>Flujo Alternativo:</b> <ul style="list-style-type: none"> <li>✓ El sistema comprueba la validez de los datos               <ul style="list-style-type: none"> <li>○ Si los datos son correctos imprime un reporte personalizado del sistema.</li> <li>○ Si los datos son incorrectos envía un mensaje de error al usuario.</li> </ul> </li> </ul>
<b>Postcondiciones:</b> <ul style="list-style-type: none"> <li>✓ Se imprime un reporte del sistema.</li> </ul>

Tabla 4. 8 Casos de Uso: Imprimir Reportes  
Elaborado por: El Investigador

## Diagrama Agregar Usuario al Sistema

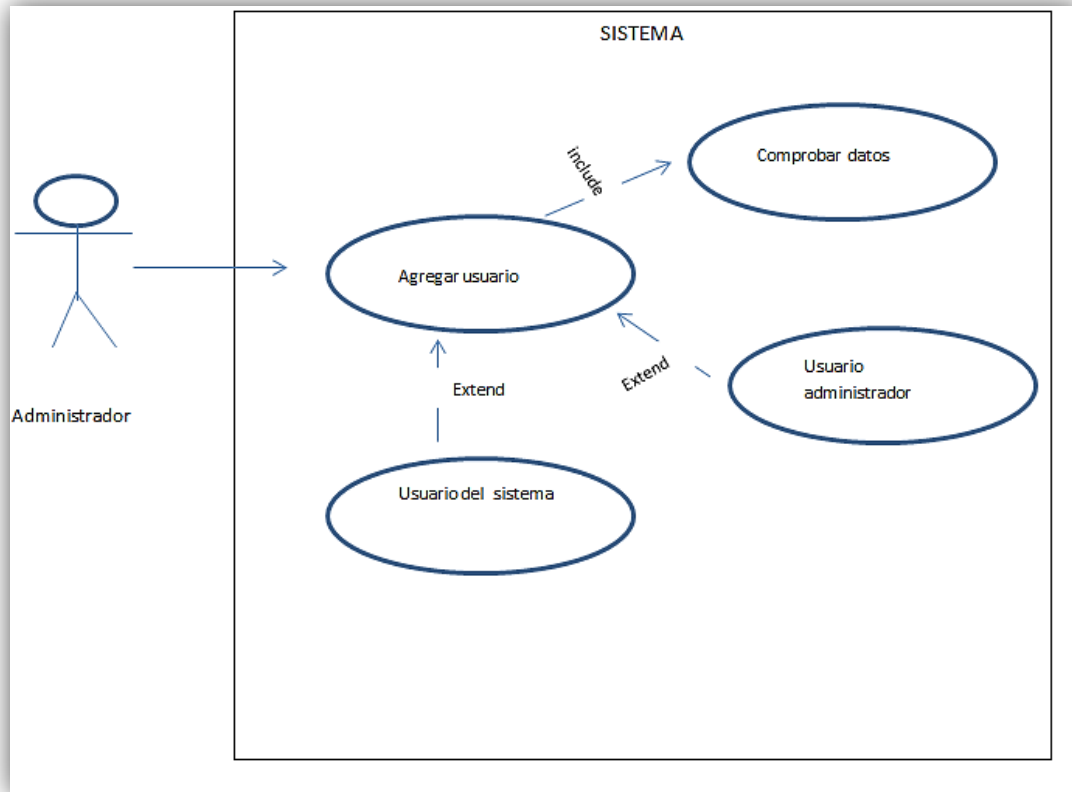


Imagen 4. 3 Diagrama Agregar usuario al Sistema  
Elaborado por: El Investigador

- Caso de Uso Agregar Usuario

<b>Casos de Uso: Agregar usuario</b>
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>✓ Permite agregar nuevos usuarios al sistema</li> </ul>
<p><b>Actores:</b></p> <ul style="list-style-type: none"> <li>✓ Administrador</li> </ul>
<p><b>Flujo Normal:</b></p> <ul style="list-style-type: none"> <li>✓ Autenticarse</li> <li>✓ Solicitar al sistema agregar nuevo usuario</li> <li>✓ Introducir los datos del nuevo usuario</li> <li>✓ El sistema comprueba si no existe un usuario con esos datos</li> <li>✓ El administrador puede asignar permisos al nuevo usuario de usuario del sistema o de administrador, en dependencia de la función que realice en la delegación.</li> <li>✓ Guardar los datos</li> </ul>
<p><b>Flujo Alternativo:</b></p> <ul style="list-style-type: none"> <li>✓ El sistema comprueba la validez de los datos (usuario y contraseña) <ul style="list-style-type: none"> <li>○ Si los datos son correctos permite al administrador agregar el nuevo usuario</li> <li>○ Si los datos son incorrectos envía un mensaje de error al usuario.</li> </ul> </li> </ul>
<p><b>Postcondiciones:</b></p> <ul style="list-style-type: none"> <li>✓ El administrador agrega nuevos usuarios al sistema</li> </ul>

Tabla 4. 9 Casos de Uso: Agregar Usuario  
Elaborado por: El Investigador

## Diagrama Acceder en el Sistema

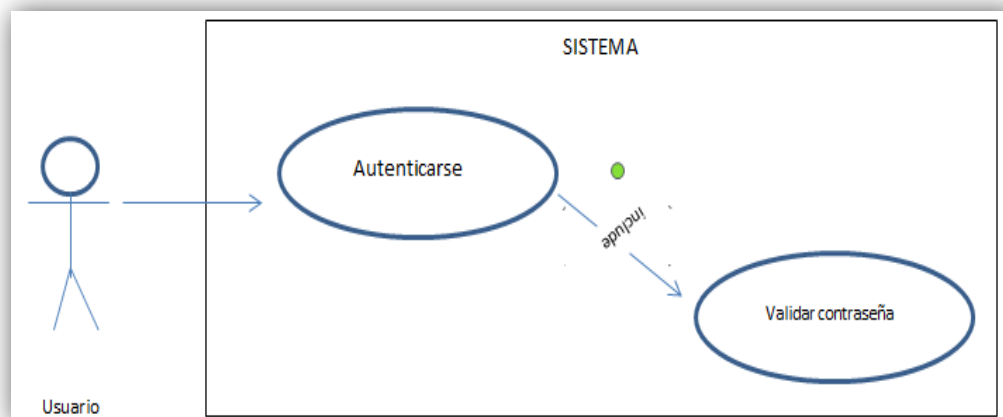


Imagen 4. 4Diagrama Acceder al Sistema  
Elaborado por: El Investigador

- Caso de Uso Validar Contraseña

Casos de Uso: Validar contraseña	
<b>Descripción:</b>	✓ Permite al sistema comprobar si realmente el usuario existe
<b>Actores:</b>	✓ Usuarios
<b>Flujo Normal:</b>	✓ Autenticarse
<b>Flujo Alternativo:</b>	✓ El sistema comprueba la validez de los datos (usuario y contraseña) <ul style="list-style-type: none"><li>○ Si los datos son correctos permite al usuario acceder a los registros que contiene el sistema en sus bases de datos</li><li>○ Si los datos son incorrectos envía un mensaje de error al usuario.</li></ul>
<b>Postcondiciones:</b>	✓ El usuario accede al registro del sistema

Tabla 4. 10 Casos de Uso: Validar Contraseña  
Elaborado por: El Investigador

#### 4.5. Cuadro Comparativo de Lenguajes de Programación

Características	PHP	ASP.net	JavaScript
Dificultad de Programación	Baja	Alta	Media
Software Libre	Si	No	Si
Licencia	Libre	Pagada	Libre
Lenguajes base de sintaxis	C y C++	VB.net o J# y C#	Java, Python, C y Perl
Multiplataforma	Si	Si	No
Ejecución de Scripts	Independientes	Independiente	Limitados
Seguridad de Código	Alta	Alta	Vulnerable
Base de Datos	MysSQL, PostgreSQL, Oracle, entre otras	MySQL y PostgreSQL	No maneja Base de Datos
Finalidad	Aplicaciones Web Dinámicas.	Aplicación Web Dinámica.	Validación de datos. Mejoras interfaz de usuarios.
Información	Fuentes de información variadas.	Fuentes de información de Windows, segura y confiable.	Fuentes de información variadas.
Recursos de Operación	Bajos	Altos	Altos
Orientado a Objetos	Si	Si	Si (Limitado)
Extensión	PHP	ASP	Cualquiera

Tabla 4. 11 Cuadro comparativo de Lenguajes de Programación  
Elaborado por: El Investigador

Una vez realizado el análisis entre los diferentes lenguajes de desarrollo de Software se ha llegado a la decisión de usar PHP como lenguaje de programación para la aplicación Web siendo el principal motivo para su elección el ser Software Libre y no tener costos económicos para la Delegación Electoral de Bolívar además de cumplir con las especificaciones necesarias para el desarrollo de la Aplicación Web.

#### 4.6. Cuadro Comparativo Sistemas Administradores de Base de Datos

Características	MySQL	SQL Server	ORACLE
Creador	MySQL AB	Microsoft	Global Development Group
Soporte de Sistema Operativo	Linux, IOS Mac, Windows	Microsoft	Linux, IOS Mac, Windows
ACID	Tiene limitaciones	Si	Si
Integridad Referencial	Necesita mecanismo externos	Si	Si
Transacciones	Necesita mecanismos externos	Si	Si
Unicode	Si	Si	Si
Interfaz	SQL	GUI, SQL, Varios	GUI, SQL
Lenguaje Soportado	Many, including C, C#, C++, D, Java, Ruby, and Objective C	Java, Ruby, Python, VB, .Net, and PHP	Many, including C, C#, C++, Java, Ruby, and Objective C
Sistema Operativo	Windows, Linux, OS X, FreeBSD, Solaris	Windows	Windows, Linux, Solaris, HP-UX, OS X, z/OS, AIX
Licencias	Libre	Propietario	Propietario

Tabla 4. 12 Tabla Sistemas Administradores de Base de Datos  
Elaborado por: El Investigador

#### **4.7. Diseño de la aplicación**

Para el diseño de esta aplicación se emplean las herramientas WAMP

WAMP es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Windows, como sistema operativo
- Apache, como servidor web
- MySQL, como gestor de bases de datos
- PHP (generalmente), Perl, o Python, como lenguajes de programación.

El uso de un WAMP permite servir páginas html a internet, además de poder gestionar datos en ellas, al mismo tiempo un WAMP, proporciona lenguajes de programación para desarrollar aplicaciones web.

#### **4.5. Servidor Web**

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se usa el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa. [7]

#### **4.6. Servidor HTTP Apache**

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual.

Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. [8]

#### 4.7. Diseño del Prototipo de la Aplicación Web

Este prototipo fue construido para tener una visión temprana en el desarrollo de la interfaz proporcionándonos una retroalimentación de la Aplicación Web



Imagen 4. 5 Diseño del Prototipo de la interfaz  
Elaborado por: El Investigador

#### 4.8. Interfaz de la aplicación

La interfaz gráfica de usuario, conocida también como GUI. Es un programa informático que actúa de tal forma que el usuario se puede comunicar con la máquina, y comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar.

Utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina. [9]



La interfaz gráfica de la aplicación se diseñó con un estilo sencillo, a fin de posibilitar la interactividad de los usuarios con el sistema. Se emplearon combinaciones de colores fuertes con fondos blancos.

- **Interfaz página principal del sistema**



Imagen 4. 6 Página principal del sistema  
Elaborado por: El Investigador

Página principal del sistema, presenta el sistema al usuario con una interfaz detallada así:

1. **Inicio.-** En esta sección está ubicada el menú principal de la aplicación el mismo que se activará las demás opciones cuando el usuario inicie la sesión.
2. **Inicio de Sesión.-** nos lleva a la página para ingresar usuario y contraseña
3. **Logo.-** Esta sección está ubicado el logo de la página web que representa a la institución.
4. **Contenido.-** En esta sección esta una leyenda de inicio y presentación de la aplicación.

- **Interfaz Página de inicio del sistema**

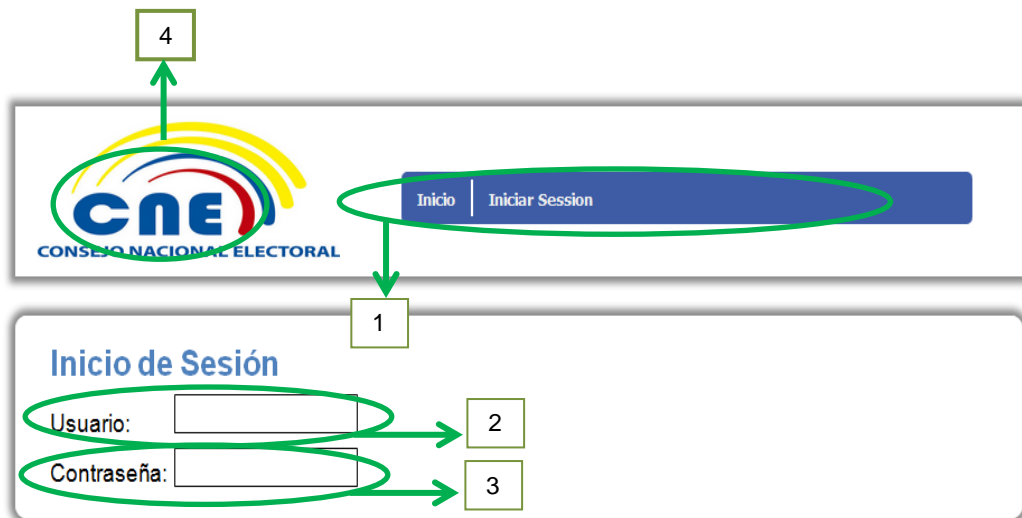


Imagen 4. 7 Página de inicio del sistema  
Elaborado por: El Investigador

Permite al usuario ingresar al sistema siempre que inserte los datos correctamente y que este habilitado en la base de datos con su correspondiente contraseña, solamente dejándolo ingresar a los sitios a los cuales tiene permitido el acceso.

1. **Inicio.-** En esta sección está ubicada el menú principal de la aplicación el mismo que se activará las demás opciones cuando el usuario inicie la sesión.
2. **Usuario.-** En este capo el usuario ingresa su identificación de usuario para iniciar su sesión
3. **Contraseña.-** Ayuda a la autenticación del usuario para permitir inicio de la sesión
4. **Logo.-** Esta sección está ubicado el logo de la página web que representa a la institución.

- **Interfaz Pagina de Cargos**

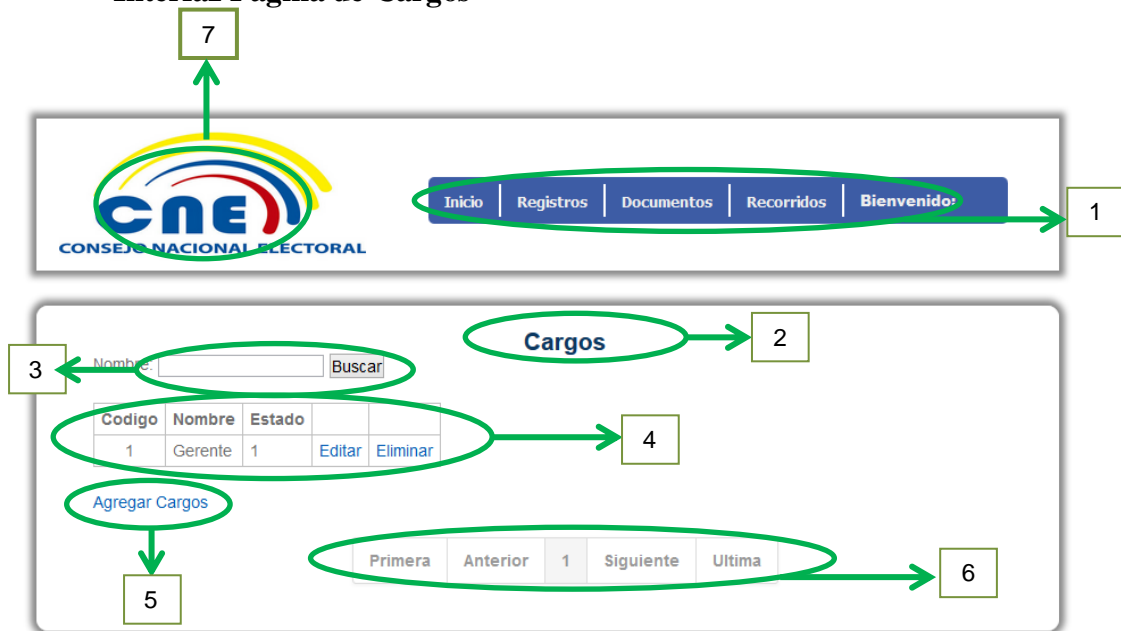


Imagen 4. 8 Página de cargos del sistema  
Elaborado por: El Investigador

Cada usuario esta contenido dentro de la plantilla de cargos, lo cual aumenta o disminuye los permisos en el sistema. En esta pantalla podemos editar o eliminar cargos, limitando así los permisos que posee cada usuario en el sistema

1. **Inicio.-** En esta sección está ubicada el menú principal de la aplicación ya iniciada la sesión se muestra únicamente las opciones a las cuales el usuario tiene acceso.
2. **Nombre de la aplicación.-** Identificador de la pantalla
3. **Campo se Búsqueda.-** Este campo ayuda al usuario a buscar un cargo específico.
4. **Tabla de datos.-** Se despliega los registros ingresados y guardados correctamente.
5. **Agregar campos.-** link para acceder a otra pantalla necesaria para ingresar los nuevos cargos.
6. **Barra de desplazamiento.-** Ayuda a desplazarse de página a página puede ser de una por una, al inicio o fin.
7. **Logo.-** Esta sección está ubicado el logo de la página web que representa a la institución.

- Interfaz de página de departamentos



Imagen 4.9 Página de departamento del sistema  
Elaborado por: El Investigador

Cada departamento contiene personal, y cada personal es un usuario en el sistema, de esta forma podremos saber a qué parte del sistema se puede acceder desde cada departamento y que privilegios tendrán estos usuarios en el sistema.

1. **Inicio.-** En esta sección está ubicada el menú principal de la aplicación ya iniciada la sesión se muestra únicamente las opciones a las cuales el usuario tiene acceso.
2. **Nombre de la aplicación.-** Identificador de la pantalla
3. **Tabla de datos.-** Se despliega los registros ingresados y guardados correctamente.
4. **Barra de desplazamiento.-** Ayuda a desplazarse de página a página puede ser de una por una, al inicio o fin.
5. **Agregar campos.-** link para acceder a otra pantalla necesaria para ingresar los nuevos departamentos.
6. **Campo de Búsqueda.-** Este campo ayuda al usuario a buscar un departamento específico.

7. **Logo.-** Esta sección está ubicada el logo de la página web que representa a la institución.

#### 4.9.1. Interfaz Pagina de usuario



Imagen 4. 10 Página de Usuarios  
Elaborado por: El Investigador

El menú usuarios permite agregar, eliminar o editar a los usuarios existentes en la base de datos del sistema, siendo necesario estar registrado para poder ingresar al mismo.

1. **Inicio.-** En esta sección está ubicada el menú principal de la aplicación ya iniciada la sesión se muestra únicamente las opciones a las cuales el usuario tiene acceso.
2. **Nombre de la aplicación.-** Identificador de la pantalla
3. **Tabla de datos.-** Se despliega los registros ingresados y guardados correctamente.
4. **Barra de desplazamiento.-** Ayuda a desplazarse de página a página puede ser de una por una, al inicio o fin.
5. **Agregar campos.-** link para acceder a otra pantalla necesaria para ingresar los nuevos departamentos.

6. **Campo de Búsqueda.-** Este campo ayuda al usuario a buscar un departamento específico.
7. **Logo.-** Esta sección está ubicado el logo de la página web que representa a la institución.

- **Interfaz página creación de usuario**

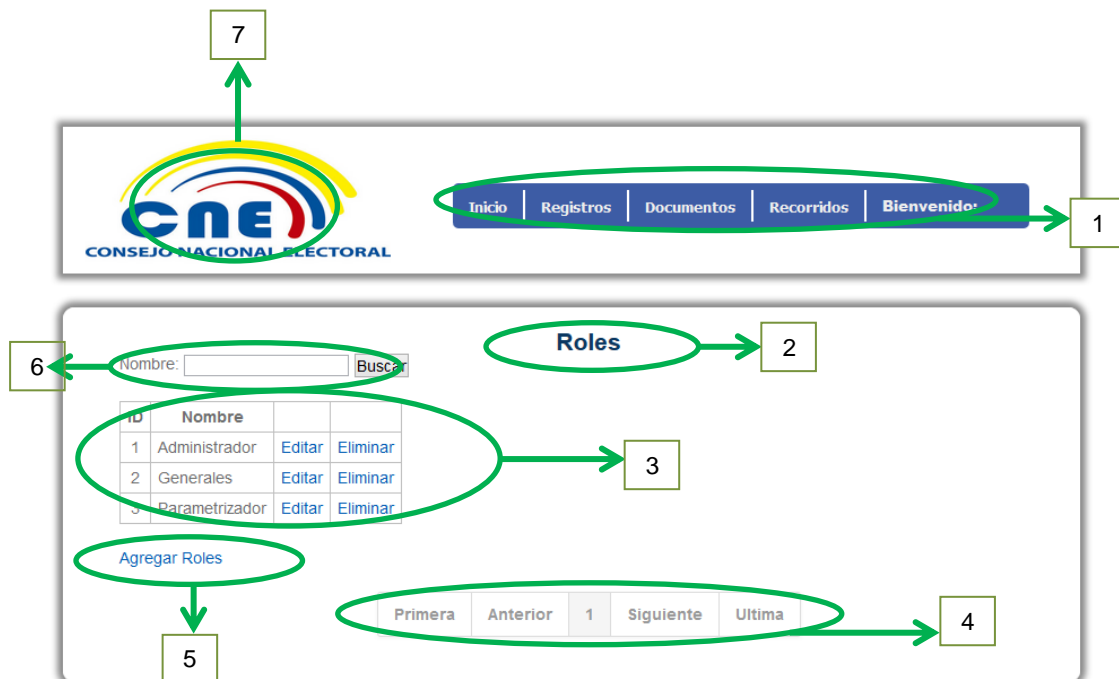


Imagen 4. 11 Página de Roles  
Elaborado por: El Investigador

La página de roles permite administrar los tipos de usuarios que acceden al sistema.

1. **Inicio.-** En esta sección está ubicada el menú principal de la aplicación ya iniciada la sesión se muestra únicamente las opciones a las cuales el usuario tiene acceso.
2. **Nombre de la aplicación.-** Identificador de la pantalla
3. **Tabla de datos.-** Se despliega los registros ingresados y guardados correctamente.

4. **Barra de desplazamiento.-** Ayuda a desplazarse de página a página puede ser de una por una, al inicio o al fin.
5. **Agregar campos.-** link para acceder a otra pantalla necesaria para ingresar los nuevos roles.
6. **Campo se Búsqueda.-** Este campo ayuda al usuario a buscar un rol específico.
7. **Logo.-** Esta sección está ubicado el logo de la página web que representa a la institución.

#### 4.9. Diseño de la Base de datos de la Aplicación

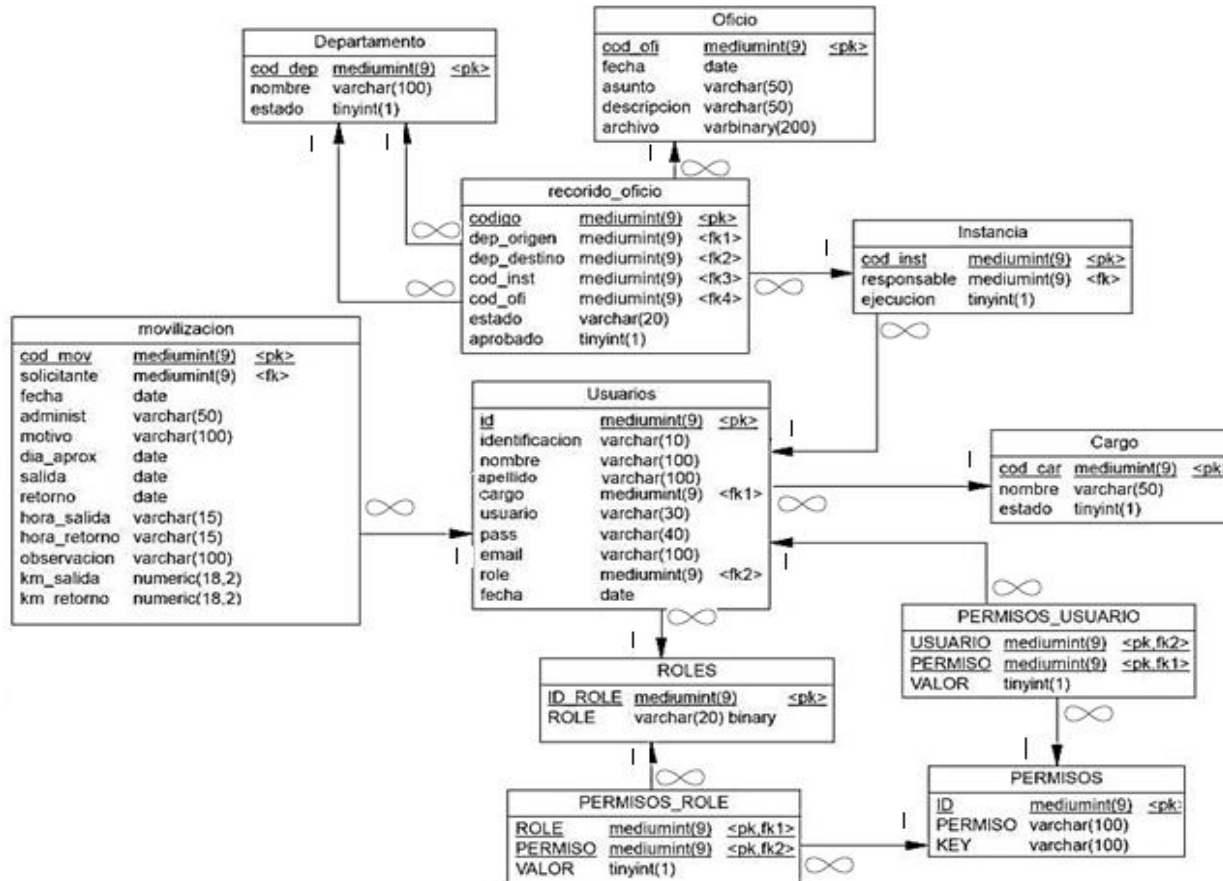


Imagen 4. 12 Diagrama Acceder en el Sistema  
Elaborado por: El Investigador



## **4.10. Codificación**

Es la fase en donde se implementa el código fuente, haciendo uso de prototipos así como de pruebas y ensayos para corregir errores. Dependiendo del lenguaje de programación y su versión se crean las bibliotecas y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.

## **4.11. Conceptualización**

### **4.11.1. Sistema de gestión de bases de datos**

Un sistema de gestión de bases de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto. [10]

### **4.11.2. MySQL**

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C. [11]

### **4.11.3. Lenguaje de Programación**

Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras.

Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. [12]

### **4.11.4. PHP**

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo. [13]

- **Características**

- ✓ Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- ✓ Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.
- ✓ El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su

resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.

- ✓ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- ✓ Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- ✓ Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Permite aplicar técnicas de programación orientada a objetos. Incluso aplicaciones como Zend framework, empresa que desarrolla PHP, están totalmente desarrolladas mediante esta metodología.
- ✓ No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- ✓ Tiene manejo de excepciones (desde PHP5).

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal.

#### **4.11.5. Inconvenientes**

- ✓ Como es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código pero no necesariamente impide que el código sea examinado.
- ✓ Debido a que es un lenguaje interpretado, un script en PHP suele funcionar considerablemente más lento que su equivalente en un lenguaje de bajo nivel, sin embargo este inconveniente se puede minimizar con técnicas de caché tanto en archivos como en memoria.
- ✓ Las variables al no ser tipificadas dificulta a los diferentes IDEs para ofrecer asistencias para el tipificado del código, aunque esto no es realmente un inconveniente del lenguaje en sí. Esto es solventado por Zend Studio añadiendo un comentario con el tipo a la declaración de la variable. [14]

#### **4.12. Diccionario de Datos**

Un diccionario de datos es un conjunto de metadatos que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El diccionario de datos guarda los detalles y descripción de todos estos elementos. [15]

<b>TABLA: cargo</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>	<b>RESTRICCIONES</b>
Código	mediumint(9)	Código del Cargo	Clave Primaria
Nombre	varchar(100)	Nombre del Cargo	Requerido
Estado	tinyint(1)	Estado del Cargo	Requerido

Tabla 4. 13 Descripción de la tabla Cargo  
Elaborado por: El Investigador

<b>TABLA: departamento</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>	<b>RESTRICCIONES</b>
Código	mediumint(9)	Código del Departamento	Clave Primaria
Nombre	varchar(100)	Nombre del Departamento	Requerido
Estado	tinyint(1)	Estado del Departamento	Requerido

Tabla 4. 14 Descripción de la tabla Departamento  
Elaborado por: El Investigador

<b>TABLA: instancia</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>	<b>RESTRICCIONES</b>
Código	mediumint(9)	Código de la Instancia	Clave Primaria
Responsable	mediumint(9)	Nombre de la Instancia	Requerido
Ejecución	tinyint(1)	Estado en el que se encuentra la Instancia	Requerido
Fecha	date	Fecha de la Instancia	Requerido

Tabla 4. 15 Descripción de la tabla Instancia  
Elaborado por: El Investigador

<b>TABLA: roles</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>	<b>RESTRICCIONES</b>
Id_role	mediumint(9)	Identificación del rol	Clave Primaria
role	Varchar(20)	Nombre del rol	Requerido

Tabla 4. 16 Descripción de la tabla Roles  
Elaborado por: El Investigador

**TABLA: movilización**

<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>	<b>RESTRICCIONES</b>
Cod_mov	mediumint(9)	Código de Movilización	Clave Primaria
Solicitante	mediumint(9)	Nombre de la Movilización	Requerido
Fecha	Date	Fecha de la Movilización	Requerido
Administ	Varchar(50)	Persona que emite la Movilización	Requerido
Motivo	Varchar(100)	Motivo de la Movilización	Requerido
Dia_Aprox	Date	Día de la Movilización	Requerido
Salida	Date	Salida de la Movilización	Requerido
Retorno	Date	Retorno de la Movilización	Requerido
Hora_Salida	Varchar(15)	Hora de Salida de Movilización	Requerido
Hora_Retorno	Varchar(15)	Hora de Retorno de la Movilización	Requerido
Observacion	Varchar(100)	Observación sobre la Movilización	Requerido
Km_Salida	Decimal(18,2)	Kilometraje de Salida del vehículo correspondiente a la Movilización	Requerido
Km_Retorno	Decimal(18,2)	Kilometraje de Retorno del vehículo correspondiente a la	Requerido

		Movilización	
--	--	--------------	--

Tabla 4. 17 Descripción de la tabla Movilización  
Elabora por: El Investigador

<b>TABLA: oficio</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>	<b>RESTRICCIONES</b>
Cód_Ofi	mediumint(9)	Código del Oficio	Clave Primaria
Fecha	date	Fecha del Oficio	Requerido
Asunto	Varchar(50)	Asunto del Oficio	Requerido
Descripción	Varchar(50)	Descripción del Oficio	Requerido
Archivo	Vabunary(200)	Oficio en archivo	Requerido

Tabla 4. 18 Descripción de la tabla Oficio  
Elaborado por: El Investigador



<b>TABLA: permiso</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>	<b>RESTRICCIONES</b>
id	mediumint(9)	Identificación del Permiso	Clave Primaria
permiso	mediumint(9)	Nombre del Permiso	Requerido
Key	Varchar(100)	Clave de Permiso	Requerido

Tabla 4. 19 Descripción de la tabla Permiso  
Elaborado por: El Investigador

<b>TABLA: permisos_role</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>	<b>RESTRICCIONES</b>
role	mediumint(9)	Identificación del rol	Clave Primaria
permiso	mediumint(9)	Nombre del Permiso al que hace referencia rol	Calve Foránea
valor	tinyint(1)	Valor Permiso_Rol	Requerido

Tabla 4. 20 Descripción de Permiso\_Role  
Elaborado por: El Investigador

<b>TABLA: permisos_usuario</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>	<b>RESTRICCIONES</b>
usuario	mediumint(9)	Identificación del usuario	Clave Primaria
permiso	mediumint(9)	Nombre del Permiso al que hace referencia	Calve Foránea

		usuario	
valor	tinyint(1)	Valor Perm_Rol	Requerido

Tabla 4. 21 Descripción de la tabla Permiso\_Usuario  
Elaborado por: El Investigador

<b>TABLA: recorrido_oficio</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>	<b>RESTRICCIONES</b>
Codigo	mediumint(9)	Identificación del rol	Clave Primaria
Dep_Origen	mediumint(9)	Departamento origen	Clave Foranea
Dep_Destino	mediumint(9)	Departamento destino	Clave foranea
Cod_Inst	mediumint(9)	Código de la instancia	Clave foranea
Cod_Ofi	mediumint(9)	Código del oficio	Clave foranea
Estado	tinyint(1)	Estado del oficio	Requerido
Aprobado	tinyint(1)	Aprobado	Requerido
Fecha	date	Fecha del oficio	Requerido

Tabla 4. 22 Descripción de la tabla Recorrido\_Oficio  
Elaborado por: El Investigador

<b>TABLA: usuarios</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>	<b>RESTRICCIONES</b>
Id	mediumint(9)	Identificación del usuario	Clave Primaria
Nombre	Varchar(100)	Nombre del usuario	Requerido
Usuario	Varchar(300)	Usuario de ingreso	Requerido
Pass	Varchar(40)	Contraseña del usuario	Requerido
Email	Varchar(100)	Correo electrónico del usuario	Requerido
Role	mediumint(9)	Rol del usuario	Clave foránea
Estado	tinyint (1)	Estado del usuario	Requerido
Fecha	date	Fecha del usuario	Requerido
Identificación	Varchar(10)	Identificación del usuario	Requerido
Cargo	Mediumint(9)	Cargo del usuario	Clave foránea

Tabla 4. 23 Descripción de la tabla Usuarios  
Elaborado por: El Investigador

#### **4.13. Implantación del sistema**

Para lograr la mayor funcionalidad de la aplicación se hace necesario alojarla en un servidor y compartirla en una red LAN, en dicho servidor debe estar instalado WAMP para permitir la mayor funcionalidad de la aplicación. De

esta forma las estaciones de trabajo conectadas a la red podrán acceder al sistema y hacer uso de todos sus módulos y funciones.

#### 4.14. Implementación de la programación

La arquitectura utilizada en la programación para el desarrollo de la aplicación web es Modelo Vista Controlador (MVC).

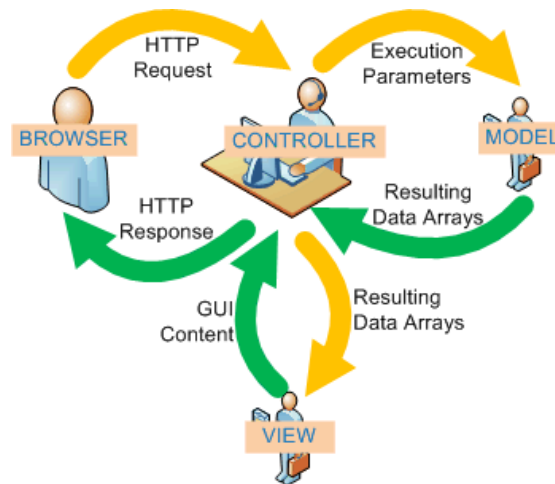


Imagen 4. 13 Diagrama de Modelo-Controlador-Vista

Fuente: <http://calidadyssoftware.blogspot.com/2011/10/aspnet-mvc-ii-ventajas.html>

##### 4.14.1. La capa del Modelo

El modelo representa la parte de la aplicación que implementa la lógica de negocio. Esto significa que es responsable de la recuperación de datos convirtiéndolos en conceptos significativos para la aplicación, así como su procesamiento, validación, asociación y cualquier otra tarea relativa a la manipulación de dichos datos.

A primera vista los objetos del modelo puede ser considerado como la primera capa de la interacción con cualquier base de datos que podría estar utilizando tu aplicación. Pero en general representan los principales conceptos en torno a los cuales se desea implementar un programa.

En el caso de una red social, la capa de modelo se haría cargo de tareas tales como guardar datos del usuario, el almacenamiento de asociaciones con amigos,

el almacenamiento y la recuperación de fotos de los usuarios, encontrar sugerencias de nuevos amigos, etc. Mientras que los objetos del modelo pueden ser considerados como “Amigo”, “Usuario”, “Comentario” y “Foto”.

#### **4.14.2. La capa de la Vista**

La vista hace una presentación de los datos del modelo estando separada de los objetos del modelo. Es responsable del uso de la información de la cual dispone para producir cualquier interfaz de presentación de cualquier petición que se presente.

Por ejemplo, como la capa de modelo devuelve un conjunto de datos, la vista los usaría para hacer una página HTML que los contenga. O un resultado con formato XML para que otras aplicaciones puedan consumir.

La capa de la Vista no se limita únicamente a HTML o texto que represente los datos, sino que puede ser utilizada para ofrecer una amplia variedad de formatos en función de sus necesidades tales como videos, música, documentos y cualquier otro formato que puedas imaginar.

#### **4.14.3. La capa del Controlador**

La capa del controlador gestiona las peticiones de los usuarios. Es responsable de responder la información solicitada con la ayuda tanto del modelo como de la vista.

Los controladores pueden ser vistos como administradores cuidando de que todos los recursos necesarios para completar una tarea se deleguen a los trabajadores más adecuados. Espera peticiones de los clientes, comprueba su validez de acuerdo a las normas de autenticación o autorización, delega la búsqueda de datos al modelo y selecciona el tipo de respuesta más adecuado

según las preferencias del cliente. Finalmente delega este proceso de presentación a la capa de la Vista. [16]

A continuación se mostrará algunos de los controladores, modelos y vistas más importantes dentro de la aplicación.

#### 4.15. Controlador de Ingreso a Departamento

```
<?php

class loginController extends Controller
{
    private $_login;

    public function __construct(){
        parent::__construct();
        $this->_login = $this->loadModel('login');
    }

    public function index()
    {
        if(Session::get('autenticado')){
            $this->redireccionar('index');
        }

        $this->_view->assign('titulo', 'Iniciar Sesion');

        if($this->getInt('enviar') == 1){
            $this->_view->assign('datos', $_POST);

            if(!$this->getSql('usuario')){
                $this->_view->assign('_error','Debe introducir su nombre de usuario');
                $this->_view->renderizar('index','login');
                exit;
            }

            if(!$this->getSql('pass')){
                $this->_view->assign('_error','Debe introducir su password');
                $this->_view->renderizar('index','login');
                exit;
            }

            $row = $this->_login->getUsuario(
                $this->getSql('usuario'),
                $this->getSql('pass')
            );

            if(!$row){
                $this->_view->assign('_error','Usuario y/o password incorrectos');
                $this->_view->renderizar('index','login');
                exit;
            }

            if($row['ESTADO'] != 1){
```

```

        $this->_view->assign('_error','Este usuario no esta habilitado');
        $this->_view->renderizar('index','login');
        exit;
    }

    Session::set('autenticado', true);
    Session::set('level', $row['role']);
    Session::set('usuario', $row['usuario']);
    Session::set('id_usuario', $row['id']);
    Session::set('tiempo', time());

    $this->redireccionar('index');
}

$this->_view->renderizar('index','login');

}

public function cerrar()
{
    Session::destroy();
    $this->redireccionar();
}
}
?>

```

#### 4.16. Controlador Departamento

```

<?php
class departamentoController extends Controller
{
    private $_departamento;
    public function __construct()
    {
        parent::__construct();
        $this->_departamento = $this->loadModel('departamento');
    }
    public function index($pagina = false)
    {
        $paginacion = new Paginador();
        $this->_view->setJs(array('index'));
        $this->_view->assign('departamentos', $paginacion->paginar($this->_departamento->getDepartamentos()));
        $this->_view->assign('paginacion', $paginacion->getView('paginacion_ajax'));
        $this->_view->assign('titulo', 'departamentos');
        $this->_view->renderizar('index', 'index');
    }
    public function indexAjax()
    {
        $pagina = $this->getInt('pagina');
        $nombre = $this->getSql('nombre');
        $registros = $this->getInt('registros');

        $paginacion = new Paginador();

        $this->_view->setJs(array('index'));
    }
}

```

```

        $this->_view->assign('departamentos', $paginacion->paginar($this-
        >_departamento->getDepartamentos($nombre), $pagina, $registros));
        $this->_view->assign('paginacion', $paginacion->getView('paginacion_ajax'));
        $this->_view->renderizar('ajax/index', false, true);
    }
    public function nuevo()
    {
        //$this->_acl->acceso('nuevo_post');
        $this->_view->assign('titulo', 'Nuevo Departamentos');
        //$this->_view->setJsPlugin(array('jquery.validate'));
        $this->_view->setJs(array('nuevo'));
        if($this->getInt('guardar') == 1){
            $this->_view->assign('datos', $_POST);
            if(!$this->getTexto('nombre')){
                $this->_view->assign('_error', 'Debe introducir el nombre del Cargo');
                $this->_view->renderizar('nuevo', 'departamento');
                exit;
            }
            $this->_departamento->insertarDepartamento(
                $this->getPostParam('nombre')
            );
            $this->redireccionar('departamento');
        }
        $this->_view->renderizar('nuevo', 'departamento');
    }
    public function editar($id)
    {
        //$this->_acl->acceso('editar_departamento');
        if(!$this->filtrarInt($id)){
            $this->redireccionar('departamento');
        }
        if(!$this->_departamento->getDepartamento($this->filtrarInt($id)){
            $this->redireccionar('departamento');
        }
        $this->_view->assign('titulo', 'Editar Departamentos');
        $this->_view->setJs(array('nuevo'));
        if($this->getInt('guardar') == 1){
            $this->_view->assign('datos', $_POST);
            if(!$this->getTexto('nombre')){
                $this->_view->assign('_error', 'Debe introducir el nombre del Estado Civil');
                $this->_view->renderizar('editar', 'departamento');
                exit;
            }
            if(!$this->getEntero('estaactivo')){
                $this->_view->assign('_error', 'Seleccione un Estado para el Estado Civil');
                $this->_view->renderizar('editar', 'departamento');
                exit;
            }
            $this->_departamento->editarDepartamento(
                $this->filtrarInt($id),
                $this->getPostParam('nombre'),
                $this->getPostParam('estaactivo')
            );
            $this->redireccionar('departamento');
        }
        $this->_view->assign('datos', $this->_departamento->getDepartamento($this-
        >filtrarInt($id));
        $this->_view->renderizar('editar', 'departamento');
    }

```



```

}
public function eliminar($id)
{
    //$this->_acl->acceso('eliminar_departamento');
    if(!$this->filtrarInt($id)){
        $this->redireccionar('departamento');
    }
    if(!$this->_departamento->getDepartamento($this->filtrarInt($id))){
        $this->redireccionar('departamento');
    }
    $this->_departamento->eliminarDepartamento($this->filtrarInt($id));
    $this->redireccionar('departamento');
}
}
?>

```

#### 4.17. Controlador Cargo

```

<?php
class cargoController extends Controller
{
    private $_estadocivil;
    public function __construct()
    {
        parent::__construct();
        $this->_estadocivil = $this->loadModel('cargo');
    }
    public function index($pagina = false)
    {
        $paginacion = new Paginador();

        $this->_view->setJs(array('index'));
        $this->_view->assign('cargos', $paginacion->paginar($this->_estadocivil->getCargos()));
        $this->_view->assign('paginacion', $paginacion->getView('paginacion_ajax'));
        $this->_view->assign('titulo', 'Cargos');
        $this->_view->renderizar('index', 'index');
    }
    public function indexAjax()
    {
        $pagina = $this->getInt('pagina');
        $nombre = $this->getSql('nombre');
        $registros = $this->getInt('registros');
        $paginacion = new Paginador();
        $this->_view->setJs(array('index'));
        $this->_view->assign('cargos', $paginacion->paginar($this->_estadocivil-
>getCargos($nombre), $pagina, $registros));
        $this->_view->assign('paginacion', $paginacion->getView('paginacion_ajax'));
        $this->_view->renderizar('ajax/index', false, true);
    }
    public function nuevo()
    {
        //$this->_acl->acceso('nuevo_post');
        $this->_view->assign('titulo', 'Nuevo Cargo');
        //$this->_view->setJsPlugin(array('jquery.validate'));
        $this->_view->setJs(array('nuevo'));
        if($this->getInt('guardar') == 1){

```

```

$this->_view->assign('datos', $_POST);
if(!$this->getTexto('nombre')){
    $this->_view->assign('_error', 'Debe introducir el nombre del Cargo');
    $this->_view->renderizar('nuevo', 'estadocivil');
    exit;
}
$this->_estadocivil->insertarCargo(
    $this->getPostParam('nombre')
);
$this->redireccionar('cargo');
}
$this->_view->renderizar('nuevo', 'cargo');
}
public function editar($id)
{
    //$this->_acl->acceso('editar_estadocivil');
    if(!$this->filtrarInt($id)){
        $this->redireccionar('cargo');
    }
    if(!$this->_estadocivil->getCargo($this->filtrarInt($id))){
        $this->redireccionar('cargo');
    }
    $this->_view->assign('titulo', 'Editar Cargo');
    $this->_view->setJs(array('nuevo'));
    if($this->getInt('guardar') == 1){
        $this->_view->assign('datos', $_POST);
        if(!$this->getTexto('nombre')){
            $this->_view->assign('_error', 'Debe introducir el nombre del Estado Civil');
            $this->_view->renderizar('editar', 'cargo');
            exit;
        }
        if(!$this->getEntero('estaactivo')){
            $this->_view->assign('_error', 'Seleccione un Estado para el Estado Civil');
            $this->_view->renderizar('editar', 'cargo');
            exit;
        }
        $this->_view->assign('_error', $this->getPostParam('estaactivo'));
        $this->_estadocivil->editarCargo(
            $this->filtrarInt($id),
            $this->getPostParam('nombre'),
            $this->getPostParam('estaactivo')
        );

        $this->redireccionar('cargo');
    }
    $this->_view->assign('datos', $this->_estadocivil->getCargo($this->filtrarInt($id)));
    $this->_view->renderizar('editar', 'cargo');
}
public function eliminar($id)
{
    //$this->_acl->acceso('eliminar_estadocivil');
    if(!$this->filtrarInt($id)){
        $this->redireccionar('cargo');
    }
    if(!$this->_estadocivil->getCargo($this->filtrarInt($id))){
        $this->redireccionar('cargo');
    }
    $this->_estadocivil->eliminarCargo($this->filtrarInt($id));
}

```

```

        $this->redireccionar('cargo');
    }
    public function prueba($pagina = false)
    {
        $paginador = new Paginador();
        $ajaxModel = $this->loadModel('ajax');
        $this->_view->setJs(array('prueba'));
        //$this->_view->assign('paises', $ajaxModel->getPaises());
        $this->_view->assign('estadociviles', $paginador->paginar($this->_estadocivil-
>getPrueba()));
        $this->_view->assign('paginacion', $paginador->getView('paginacion_ajax'));
        $this->_view->assign('titulo', 'estadociviles');
        $this->_view->renderizar('prueba', 'prueba');
    }
}
?>

```

#### 4.18. Controlador Oficio

```

<?php
class oficioController extends Controller
{
    private $_oficio;
    public function __construct()
    {
        parent::__construct();
        $this->_oficio = $this->loadModel('oficio');
    }
    public function index($pagina = false)
    {
        $paginacion = new Paginador();
        $this->_view->setJs(array('index'));
        $this->_view->assign('oficios', $paginacion->paginar($this->_oficio->getOficios()));
        $this->_view->assign('paginacion', $paginacion->getView('paginacion_ajax'));
        $this->_view->assign('titulo', 'oficios');
        $this->_view->renderizar('index', 'index');
    }
    public function indexAjax()
    {
        $pagina = $this->getInt('pagina');
        $nombre = $this->getSql('nombre');
        $registros = $this->getInt('registros');
        $paginacion = new Paginador();
        $this->_view->setJs(array('index'));
        $this->_view->assign('oficios', $paginacion->paginar($this->_oficio->getOficios($nombre),
$pagina, $registros));
        $this->_view->assign('paginacion', $paginacion->getView('paginacion_ajax'));
        $this->_view->renderizar('ajax/index', false, true);
    }
    public function nuevo()
    {
        //$this->_acl->acceso('nuevo_post');

        $this->_view->assign('titulo', 'Nuevo Oficios');
        //$this->_view->setJsPlugin(array('jquery.validate'));
        $this->_view->setJs(array('nuevo'));
    }
}

```

```

if($this->getInt('guardar') == 1){

    $this->_view->assign('datos', $_POST);
        try{
    if(!$this->getTexto('fecha')){
        $this->_view->assign('_error', 'Debe introducir el fecha del Oficio');
        $this->_view->renderizar('nuevo', 'oficio');
        exit;
    }

        if(!$this->getTexto('asunto')){
        $this->_view->assign('_error', 'Debe introducir el asunto del Oficio');
        $this->_view->renderizar('nuevo', 'oficio');
        exit;
    }

        if(!$this->getTexto('descripcion')){
        $this->_view->assign('_error', 'Debe introducir el descripcion del Oficio');
        $this->_view->renderizar('nuevo', 'oficio');
        exit;
    }

            $imagen = "";

if($_FILES['imagen']['name']){
    $ruta = ROOT . 'public' . DS . 'file' . DS ;
    $upload = new upload($_FILES['imagen'], 'es_Es');
    $upload->allowed = array('image/*');
    $upload->file_new_name_body = 'upl_' . uniqid();
    $upload->process($ruta);

    if($upload->processed){
        $imagen = $upload->file_dst_name;
        $thumb = new upload($upload->file_dst_pathname);
        $thumb->image_resize = true;
        $thumb->image_x = 100;
        $thumb->image_y = 70;
        $thumb->file_name_body_pre = 'thumb_';
        $thumb->process($ruta . 'thumb' . DS);
    }
    else{
        $this->_view->assign('_error', $upload->error);
        $this->_view->renderizar('nuevo', 'oficio');
        exit;
    }
}

    $this->_oficio->insertarOficio(
        $this->getPostParam('fecha'),
        $this->getPostParam('asunto'),
        $this->getPostParam('descripcion'),
        $imagen
    );

}

        catch(Exception $e){

```

```

        echo $e->getMessage();
    }

    $this->redireccionar('oficio');
}

$this->_view->renderizar('nuevo', 'oficio');
}

public function editar($id)
{
    //$this->_acl->acceso('editar_oficio');

    if(!$this->filtrarInt($id)){
        $this->redireccionar('departamento');
    }

    if(!$this->_oficio->getOficio($this->filtrarInt($id)){
        $this->redireccionar('departamento');
    }

    $this->_view->assign('titulo', 'Editar Departamentos');
    $this->_view->setJs(array('editar'));

    if($this->getInt('guardar') == 1){
        $this->_view->assign('datos', $_POST);

        //echo "<pre>"; var_dump($_POST);
        //exit;

        if(!$this->getTexto('fecha')){
            $this->_view->assign('_error', 'Debe introducir el fecha del Oficio');
            $this->_view->renderizar('editar', 'oficio');
            exit;
        }

        if(!$this->getTexto('asunto')){
            $this->_view->assign('_error', 'Debe introducir el asunto del Oficio');
            $this->_view->renderizar('editar', 'oficio');
            exit;
        }

        if(!$this->getTexto('descripcion')){
            $this->_view->assign('_error', 'Debe introducir el descripcion del Oficio');
            $this->_view->renderizar('editar', 'oficio');
            exit;
        }
    }
    $this->_oficio->editarOficio(
        $this->filtrarInt($id),
        $this->getPostParam('fecha'),
        $this->getPostParam('asunto'),
        $this->getPostParam('descripcion')
    );
    $this->redireccionar('oficio');
}
$this->_view->assign('datos', $this->_oficio->getOficio($this->filtrarInt($id)));
$this->_view->renderizar('editar', 'oficio');
}

```

```

public function eliminar($id)
{
    //$this->_acl->acceso('eliminar_oficio');

    if(!$this->filtrarInt($id)){
        $this->redireccionar('oficio');
    }

    if(!$this->_oficio->getOficio($this->filtrarInt($id))){
        $this->redireccionar('oficio');
    }

    $this->_oficio->eliminarOficio($this->filtrarInt($id));
    $this->redireccionar('oficio');
}
}
?>

```

#### 4.19. Controlador Recorrido

```

<?php

class recorridoController extends Controller
{
    private $_recorrido;

    public function __construct()
    {
        parent::__construct();
        $this->_recorrido = $this->loadModel('recorrido');
        $this->_oficio = $this->loadModel('oficio');
        $this->_departamento = $this->loadModel('departamento');
    }

    public function index($pagina = false)
    {
        $paginacion = new Paginador();

        $this->_view->setJs(array('index'));
        $this->_view->assign('recorridos', $paginacion->paginar($this->_recorrido->getRecorridos()));
        $this->_view->assign('paginacion', $paginacion->getView('paginacion_ajax'));
        $this->_view->assign('titulo', 'Recorridos');
        $this->_view->renderizar('index', 'index');
    }

    public function indexAjax()
    {
        $pagina = $this->getInt('pagina');
        $nombre = $this->getSql('nombre');
        $registros = $this->getInt('registros');

        $paginacion = new Paginador();
    }
}

```

```

    $this->_view->setJs(array('index'));
    $this->_view->assign('recorridos', $paginacion->paginar($this->_recorrido-
>getRecorridos($nombre), $pagina, $registros));
    $this->_view->assign('paginacion', $paginacion->getView('paginacion_ajax'));
    $this->_view->renderizar('ajax/index', false, true);
}

public function nuevo()
{
    //$this->_acl->acceso('nuevo_post');

    $this->_view->assign('titulo', 'Nuevo Recorrido');

    $this->_view->assign('oficios', $this->_oficio->getOficios());
    $this->_view->assign('departamentosorigen', $this->_departamento->getDepartamentos());
    $this->_view->assign('departamentosdestino', $this->_departamento->getDepartamentos());

    //$this->_view->setJsPlugin(array('jquery.validate'));
    $this->_view->setJs(array('nuevo'));

    if($this->getInt('guardar') == 1){

        $this->_view->assign('datos', $_POST);

        if(!$this->getInt('oficios')){
            $this->_view->assign('_error', 'Debe introducir el oficio del Oficio');
            $this->_view->renderizar('nuevo', 'recorrido');
            exit;
        }

        if(!$this->getInt('departamentosdestino')){
            $this->_view->assign('_error', 'Debe introducir el Departamento Origen del Oficio');
            $this->_view->renderizar('nuevo', 'recorrido');
            exit;
        }

        if(!$this->getInt('departamentosdestino')){
            $this->_view->assign('_error', 'Debe introducir el Departamento Destino del Oficio');
            $this->_view->renderizar('nuevo', 'recorrido');
            exit;
        }

        if(!$this->getInt(Session::get('autenticado'))){
            $this->_view->assign('_error', 'Debe Iniciar Session');
            $this->_view->renderizar('nuevo', 'recorrido');
            exit;
        }

        $this->_recorrido->insertarRecorrido(
            Session::get('autenticado'),
            $this->getPostParam('oficios'),
            $this->getPostParam('departamentosdestino'),
            $this->getPostParam('departamentosdestino')
        );

        $this->redireccionar('recorrido');
    }
}

```

```

$this->_view->renderizar('nuevo', 'recorrido');
}

public function editar($id)
{
    //$this->_acl->acceso('editar_recorrido');

    if(!$this->filtrarInt($id)){
        $this->redireccionar('recorrido');
    }

    if(!$this->_recorrido->getRecorrido($this->filtrarInt($id)){
        $this->redireccionar('recorrido');
    }

    $this->_view->assign('titulo', 'Editar Recorrido');

    $this->_view->setJs(array('editar'));

    if($this->getInt('guardar') == 1){
        $this->_view->assign('datos', $_POST);

        //echo "<pre>"; var_dump($_POST);
        //exit;

        $estado = 1;
        if(!$this->getTexto('estado')){
            $estado = 0;
        }

        $aprobado = 1;
        if(!$this->getTexto('estado')){
            $estado = 0;
        }

        if(!$this->getTexto('fecha')){
            $this->_view->assign('_error', 'Debe introducir el fecha del Oficio');
            $this->_view->renderizar('editar', 'recorrido');
            exit;
        }

        if(!$this->getTexto('asunto')){
            $this->_view->assign('_error', 'Debe introducir el asunto del Oficio');
            $this->_view->renderizar('editar', 'recorrido');
            exit;
        }

        if(!$this->getTexto('descripcion')){
            $this->_view->assign('_error', 'Debe introducir el descripcion del Oficio');
            $this->_view->renderizar('editar', 'recorrido');
            exit;
        }

        $this->_recorrido->editarRecorrido(
            $this->filtrarInt($id),

```



```

        $this->getPostParam('fecha'),
        $this->getPostParam('asunto'),
        $this->getPostParam('descripcion')
    );

    $this->redireccionar('recorrido');
}

$this->_view->assign('datos', $this->_recorrido->getRecorrido($this->filtrarInt($id));
$this->_view->assign('datos2', $this->_recorrido->getDetalleRecorridoUltimo($this->filtrarInt($id));
$this->_view->assign('detallerecorrido', $this->_recorrido->getDetalleRecorrido($id));
$this->_view->assign('departamentosorigen', $this->_departamento->getDepartamentos());
$this->_view->assign('departamentosdestino', $this->_departamento->getDepartamentos());
$this->_view->renderizar('editar', 'recorrido');
}
public function eliminar($id)
{
    //$this->_acl->acceso('eliminar_recorrido');

    if(!$this->filtrarInt($id)){
        $this->redireccionar('recorrido');
    }

    if(!$this->_recorrido->getRecorrido($this->filtrarInt($id)){
        $this->redireccionar('recorrido');
    }
    $this->_recorrido->eliminarRecorrido($this->filtrarInt($id));
    $this->redireccionar('recorrido');
}
}
?>

```

## 4.20. Modelo Login

```

<?php
class loginModel extends Model
{
    public function __construct() {
        parent::__construct();
    }
    public function getUsuario($usuario, $password)
    {
        $pass = Hash::getHash('sha1', $password, '4f6a6d832be79');
        $datos = $this->_db->query(
            "select * from usuarios " .
            "where USUARIO = '$usuario' " .
            "and PASS = " . $pass . ""
        );

        return $datos->fetch();
    }
}
?>

```

## 4.21. Modelo Departamento

```
<?php

class departamentoModel extends Model
{
    public function __construct() {
        parent::__construct();
    }

    public function getDepartamentos($condicion = "")
    {
        $departamento = $this->_db->query("select cod_dep, nombre, estado from
departamento WHERE nombre like '%$condicion%' order by nombre");
        return $departamento->fetchall();
    }

    public function getDepartamento($secuencial)
    {
        $secuencial = (int) $secuencial;
        $departamento = $this->_db->query("select cod_dep, nombre, estado from departamento
where cod_dep = $secuencial");
        return $departamento->fetch();
    }

    public function insertarDepartamento($nombre)
    {
        $this->_db->prepare("INSERT INTO departamento(cod_dep,nombre,estado) VALUES
(null, :nombre, 1)")
        ->execute(
            array(
                ':nombre' => $nombre
            ));
    }

    public function editarDepartamento($secuencial, $nombre, $estado)
    {
        $secuencial = (int) $secuencial;

        $this->_db->prepare("UPDATE departamento SET nombre = :nombre, estado = :estado
WHERE cod_dep = :secuencial")
        ->execute(
            array(
                ':secuencial' => $secuencial,
                ':nombre' => $nombre,
                ':estado' => $estado
            ));
    }

    public function eliminarDepartamento($secuencial)
    {
        $secuencial = (int) $secuencial;
        $this->_db->query("DELETE FROM departamento WHERE cod_dep = $secuencial");
    }
}
}
```

?>

## 4.22. Modelo Cargo

```
<?php

class cargoModel extends Model
{
    public function __construct() {
        parent::__construct();
    }

    public function getCargos($condicion = "")
    {
        $estadocivil = $this->_db->query("select cod_car, nombre, estado from cargo
WHERE nombre like '%$condicion%' order by nombre");
        return $estadocivil->fetchall();
    }

    public function getEstadosCivilesPorNombre($nombre)
    {
        $estadocivil = $this->_db->query("select cod_car, nombre, estado from cargo WHERE
nombre like '%".$nombre."' order by nombre");
        return $estadocivil->fetchall();
    }

    public function getCargo($secuencial)
    {
        $secuencial = (int) $secuencial;
        $estadocivil = $this->_db->query("select cod_car, nombre, estado from cargo where
cod_car = $secuencial");
        return $estadocivil->fetch();
    }

    public function insertarCargo($nombre)
    {
        $this->_db->prepare("INSERT INTO cargo(cod_car,nombre,estado) VALUES (null,
:nombre, 1)")
        ->execute(
            array(
                ':nombre' => $nombre
            ));
    }

    public function editarCargo($secuencial, $nombre, $estado)
    {
        $secuencial = (int) $secuencial;

        $this->_db->prepare("UPDATE cargo SET nombre = :nombre, estado = :estado WHERE
cod_car = :secuencial")
        ->execute(
            array(
                ':secuencial' => $secuencial,
                ':nombre' => $nombre,
                ':estado' => $estado
            ));
    }
}
```

```

        ));
    }
    public function eliminarCargo($secuencial)
    {
        $secuencial = (int) $secuencial;
        $this->_db->query("DELETE FROM cargo WHERE cod_car = $secuencial");
    }
    public function insertarPrueba($nombre)
    {
        $this->_db->prepare("INSERT INTO cargo VALUES (null, :nombre)")
        ->execute(
            array(
                ':nombre' => $nombre
            )
        );
    }
}
?>

```

### 4.23. Modelo Oficio

```

<?php

class oficioModel extends Model
{
    public function __construct() {
        parent::__construct();
    }

    public function getOficios($condicion = "")
    {
        $oficio = $this->_db->query("SELECT cod_ofi, fecha, asunto, descripcion,
archivo FROM oficio where asunto like '%".$condicion."' order by fecha");
        return $oficio->fetchall();
    }

    public function getOficio($secuencial)
    {
        $secuencial = (int) $secuencial;
        $oficio = $this->_db->query("SELECT cod_ofi, fecha, asunto, descripcion, archivo FROM
oficio where cod_ofi = $secuencial");
        return $oficio->fetch();
    }

    public function insertarOficio($fecha, $para, $de, $asunto, $descripcion, $archivo)
    {
        $this->_db->prepare("INSERT INTO oficio(cod_ofi, fecha, asunto,
descripcion, archivo) VALUES (null, ".$fecha.", ".$asunto.", ".$descripcion.", ".$archivo.")")
        ->execute();
    }

    public function editarOficio($secuencial, $fecha, $para, $de, $asunto, $descripcion)
    {
        $secuencial = (int) $secuencial;
        $this->_db->prepare("UPDATE oficio SET fecha = ".$fecha.", asunto = ".$asunto.",
descripcion = ".$descripcion." WHERE cod_ofi = ".$secuencial.")
        ->execute();
    }
}

```

```

    }

    public function eliminarOficio($secuencial)
    {
        $secuencial = (int) $secuencial;
        $this->_db->query("DELETE FROM oficio WHERE cod_ofi = $secuencial");
    }
}
?>

```

#### 4.24. Modelo Recorrido

```

<?php

class recorridoModel extends Model
{
    public function __construct() {
        parent::__construct();
    }

    public function getRecorridos($condicion = "")
    {
        $recorrido = $this->_db->query("SELECT i.cod_inst, i.responsable,
i.ejecucion, i.fecha, u.nombre as nombrepersonas, Concat('Asuntos: ',ifnull((select
GROUP_CONCAT(o.asunto) from oficio o
        inner join recorrido_oficio ro on o.cod_ofi = ro.cod_ofi where
ro.cod_inst = i.cod_inst),')) as nombreasunto
        FROM instancia i
        inner join usuarios u on u.id = i.responsable

        where u.nombre like '%".$condicion."' order by i.fecha");
        return $recorrido->fetchall();
    }

    public function getRecorrido($secuencial)
    {
        $secuencial = (int) $secuencial;
        $recorrido = $this->_db->query("SELECT i.cod_inst, i.responsable, i.ejecucion, i.fecha,
u.nombre as nombrepersonas, Concat('Asuntos: ',ifnull((select GROUP_CONCAT(o.asunto)
        from oficio o
        inner join recorrido_oficio ro on o.cod_ofi = ro.cod_ofi where
ro.cod_inst = i.cod_inst),')) as nombreasunto
        FROM instancia i
        inner join usuarios u on u.id = i.responsable where i.cod_inst =
$secuencial");
        return $recorrido->fetch();
    }

    public function getDetalleRecorrido($secuencial)
    {
        $secuencial = (int) $secuencial;
        $recorrido = $this->_db->query("SELECT ro.codigo, ro.dep_origen, ro.dep_destino,
ro.cod_inst, ro.cod_ofi, ro.estado, ro.aprobado, ro.fecha,
        d1.nombre as nombreDepOrigen, d2.nombre as nombreDepDestino,
o.asunto, o.archivo FROM recorrido_oficio ro
        inner join departamento d1 on d1.cod_dep = ro.dep_origen

```

```

        inner join departamento d2 on d2.cod_dep = ro.dep_destino
        inner join oficina o on o.cod_ofi = ro.cod_ofi where ro.cod_inst =
$secuencial and ro.aprobado = 1");
        return $recorrido->fetchall();
    }

    public function getDetalleRecorridoUltimo($secuencial)
    {
        $secuencial = (int) $secuencial;
        $recorrido = $this->_db->query("SELECT ro.codigo, ro.dep_origen, ro.dep_destino,
ro.cod_inst, ro.cod_ofi, ro.estado, ro.aprobado, ro.fecha,
        d1.nombre as nombreDepOrigen, d2.nombre as nombreDepDestino,
o.asunto, o.archivo
        FROM recorrido_oficio ro
        inner join departamento d1 on d1.cod_dep = ro.dep_origen
        inner join departamento d2 on d2.cod_dep = ro.dep_destino
        inner join oficina o on o.cod_ofi = ro.cod_ofi where ro.cod_inst =
$secuencial and ro.aprobado <> 1 and ro.fecha = (select max(ro2.fecha) FROM recorrido_oficio
ro2 where ro2.cod_inst = $secuencial and ro2.aprobado <> 1)");
        return $recorrido->fetch();
    }
    public function insertarRecorrido($responsable, $oficio, $depOrigen, $depDestino)
    {
        $this->_db->prepare("INSERT INTO instancia(cod_inst, responsable,
ejecucion, fecha)
        VALUES (null, ".$responsable.", 1, CURDATE())")
        ->execute();

        $this->_db->prepare("INSERT INTO recorrido_oficio(codigo, dep_origen, dep_destino,
cod_inst, cod_ofi, estado, aprobado, fecha)
        VALUES (null, ".$depOrigen.", ".$depDestino.", ".$codig_inst.",
".$oficio.", 1, 0, CURDATE())")
        ->execute();
    }
    public function editarRecorrido($secuencial, $fecha, $para, $de, $asunto, $descripcion)
    {
        $secuencial = (int) $secuencial;
        $this->_db->prepare("UPDATE oficina SET fecha = ".$fecha.", asunto = ".$asunto.",
descripcion = ".$descripcion." WHERE cod_ofi = ".$secuencial."")
        ->execute();
    }
    public function eliminarRecorrido($secuencial)
    {
        $secuencial = (int) $secuencial;
        $this->_db->query("DELETE FROM oficina WHERE cod_ofi = $secuencial");
    }
}
?>

```

## 4.25. Vistas Departamento

```
<center><h2>Departamentos</h2></center>
```

```
<div class="well well-small">
```

```
<form id="form1" class="form-inline">
```

```
Nombre: <input type="text" name="nombre" id="nombre">
```

```

        <button type="button" id="btnBucar" class="btn" ><i class="icon-
search"></i>Buscar</button>

    </form>
</div>
<br>

<div id="lista_registros">
    {if isset($departamentos) && count($departamentos)}

        <table class="table table-bordered table-condensed table-striped">
            <tr>
                <th>Codigo</th>
                <th>Nombre</th>
                <th>Estado</th>
                <th></th>
            </tr>

            {foreach item=departamento from=$departamentos}
                <tr>
                    <td style="text-align: center;">{$departamento.cod_dep}</td>
                    <td>{$departamento.nombre}</td>
                    <td>{$departamento.estado}</td>
                    <td style="text-align: center; vertical-align: middle;"><a
href="/departamento/editar/{ $departamento.cod_dep }">Editar</a></td>
                    <td style="text-align: center; vertical-align:
middle;"><a href="/departamento/eliminar/{ $departamento.cod_dep }">Eliminar</a></td>
                </tr>

            {/foreach}
        </table>

    {else}

        <p><strong>No hay Departamentos!</strong></p>

    {/if}

    <br>
    <a href="{ $ _layoutParams.root }departamento/nuevo" class="btn btn-primary"><i
class="icon-plus-sign icon-white"> </i> Agregar Departamentos</a>

    { $paginacion|default:""}
</div>

```

## 4.26. Vista Oficio

```

<center><h2>Oficios</h2></center>

<div class="well well-small">
    <form id="form1" class="form-inline">
        Asunto: <input type="text" name="nombre" id="nombre">
        <button type="button" id="btnBucar" class="btn" ><i class="icon-
search"></i>Buscar</button>

```

```

    </form>
</div>
<br>

<div id="lista_registros">
    {if isset($oficios) && count($oficios)}

        <table class="table table-bordered table-condensed table-striped">
            <tr>
                <th>Fecha</th>
                <th>Asunto</th>
                <th>Codigo</th>
                <th>Descripcion</th>
                <th>Archi vo</th>
                <th></th>
                <th></th>
            </tr>

            {foreach item=oficio from=$oficios}
                <tr>
                    <td style="text-align: center;">{$oficio.cod_ofi}</td>
                    <td>{$oficio.fecha}</td>
                    <td>{$oficio.asunto}</td>
                    <td>{$oficio.descripcion}</td>
                    <td>{$oficio.archivo}</td>
                    <td style="text-align: center; vertical-align: middle;"><a
href="/oficio/editar/{ $oficio.cod_ofi }">Editar</a></td>
                    <td style="text-align: center; vertical-align:
middle;"><a href="/oficio/eliminar/{ $oficio.cod_ofi }">Eliminar</a></td>
                </tr>

            {/foreach}
        </table>

    {else}

        <p><strong>No hay Oficios!</strong></p>

    {/if}

    <br>
    <a href="/oficio/nuevo" class="btn btn-primary"><i class="icon-plus-sign icon-white">
</i> Agregar Oficios</a>

    { $paginacion|default:"" }
</div>

```

## 4.27. Vista Recorrido

```

<center><h2>Recorridos</h2></center>

<div class="well well-small">
    <form id="form1" class="form-inline">
        Asunto: <input type="text" name="nombre" id="nombre">
    </form>
</div>

```



```

        <button type="button" id="btnBucar" class="btn" ><i class="icon-search"></i>Buscar</button>

    </form>
</div>
<br>

<div id="lista_registros">
    {if isset($recorridos) && count($recorridos)}

    <table class="table table-bordered table-condensed table-striped">
        <tr>
            <th>ID</th>
            <th>Fecha</th>
            <th>Responsable</th>
            <th>Descripcion</th>
            <th>En Ejecucion</th>
            <th></th>
            <th></th>
        </tr>

        {foreach item=recorrido from=$recorridos}
        <tr>
            <td style="text-align: center;">{$recorrido.cod_inst}</td>
            <td>{$recorrido.fecha}</td>
            <td>{$recorrido.nombrepersonas}</td>
            <td>{$recorrido.nombreasunto}</td>
            <td>{$recorrido.ejecucion}</td>
            <td style="text-align: center; vertical-align: middle;"><a href="/recorrido/editar/{$recorrido.cod_inst}">Editar</a></td>
            <td style="text-align: center; vertical-align: middle;"><a href="/recorrido/eliminar/{$recorrido.cod_inst}">Eliminar</a></td>
        </tr>

        {/foreach}
    </table>

    {else}

    <p><strong>No hay Recorridos!</strong></p>

    {/if}

    <br>
    <a href="/recorrido/nuevo" class="btn btn-primary"><i class="icon-plus-sign icon-white"></i> Agregar Recorridos</a>

    { $paginacion|default:"" }
</div>

```

#### 4.28. Manual de usuario

Un manual de usuario es una guía en la cual se detallan las funciones que tiene la aplicación web, para un correcto uso de los usuarios finales del sistema, también se explica el ingreso de datos para poder alimentar de una manera funcional la aplicación web. Durante la elaboración del Manual de Usuario se debe tomar en cuenta al personal que va dirigido para que este sea lo más entendible y práctico.

La aplicación consta de un menú principal compuesto por 4 módulos principales con las siguientes funciones:

**Inicio:** Permite al usuario acceder a la pantalla principal del sistema, donde puede entre otras opciones autenticarse con su nombre de usuario y contraseña.

**Registros:** Contiene toda la información registrada en el sistema, la cual se distribuye en 4 módulos.

**Documentos:** Contiene documentos de interés para el trabajo de la delegación, el administrador puede incluir nueva documentación en dependencia de las necesidades de los usuarios del sistema.

**Recorridos:** Contiene un mapa del sitio y muestra la funcionabilidad de los diferentes módulos que componen el sistema.

Dentro del módulo Registros

Submenú	Función
<b>Cargos</b>	Posee la plantilla de cargos de todos los miembros del centro que estén registrados en el sistema
<b>Departamentos</b>	Relación nominal de todos los departamentos que integran la delegación.
<b>Usuarios</b>	Listado de usuarios que contiene el sistema. EN caso de acceder como administrador le permitirá agregar,

	eliminar o modificar los datos de los usuarios que contiene el sistema.
<b>Roles</b>	Determina los roles y permisos que tiene cada usuario al acceder al sistema.

Tabla 4. 24. Submenú del módulo registros  
Elaborado por: El Investigador

- **Árbol de Navegación**

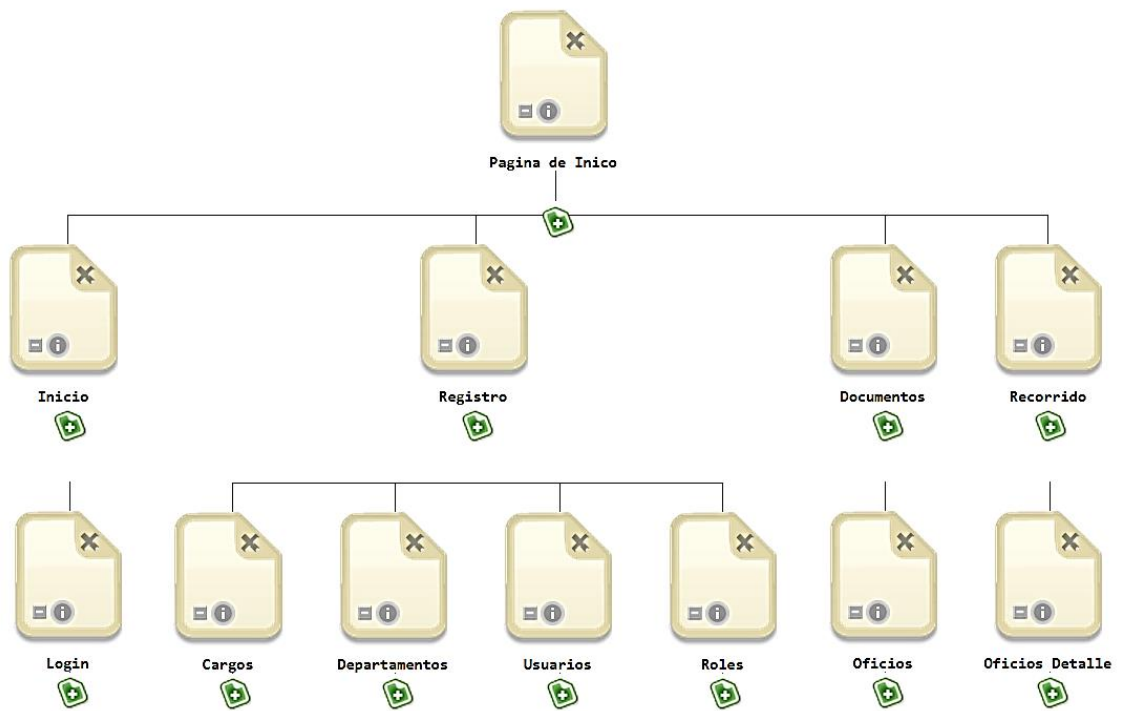


Imagen 4. 14 Árbol de Navegación  
Elaborado por: El Investigador

### **Pantalla de Inicio.**

En esta página se muestra el nombre de la institución, nombre de la aplicación y además es el primer menú de la aplicación que lleva hasta la página de Autenticación.



Imagen 4. 15 Interfaz de la Página de Inicio  
Elaborado por: El Investigador

1.-Logo de la Institución.

2.-Información de la Institución.

3.-Menú de Inicio de Sesión.

### **Pantalla de Inicio de Sesión.**

En esta página se muestra la información que debe ingresar el usuario para autenticarse como es el Usuario y la Contraseña los mismos que deben ser ingresados de manera correcta para poder seguir utilizando la aplicación web.

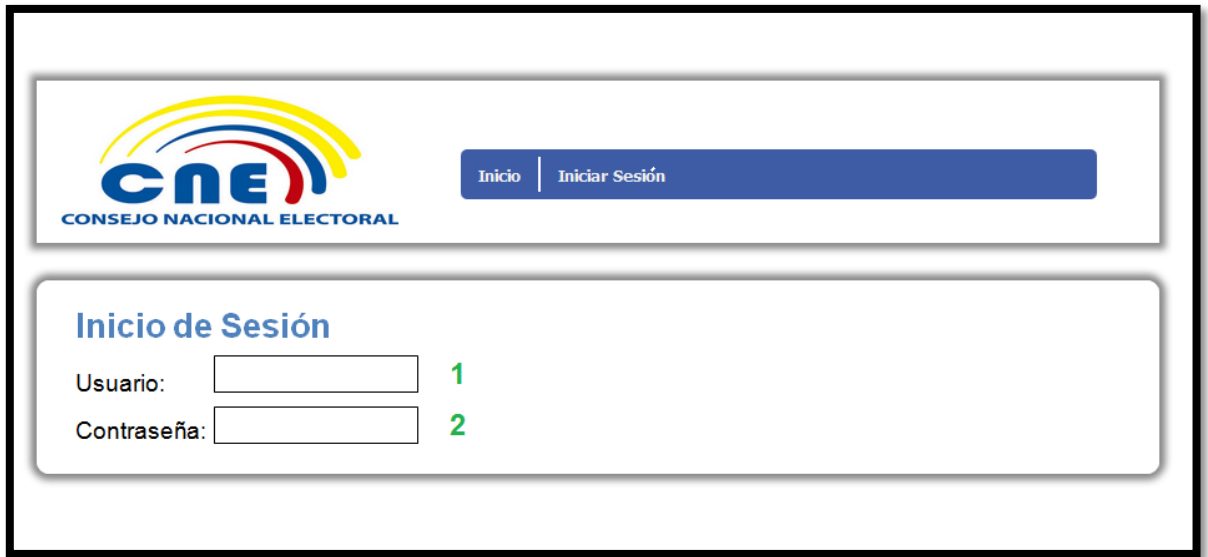


Imagen 4. 16 Interfaz de la Pantalla de inicio de sesión  
Elaborado por: El Investigador

1.-Nombre del usuario.

2.-Contraseña asignada al usuario.

### **Menú Principal.**

En esta imagen se muestra el menú principal el mismo que consta de 5 opciones detalladas a continuación.

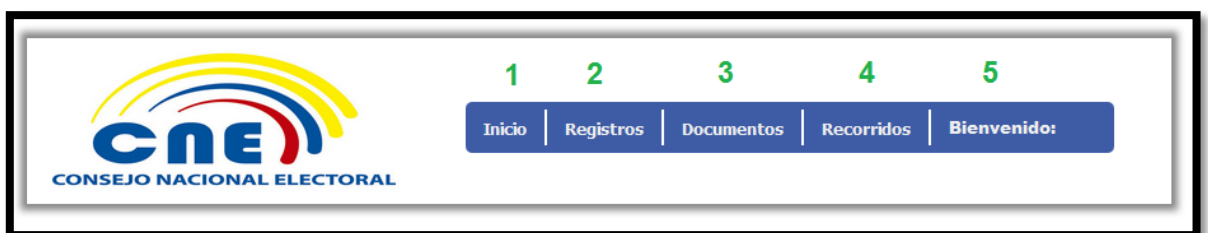


Imagen 4. 17 Interfaz de la Página Menú principal  
Elaborado por: El Investigador

1.-Muestra la pantalla de Inicio

2.-Dentro de Registros del menú principal se encuentra un submenú que consta de:

- ✓ Cargos
- ✓ Departamentos
- ✓ Usuarios
- ✓ Roles

3.-Documentos muestra los oficios que están dentro del sistema.

4.-Recorridos muestra el ciclo que ha pasado un oficio.

5.-Bienvenido desde esta opción se puede finalizar la sesión con la cual está trabajando el usuario actual.

### Pantalla de Departamentos.

En esta pantalla se observa la interfaz para agregar información acerca de los departamentos en la aplicación web.



Imagen 4. 18 Interfaz de la Página Departamentos  
Elaborado por: El Investigador

1.- En esta opción se puede realizar una búsqueda inteligente para agilizar el proceso de encontrar un departamento en específico.

2.-En este espacio de trabajo se muestra de manera ordenada y en forma de lista de departamentos ingresados correctamente.

3.-Si se da un clic en la opción Agregar Departamentos se obtiene la pantalla de ingreso de datos necesarios para crear un Departamento.

4.-En esta opción se muestra diferente botones con los cuales se navega por las diferentes páginas del listado de Departamentos.

### **Pantalla de Agregar Departamentos.**

En esta pantalla se observa la interfaz para agregar departamentos en la aplicación web.



The screenshot displays the user interface for adding a new department. At the top left is the logo for the Consejo Nacional Electoral (CNE). To the right is a blue navigation bar with links for 'Inicio', 'Registros', 'Documentos', 'Recorridos', 'Reporte', and 'Bienvenido:'. The main content area is titled 'Nuevo Departamentos' and features a text input field containing 'Recursos Humanos' with a green '1' indicating the input field. Below the input field is a 'Guardar' button with a green '2' indicating the save button.

Imagen 4. 19 Interfaz de la página de Agregar Departamento  
Elaborado por: El Investigador

1.- En esta opción se ingresa el nombre del nuevo departamento para su posterior utilización.

2.-Con este botón se guarda la información ingresada.

### **Pantalla de Editar Departamento.**

En esta pantalla podemos observar la interfaz para editar la información de los Departamentos en la aplicación web.



Imagen 4. 20 Editar departamento  
Elaborado por: El Investigador

- 1.- En esta opción se modifica el nombre del departamento en caso de haber tenido un error o de ser el caso se cambia de nombre al departamento.
- 2.-Esta pestaña permite dar el estado al departamento pudiendo ser “Activo” o “Inactivo”.
- 3.- Al dar un clic se guarda los cambios realizados.

### **Pantalla de Cargos.**

En esta pantalla se observa la interfaz para agregar información acerca de cargos en la aplicación web.





Imagen 4. 21 Interfaz de la Pantalla de Cargo  
Elaborado por: El Investigador

- 1.- En esta opción se puede realizar una búsqueda inteligente para agilizar el proceso de encontrar un cargo en específico.
- 2.-En este espacio de trabajo se muestra de manera ordenada y en forma de lista los cargos ingresados correctamente.
- 3.-Si se da un clic en la opción Agregar Cargos se obtiene la pantalla donde se ingresa los datos necesarios para crear un cargo.
- 4.-En esta opción se mostraran diferente botones con los cuales se navega por las diferentes páginas del listado de cargos.

### **Pantalla de Editar Cargo.**

En esta pantalla podemos observar la interfaz para editar la información de los Cargos en la aplicación web.



Imagen 4. 22 Página Editar Cargo  
Elaborado por: El Investigador

- 1.- En esta opción se modifica el nombre del cargo en caso de haber tenido un error o de ser el caso se cambia de nombre al mismo.
- 2.-Esta pestaña permite dar el estado al cargo pudiendo ser este “Activo” o “Inactivo”.
- 3.- Al dar un clic se guarda los cambios realizados.

### **Pantalla de Roles.**

En esta pantalla se observa la interfaz para agregar información acerca de los Roles en la aplicación web.



Imagen 4. 23 Página Roles  
Elaborado por: El Investigador

- 1.- En este espacio se identifica el nombre de la página en la que se está trabajando.
- 2.- En esta opción se realiza una búsqueda inteligente para agilizar el proceso de encontrar un rol en específico.
- 3.-En este espacio de trabajo se muestra de manera ordenada y en forma de lista de roles ingresados correctamente.
- 4.-Dando un clic en la opción Agregar Roles se obtiene la pantalla donde se ingresa los datos necesarios para crear un Rol.
- 5.-En esta opción se mostraran diferentes botones con los cuales se navega por las diferentes páginas del listado de roles.
- 6.- Esta opción permite al usuario modificar el número de roles que se puede visualizar en la pantalla.

### **Pantalla de Editar Rol.**

En esta pantalla podemos observar la interfaz para editar la información de los Cargos en la aplicación web.

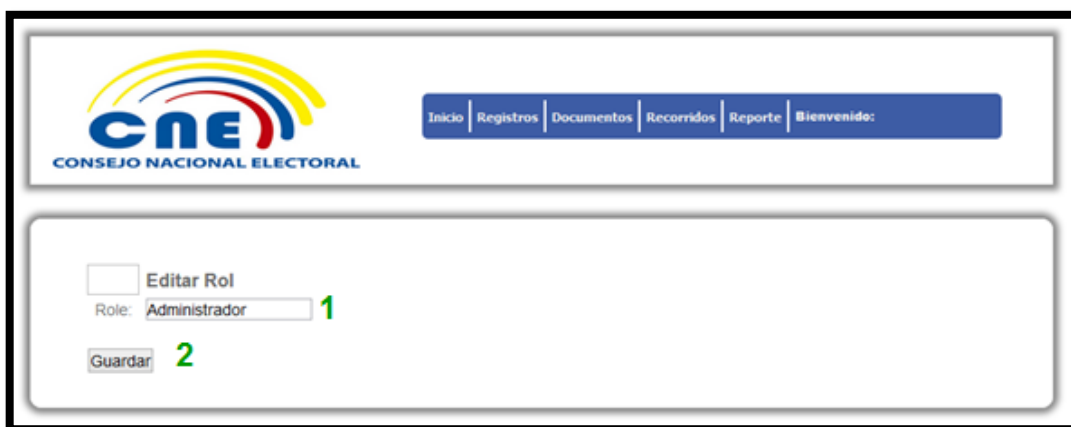


Imagen 4. 24 Página Editar Rol  
Elaborado por: El Investigador

- 1.- En esta opción se modifica el nombre del rol en caso de haber tenido un error o de ser el caso se cambia de nombre al mismo.
- 2.- Al dar un clic se guarda los cambios realizados.

### **Pantalla de Usuarios.**

En esta pantalla se observa la interfaz para agregar información acerca de los Usuarios en la aplicación web.



Imagen 4. 25 Página de usuarios  
Elaborado por: El Investigador

- 1.- En este espacio se realiza una búsqueda inteligente para agilizar el proceso de encontrar un usuario en específico.
- 2.-En este espacio de trabajo se muestra de manera ordenada y en forma de lista de usuarios ingresados correctamente.
- 3.-Al dar clic en la opción Agregar Usuarios se visualiza la pantalla de ingreso de datos necesarios para crear un Usuario.
- 4.-En esta opción se mostraran diferente botones con los cuales se navega por las diferentes páginas del listado de Usuarios.

### **Pantalla Creación de Usuario.**

En esta pantalla se observa los campos necesarios que deben ser llenados con información válida para poder crear un usuario.

**CNE**  
CONSEJO NACIONAL ELECTORAL

Inicio | Registros | Documentos | Recorridos | Bienvenido:

**Nuevo Usuario**

Usuario:  1

Contraseña:  Repetir Contraseña:

Identificación:  Nombre:

E-mail:

Cargos:  2

Roles:  2

Estado: Estado  3

Guardar 4

Imagen 4. 26 Interfaz de la Pantalla creación de usuario  
Elaborado por: El Investigador

- 1.-Campos obligatorios que deben ser llenados con información personal del Usuario.
- 2.-ComboBox que permite elegir un cargo y un rol para el usuario que está en proceso de creación.
- 3.- Checklist permite marcar como activo al usuario que se acaba de crear.
- 4.- Botón Guardar se aplica una vez que se haya llenado todos los campos antes mencionados para almacenar el nuevo usuario en la base de datos de la aplicación web.

### **Pantalla Editar Usuario.**

En esta pantalla se observa los campos necesarios que deben ser llenados con información válida para poder crear un usuario.

**Editar Usuarios**

Usuario:

Contraseña:  Repetir Contraseña:  **1**

Identificación:

Nombre:  Apellido:

E-mail:

Cargos:

Roles:

Estado:

**2**

Imagen 4. 27 Página editar usuario  
Elaborado por: El Investigador

1.-Campos para modificar la información de los usuarios, los mismos que ya fueron detallados en el ingreso de usuario.

2.- Botón Guardar para almacenar las correcciones del usuario en la base de datos de la aplicación web.

### **Pantalla Oficios.**

En esta pantalla se observa los oficios ingresados con la información más relevante de los mismos.

Logo: **CNE** CONSEJO NACIONAL ELECTORAL

Inicio | Registros | Documentos | Recorridos | Reporte | Bienvenido:

Asunto:  **2** **Oficios** **1**

Codigo	Fecha	Asunto	Descripcion	Archivo	<b>3</b>	
2	2014-10-03	pruebas 22	pruebas de oficio 2	sadfdsfasdf	Editar	Eliminar
1	2014-10-04	pruebas	preubas de oficios	sdfas	Editar	Eliminar
4	2015-01-02	upl_54a8310e315b6.jpg			Editar	Eliminar
7	2015-01-05	Pago	Pago Vehiculo	upl_imagen.jpg54a837bbdc4b5.jpg	Editar	Eliminar
8	2015-01-05	Pago	Camioneta	upl_imagen.jpg.jpg	Editar	Eliminar
10	2015-01-05	Pedido de Votacion	Certifique donde Sufrago el Ciudadano	upl_imagen_mod.jpg_1.jpg	Editar	Eliminar
11	2015-01-05	Derechos Politicos	Perdida de Derechos	upl_imagen.jpg_1.jpg	Editar	Eliminar
12	2015-01-05	Cedu Certificado	Cedula Y certificado	upl_el.jpg.jpg	Editar	Eliminar

**Agregar Oficios** **4**

<< < 1 > >> **5**

10 **6**

Página 1 de 1  
Registros por pagina:

Imagen 4. 28 Página de Oficios  
Elaborado por: El Investigador

- 1.- En este espacio se identifica el nombre de la página en la que se está trabajando.
- 2.- En esta opción se realiza una búsqueda inteligente para agilizar el proceso de encontrar un oficio en específico.
- 3.-En este espacio de trabajo se muestra de manera ordenada y en forma de lista de oficios ingresados correctamente.
- 4.-Si se da un clic en la opción Agregar Oficios se visualiza la pantalla donde se ingresa los datos necesarios para crear un Oficio.
- 5.-En esta opción se mostraran diferente botones con los cuales se navega por las diferentes páginas del listado de Oficios.



6.- Esta opción permite modificar el número de oficios que se puede ver en la pantalla.

### **Pantalla Nuevo Oficio.**

En esta pantalla se observa los campos necesarios que deben ser llenados con información válida para crear un nuevo oficio.

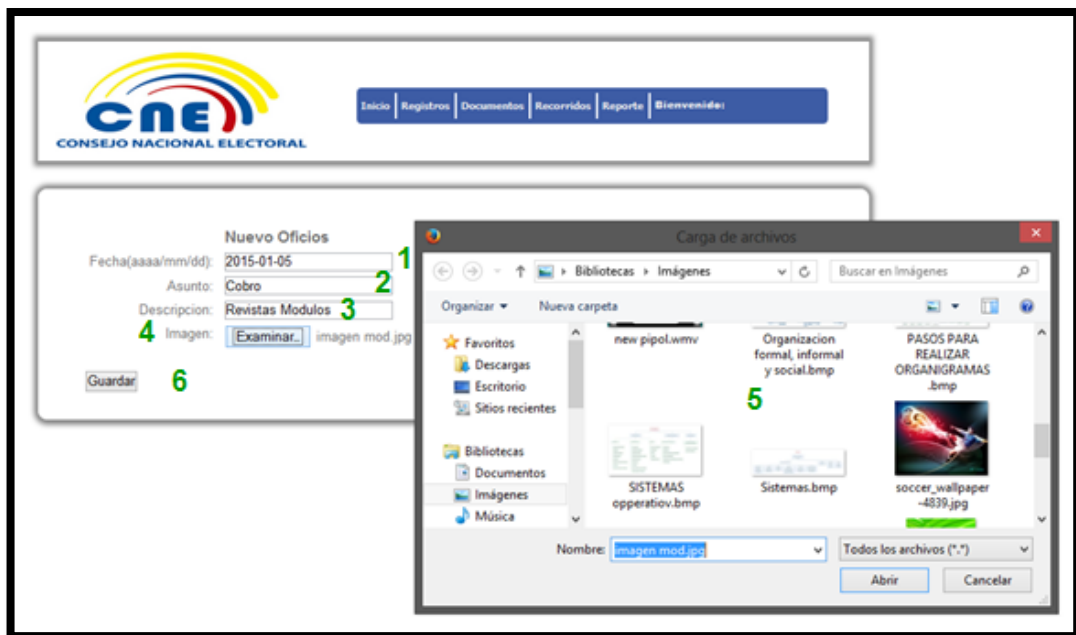


Imagen 4. 29 Pagina Nuevo Oficio  
Elaborado por: El Investigador

- 1.- Espacio para ingresar la fecha del oficio.
- 2.- Todo oficio tiene un asunto por lo tanto el asunto del oficio tiene que ser ingresado en este espacio.
- 3.- Redactar una pequeña descripción que le permita identificar el oficio.
- 4.- Botón que permite acceder a una ventana auxiliar para cargar el oficio que se encontrara ya en forma digital.
- 5.- Ventana auxiliar para navegar por las ubicaciones del ordenador con la finalidad de encontrar el archivo.

6.- Botón que permite guardar la información ingresada en la pantalla.

### **Pantalla Editar Oficio.**

En esta pantalla se puede modificar los datos ingresados en caso de existir algún error, pero se debe tener en cuenta que no se podrá cargar otro archivo, únicamente se puede realizar cambios en los datos.



Imagen 4. 30 Página Editar Oficio  
Elaborado por: El Investigador

1.- Información que puede ser editada del oficio detallado en nuevo oficio.

2.- Botón que sirve para guardar los cambios realizados en la base de datos de la Aplicación.

### **Pantalla Recorrido.**

En esta pantalla se crea los recorridos de los oficios creados anteriormente.

**CNE**  
CONSEJO NACIONAL ELECTORAL

Inicio | Registros | Documentos | Recorridos | Reporte | Bienvenido

Asunto:  **Buscar 2**

### Recorridos 1

Codigo	Fecha	Asunto	Descripcion	Archivo		
2	2014-10-03	pruebas 22	pruebas de oficio 2	sadfósfasdf	Editar	Eliminar
1	2014-10-04	pruebas	preubas de oficios	sdfas	Editar	Eliminar
4	2015-01-02	upl_54a8310e315b6.jpg	<b>3</b>		Editar	Eliminar
10	2015-01-05	Pedido de Votacion	Certifique donde Sufrago el Ciudadano	upl_imagen_mod.jpg_1.jpg	Editar	Eliminar
11	2015-01-05	Derechos Politicos	Perdida de Derechos	upl_imagen.jpg_1.jpg	Editar	Eliminar
12	2015-01-05	Cedu Certificado	Cedula Y certificado	upl_el.jpg.jpg	Editar	Eliminar
13	2015-01-05	Cobro	Revistas Modulos	upl_imagen_mod.jpg_2.jpg	Editar	Eliminar

Agregar Oficios **4**

< < 1 > >

Pagina 1 de 1

10 **6** Registros por pagina:

Imagen 4. 31 Página Recorrido  
Elaborado por: El Investigador

- 1.- En este espacio se identifica el nombre de la página en la que se está trabajando.
- 2.- En esta opción se realiza una búsqueda inteligente para agilizar el proceso de encontrar un oficio en específico con el correspondiente recorrido.
- 3.-En este espacio de trabajo se muestra de manera ordenada y en forma de lista de oficios que están interviniendo en un proceso de recorrido.
- 4.-Si damos un clic en la opción Agregar Oficios vamos a obtener la pantalla para seleccionar el oficio y empezar el recorrido del mismo.
- 5.-En esta opción se mostraran diferente botones con los cuales podemos navegar por las diferentes páginas del listado de Recorridos.
- 6.- Esta opción permite modificar el número de recorridos que se puede ver en la pantalla.

### **Pantalla Nuevo Recorrido.**

En esta pantalla se inicia el envío del oficio de un departamento solicitante a otro departamento el cual tendrá que validar y así autorizar el paso del mismo.

Imagen 4. 32 Página Nuevo Recorrido  
Elaborado por: El Investigador

- 1.- Pestaña para seleccionar un oficio ingresado previamente, se lo identificara ya que en el nombre se encuentra la fecha, asunto y descripción ingresada en el orden mencionado.
- 2.- Pestaña para seleccionar el departamento del cual se envía el oficio.
- 3.- Pestaña para seleccionar el departamento al que va dirigido el oficio.
- 4.- Botón para guardar el recorrido realizado.

### **Pantalla Editar Recorrido.**

Esta pantalla se utiliza para que reenviar un oficio a otro departamento pudiendo ser el de origen u otro diferente.

'. Below it, it says 'No hay Recorridos Aprobados!'. The 'Nuevo Recorrido' section shows 'Departamento Origen: Juridico' and 'Departamento Destino: Sistemas' with a green '2' next to it. Below that, 'Oficio: Asuntos: Cedu Certificado' and 'Aprobado: 

Imagen 4. 33 Página Editar recorrido  
Elaborado por: El Investigador

- 1.- En este espacio se visualiza el responsable de la solicitud y un checklist para finalizar el proceso.
- 2.- En este espacio se encuentra especificado el departamento del origen, el de destino y el nombre del oficio.
- 3.- Checklist para aprobar la solicitud.
- 4.- Departamento de siguiente destino, es decir en la pestaña se selecciona el departamento puede ser que regrese al origen o puede enviarse a otro departamento para continuar con el proceso de la solicitud.
- 5.- Botón para guardar los cambios efectuados

## Pantalla Reporte

Esta pantalla permite seleccionar un oficio para conocer la situación actual del oficio.



Imagen 4. 34 Página Reporte  
Elaborado por: El Investigador

1. Espacio para seleccionar el oficio del cual se desea general el reporte.
2. Botón para generar el reporte.

## Reporte generado finalmente.



#	Departamento Origen	Departamento Destino	Estado
1	Juridico	Sistemas	Finalizado
2	Sistemas	Juridico	Finalizado
3	Juridico	Recursos Humanos	Finalizado

Imagen 4. 35 Reporte Generado  
Elaborado por: El Investigador

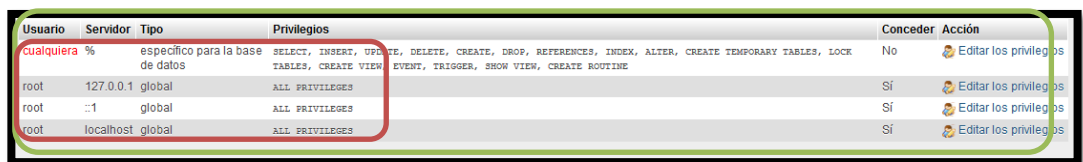
#### 4.29. Pruebas de Funcionamiento

Después de realizar la implementación de la Aplicación Web se procede a realizar las pruebas de funcionamiento teniendo las pruebas de caja negra y caja blanca, con ello se detecta los posibles errores que se han cometido.

- **Pruebas de Caja Negra**

Estas pruebas se basan únicamente en lo que se presenta en la salida y en la entrada sin importar lo que esté sucediendo internamente en la Aplicación es por esta razón que se realizan las dos pruebas para poder comprobar el funcionamiento interno y externo de la misma.

Con estas pruebas se realiza la comprobación de que cada página y dato se ajuste correctamente a las necesidades de los usuarios y los requerimientos planteados inicialmente, así como la seguridad que deben tener cada uno de los datos según lo estipule el usuario.



Usuario	Servidor	Tipo	Privilegios	Conceder	Acción
cualquiera	%	especifico para la base de datos	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES, CREATE VIEW, EVENT, TRIGGER, SHOW VIEW, CREATE ROUTINE	No	Editar los privilegios
root	127.0.0.1	global	ALL PRIVILEGES	Sí	Editar los privilegios
root	:::1	global	ALL PRIVILEGES	Sí	Editar los privilegios
root	localhost	global	ALL PRIVILEGES	Sí	Editar los privilegios

Imagen 4. 36 Prueba de Caja Negra  
Elaborado por: El Investigador

- **Pruebas de Caja Blanca**

Estas pruebas sirven para identificar la funcionalidad del código, se realizaron con la finalidad de comprobar el funcionamiento completo de la Aplicación Web.

Para verificar y corregir posible errores presentados en la estructura interna de la Aplicación Web se realizaron pruebas descritas a continuación.

- Ingreso de Usuario

**CNE**  
CONSEJO NACIONAL ELECTORAL

Inicio | Registros | Documentos | Recorridos | Reporte | Bienvenidos

**Nuevo Usuario**

Usuario:

Contraseña:  Repetir Contraseña:

Identificación:

Nombre:  Apellido:

E-mail:

Cargos:  Roles:

Estado:

[Guardar](#)

Imagen 4. 37 Prueba Ingreso Usuario  
Elaborado por: El Investigador

**CNE**  
CONSEJO NACIONAL ELECTORAL

Inicio | Registros | Documentos | Recorridos | Reporte | Bienvenidos

**Usuarios**

Asunto:  [Buscar](#)

Id	Identificación	Nombre	Cargo	Role		
2	1804528386	Juan	Gerente	Gerente	<a href="#">Editar</a>	<a href="#">Eliminar</a>
3	0201928067	Katerine	Gerente	Gerente	<a href="#">Editar</a>	<a href="#">Eliminar</a>
4	0201	Marisol	Secretaria General	Secretaria General	<a href="#">Editar</a>	<a href="#">Eliminar</a>
5	0202	Hugo	Jefe Centro de Computo	Jefe Centro de Computo	<a href="#">Editar</a>	<a href="#">Eliminar</a>

[Agregar Usuarios](#)

10

Página 1 de 1  
Registros por página:

Imagen 4. 38 Comprobación del Ingreso de Usuario  
Elaborado por: El Investigador

- Ingreso de Rol





Imagen 4. 39 Prueba ingreso Rol  
Elaborado por: El Investigador

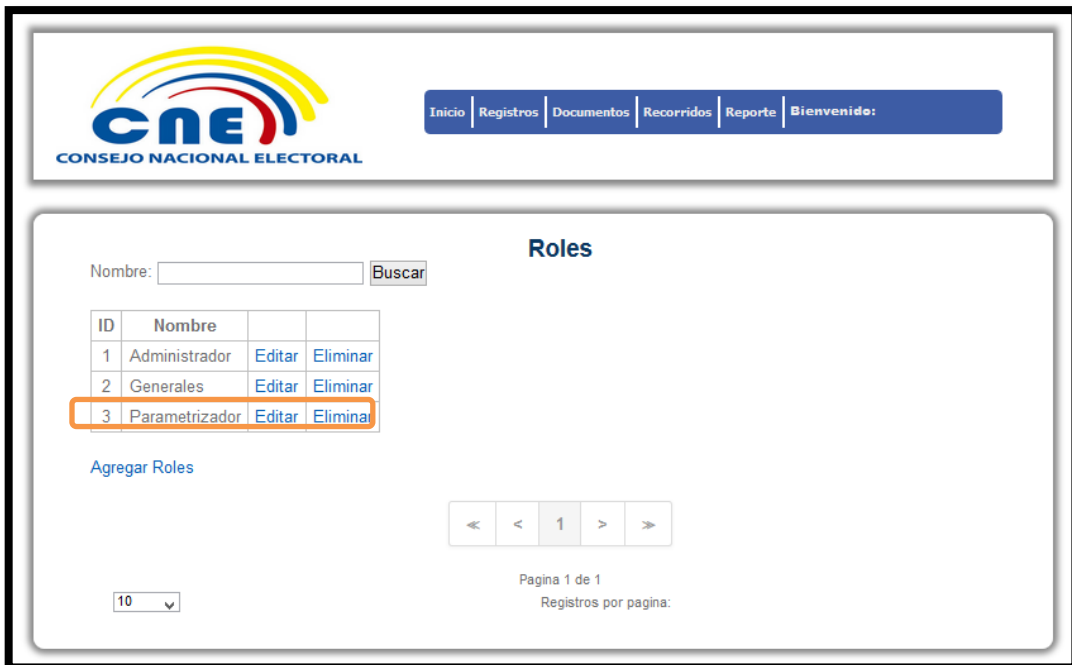



Imagen 4. 40 Comprobación ingreso de rol  
Elaborado por: El Investigador


- **Ingreso de Departamento**

  
**CONSEJO NACIONAL ELECTORAL**

[Inicio](#) | [Registros](#) | [Documentos](#) | [Recorridos](#) | [Reporte](#) | [Bienvenido:](#)

**Nuevo Departamentos**  
 Nombre:

Imagen 4. 41 Prueba Ingreso Nuevo Departamento  
Elaborado por: El Investigador

  
**CONSEJO NACIONAL ELECTORAL**

[Inicio](#) | [Registros](#) | [Documentos](#) | [Recorridos](#) | [Reporte](#) | [Bienvenido:](#)

**Departamentos**  
 Nombre:

Codigo	Nombre	Estado		
3	Juridico	1	<a href="#">Editar</a>	<a href="#">Eliminar</a>
2	Recursos Humanos	1	<a href="#">Editar</a>	<a href="#">Eliminar</a>
1	Sistemas	1	<a href="#">Editar</a>	<a href="#">Eliminar</a>

[Agregar Departamentos](#)

Pagina 1 de 1  
 Registros por pagina:

Imagen 4. 42 Comprobación Nuevo Departamento  
Elaborado por: El Investigador

- **Editar Departamento**



Imagen 4. 43 Prueba Editar Departamento  
Elaborado por: El Investigador



Imagen 4. 44 Comprobación Editar Departamento  
Elaborado por: El Investigador

#### 4.30. Discusión de Resultados

Después de la implementación de la Aplicación Web como propuesta a la solución del problema planteado, se realiza un cuadro comparativo en el cual se puede identificar los beneficios del nuevo método de la gestión documental sobre el anterior método.

Aspecto	Proceso Anterior	Proceso Posterior
<b>Almacenamiento de información</b>	En Archivadores, Hojas de Cálculo, carpetas.	Base de datos configurada en MySQL
<b>Búsqueda de la Información.</b>	Revisión manual de documentación.	Acceso rápido a la información.
<b>Seguridad de la información.</b>	Acceso a toda la información por cualquier persona con acceso a la Delegación.	Acceso a la información únicamente con el usuario asignado por el administrador.
<b>Deterioro de la información.</b>	Deterioro de hojas de papel por la manipulación de las mismas.	La información digital almacenada en la base de datos no se deteriora físicamente.
<b>Consistencia</b>	Registros inconsistentes por la duplicidad de información.	Registros actualizados y centralizados en la base de datos.
<b>Control de Actividades del Personal</b>	Mediante observación de las actividades realizadas departamento por departamento.	Mediante reportes y tiempo de procesamiento de una solicitud.

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1. CONCLUSIONES**

- Se determina que los canales de comunicación entre los departamentos de la Delegación Electoral de Bolívar son: documentos escritos, correo electrónico y telefonía IP.
- El método para determinar el estado de un documento dentro de un determinado proceso en la Delegación Electoral de Bolívar es ineficiente.
- El método utilizado para la gestión documental en la Delegación Electoral de Bolívar, ocasiona un ineficaz procesamiento de información y generación de informes.
- Los equipos hardware que tiene la Delegación Electoral de Bolívar, cumplen con los requerimientos básicos para un correcto funcionamiento de la aplicación web.
- El levantamiento y análisis de requerimientos ayudaron a generar los diagramas de casos de uso los mismos que aportaron para el desarrollo de la aplicación web.
- Las pruebas de funcionamiento ayudaron a encontrar y corregir errores para tener un excelente desempeño de la aplicación web.
- La implementación de la aplicación web permitió automatizar la Gestión Documental de la Delegación Electoral de Bolívar.

## 5.2. RECOMENDACIONES

- Se recomienda a la Delegación Electoral de Bolívar brindar una capacitación a los usuarios designados a interactuar con el sistema, para evitar la resistencia al cambio.
- Se recomienda a los usuarios del sistema mantener la comunicación correcta y oportuna con el administrador para evitar errores en los procesos de la Aplicación Web.
- Se recomienda que los usuarios de la aplicación web tengan un grado de confidencialidad con las claves otorgadas por el administrador.
- Se recomienda al administrador realizar un mantenimiento y respaldo del sistema periódicamente.
- Se recomienda a los usuarios trabajar con Mozilla Firefox o Google Chrome como navegador predeterminado para ejecutar la aplicación, debido a la mejor administración de estos con los cookies en la Web.
- Se recomienda a los usuarios prestar la debida atención a los mensajes de advertencia que muestra la aplicación web.

## Bibliografía

- [1] Sistema de Gestión Documental [online]. Disponible en:  
[http://es.wikipedia.org/wiki/Gesti%C3%B3n\\_documental](http://es.wikipedia.org/wiki/Gesti%C3%B3n_documental)
- [2] Aplicación Web [online]. Disponible en:  
[http://www.suronline.net/nuevo\\_sitio/beneficios-funcionamiento-aplicaciones-web.asp](http://www.suronline.net/nuevo_sitio/beneficios-funcionamiento-aplicaciones-web.asp)
- [3] Luján Mora Sergio, Programación de aplicaciones web: historia, principios básicos y clientes web, Ed. Club Universitario, (2002).
- [4] Base de datos [online]. Disponible en:  
<http://www.maestrosdelweb.com/editorial/%C2%BFque-son-las-bases-de-datos/>
- [5] Lenguaje de Programación Web [online]. Disponible en:  
[http://www.ecured.cu/index.php/Lenguaje\\_de\\_Programaci%C3%B3n\\_Web](http://www.ecured.cu/index.php/Lenguaje_de_Programaci%C3%B3n_Web)
- [6] Lenguaje de Programación Web [online]. Disponible en:  
<http://www.larevistainformatica.com/lenguajes-programacion-web.htm>
- [7] Servidor Web[online]. Disponible en:  
<http://php.net/manual/es/features.commandline.websserver.php>
- [8] Apache [online]. Disponible en: <http://www.digitalllearning.es/blog/apache-servidor-web-configuracion-apache2-conf/>
- [9] Interfaz gráfica [online]. Disponible en:  
<https://prezi.com/stkgahgrjbsn/interfaz-grafica-de-usuario/>
- [10] Sistemas Gestores de Base de Datos [online]. Disponible en:  
<http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>
- [11] MySQL [online]. Disponible en:  
<http://www.mastermagazine.info/termino/6051.php>
- [12] Lenguajes de Programación [online]. Disponible en:  
<http://www.areatecnologia.com/informatica/lenguajes-de-programacion.html>
- [13] PHP [online]. Disponible en:  
<http://www.webestilo.com/php/>
- [14] Ventajas, Inconvenientes [online]. Disponible en:  
[https://prezi.com/03s9bewast7\\_/que-es-un-php/](https://prezi.com/03s9bewast7_/que-es-un-php/)

[15] Diccionario de Datos [online]. Disponible en:

[http://www.elprofesionaldelainformacion.com/contenidos/1997/diciembre/una\\_propuesta\\_de\\_metodologia\\_para\\_el\\_diseo\\_de\\_bases\\_de\\_datos\\_documentales\\_parte\\_ii.html](http://www.elprofesionaldelainformacion.com/contenidos/1997/diciembre/una_propuesta_de_metodologia_para_el_diseo_de_bases_de_datos_documentales_parte_ii.html)

[16] Modelo-Vista-Controlador [online]. Disponible en:

<http://book.cakephp.org/2.0/es/cakephp-overview/understanding-model-view-controller.html>



**ANEXOS**

## **Anexo N° 1.**

### **Guía de Observación**

#### **GUIA DE OBSERVACION**

**Objetivo:** Recolectar la información sobre la Gestión Documental en la Delegación Electoral de Bolívar

**Tipo de Observación:** Observación Directa

#### **Indicadores a Observar**

##### **Numero de Departamentos**

Son en total 15 departamentos detallados:

- Dirección
- Talento Humano
- Comunicación Social
- Ventanilla Única
- Jurídico
- Administrativo
- Centro de cómputo
- Capacitación
- Organizaciones políticas
- Secretaria general
- Recaudación
- Compras públicas
- Planificación
- Financiero
- Seguridad

### **Número de Funcionarios**

- El número de funcionarios que laboran dentro de la institución son 28

### **Usuarios que generan tramites en la institución**

- Funcionarios Internos
- Ciudadanía
- Partidos Políticos

### **Medios que se almacena la información**

- Hojas de Calculo
- Carpetas físicas
- Archivos de texto
- Muebles Archivadores

### **Documentos que se generan**

- Registro de ingreso y egreso de correspondencia
- Oficios con requerimientos de otras instituciones y de la ciudadanía
- Memorandos con requerimientos de las diferentes unidades del Consejo Nacional Electoral
- Documentos Certificados
- Informes de actividades, productos y servicios.

### **Disponibilidad de la información:**

- Búsqueda de archivos digitales
- Búsqueda de archivos físicos

### **Demanda de trámites**

- Tiene una demanda de alrededor de 20 tramites diarios

## **Forma de gestión de información entre departamentos**

- La información se gestiona de manera manual

## **Anexo N° 2**

### **Entrevista**

Las preguntas fueron realizadas a los dos departamentos, se detalla un resumen de las entrevistas realizadas:

**1. ¿Cómo considera Ud. que está funcionando la gestión documental en la Institución?**

La Gestión Documental dentro de la Delegación Electoral de Bolívar se está llevando de forma manual.

**2. ¿Cuántos y cuáles son los departamentos existentes en la Delegación Electoral de Bolívar?**

La delegación electoral de Bolívar trabaja con 15 departamentos cuando esta no se encuentra en proceso electoral.

- Dirección
- Talento Humano
- Comunicación Social
- Ventanilla Única
- Jurídico
- Administrativo
- Centro de cómputo
- Capacitación
- Organizaciones políticas
- Secretaria general
- Recaudación
- Compras públicas
- Planificación
- Financiero
- Seguridad

**3. ¿Cuántas personas trabajan en cada departamento?**

Estimando de uno a dos funcionarios por cada departamento.

**4. ¿Cómo se procesa un trámite y que departamentos interactúan en este proceso?**

Esto depende del tipo de trámite que se esté llevando a cabo, por ejemplo si es un trámite pedido por un cliente externo este pasa por 3 departamentos de la Delegación que son:

- Ventanilla única
- Secretaria General
- Centro de Cómputo.

**5. ¿Qué departamentos pueden generar necesidades mediante un documento escrito?**

Todos los departamentos son capaces de crear una necesidad siendo esta requerida al departamento administrativo para que este se encargue de aprobar o negar la necesidad con los demás departamentos involucrado en el proceso.

**6. ¿Si un departamento genera un pedido, cómo puede verificar Ud. O el departamento que genera la necesidad la situación en la que se encuentra el documento en cuestión?**

Para verificar en qué estado se encuentra el proceso se debe buscar la carpeta donde conste todo el trámite y observar cual ha sido el último responsable del departamento que ha firmado y recibido el mencionado trámite.

**7. Estaría Ud. De acuerdo en la automatización del proceso de gestión Documental con una aplicación Web?**

Si debido a que toda institución mejora sus procesos mediante la implementación de las nuevas tecnologías.

Anexo N° 3

Formulario de Solicitud Gestionado Manualmente

**FORMULARIO DE SOLICITUD**

**CNE**  
CONSEJO NACIONAL ELECTORAL

USD. 2.00

N° 193074

Guaranda, 17 de octubre del 2014

Ingeniero  
Luis Alfonso Rivera Guerra  
**DIRECTOR PROVINCIAL CNE-BOLIVAR**  
Presente

De mi consideración:

Yo, **Segundo Arturo Chimbo Cando** con cédula de identidad N° 020108967-9, me dirijo hacia usted para solicitarle se me desafilie del Movimiento de Unidad Prurinacional Pachakutik Lista 18, por motivos personales.

Por la atención que se brinde dar a la presente, reitero mi agradecimiento.

**CONSEJO NACIONAL ELECTORAL**

Atentamente,

*[Firma]*

Sr. **Segundo Arturo Chimbo Cando**  
Solicitante

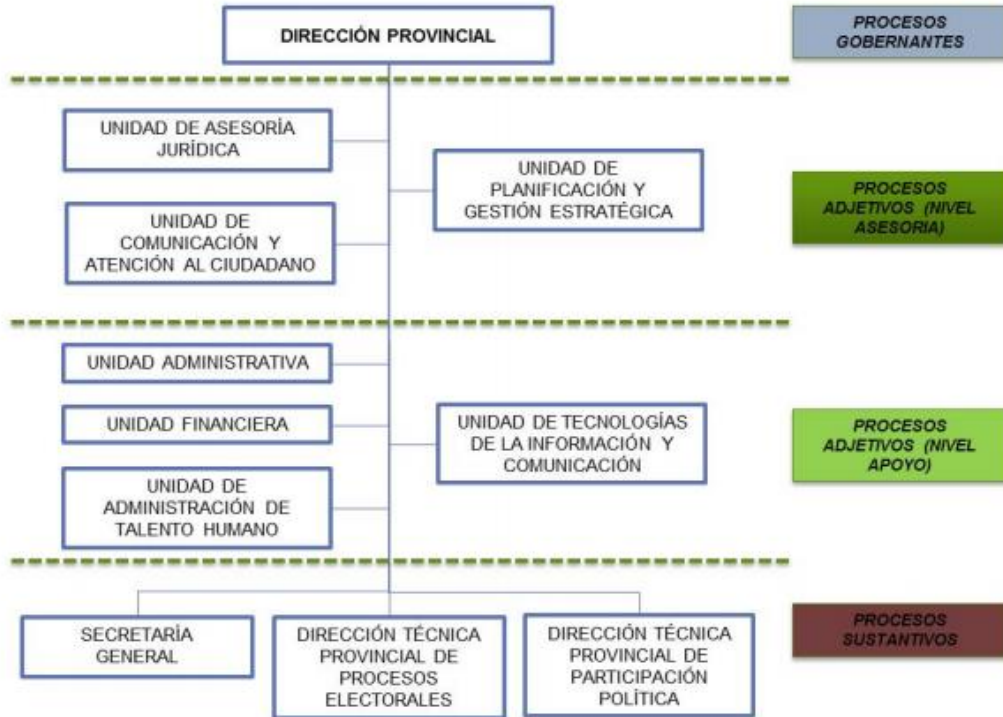
DELEGACION PROVINCIAL ELECTORAL DE BOLIVAR  
CNE Ventanilla Unica de Atención al Ciudadano  
Recibido por: *Clara Pazmiño*  
Fecha: *17-10-2014* Hora: *15:01*  
Hojas Anexas: *Una*  
Firma: *[Firma]*

*Si omite la desafiliación, con Tar en la OP y solicitó 20/10/2014*

*AB. Moposita Para su atención*

Anexo N° 4

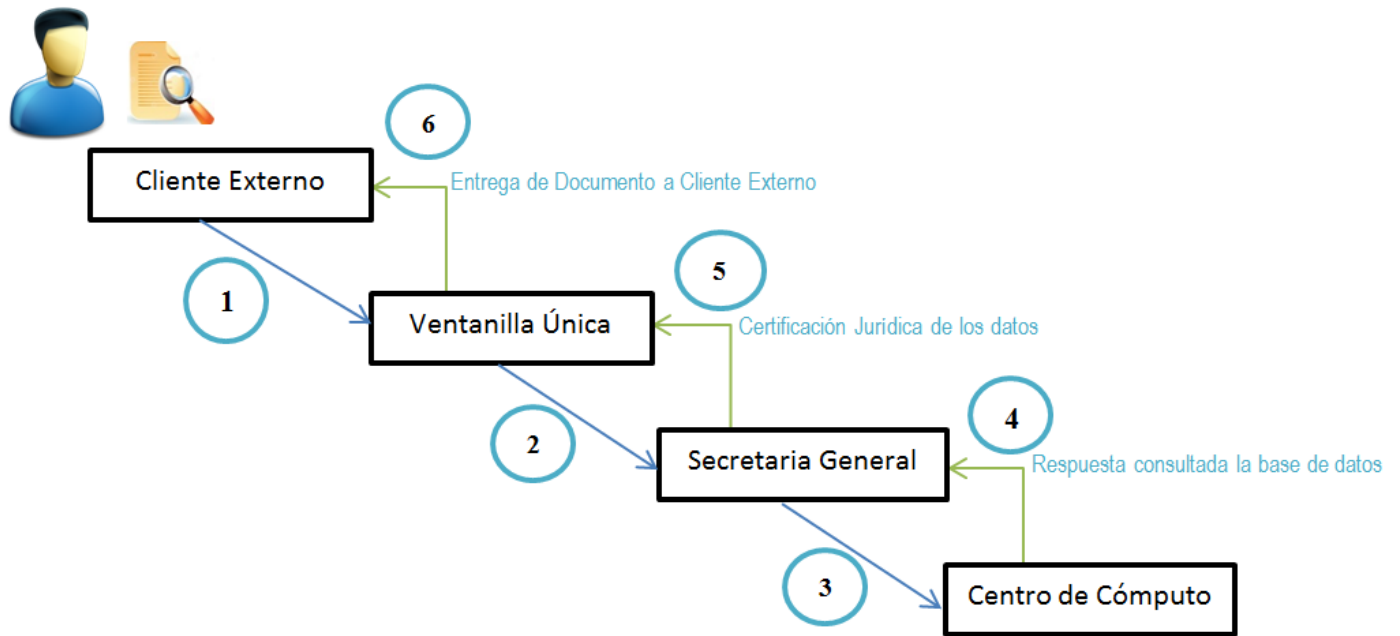
Organigrama Estructural Delegación Electoral de Bolívar





## Anexo N°5

### SEGUIMIENTO DEL DOCUMENTO “PEDIDO DE INFORMACION CLIENTE EXTERNO”



## Anexo N°6

