



UNIVERSIDAD TÉCNICA DE BABAHOYO

**FACULTAD DE ADMINISTRACIÓN, FINANZAS E
INFORMÁTICA**

PROCESO DE TITULACIÓN

OCTUBRE 2017 – MARZO 2018

PROPUESTA TECNOLÓGICA

INGENIERIA EN SISTEMAS

PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN SISTEMAS

TEMA:

**APLICACIÓN WEB PARA AUTOMATIZAR LOS PROCESOS
ELECTORALES EN LA UNIVERSIDAD TÉCNICA DE BABAHOYO.**

EGRESADA:

AMELIA ALEXANDRA ESTRADA ALBIÑO

TUTOR:

ING. HUGO JAVIER GUERRERO TORRES. MGRT

AÑO 2018

INDICE

INDICE	II
INDICE DE ILUSTRACIONES	III
INTRODUCCIÓN	IV
1. CAPITULO I. DIAGNOSTICO DE NECESIDADES Y REQUERIMIENTOS	
1	
1.1. Ámbito de Aplicación: Descripción del contexto y hechos de interés	1
1.2. Establecimiento de requerimientos	3
1.2.1. Metodología.....	5
1.2.2. Arquitectura	5
1.2.3. Requerimientos de funcionalidad	6
1.2.4. Requerimientos de desarrollo	7
1.3. Justificación del requerimiento a satisfacer	7
2. CAPITULO II. DESARROLLO DEL PROTOTIPO	10
2.1. Definición del prototipo tecnológico	10
2.2. Fundamentación teórica del prototipo	10
2.2.1. Página web.....	10
2.2.2. HTML.....	11
2.2.3. Lenguaje CSS	12
2.2.4. PHP.....	13
2.2.5. SQL y MySQL	14
2.2.6. LENGUAJE UML.....	15
2.2.7. Servidor web.....	15
2.2.8. Apache	16
2.2.9. Arquitectura Cliente – Servidor.....	17
2.2.10. Patrón de desarrollo de software MVC.....	18
2.2.10.1. Modelos	18
2.2.10.2. Vistas	18
2.2.10.3. Controladores	19
2.3. Objetivos del prototipo	19
2.4. Diseño del prototipo.....	20
2.4.1. Fase de Análisis	20
2.4.2. Fase de Diseño.....	20
2.4.3. Fase de Desarrollo	44
2.5. Ejecución y/o ensamblaje del prototipo.....	46
3. CAPITULO III. EVALUACION DEL PROTOTIPO	54
3.1. Evaluación Técnica.....	54
3.2. Evaluación Económica	54
3.3. Evaluación de Funcionalidad.....	55
4. CONCLUSIONES Y RECOMENDACIONES.....	56
4.1. Conclusiones.....	56
4.2. Recomendaciones.....	56
5. BIBLIOGRAFIA.....	57

INDICE DE ILUSTRACIONES

Ilustración 1	Funcionalidad del perfil Administrador.....	20
Ilustración 2	Funcionalidad del perfil Integrantes de Junta Receptora del Voto.....	20
Ilustración 3	Funcionalidad del perfil Sufragante.....	21
Ilustración 4	Diagrama de Caso de Uso General.....	21
Ilustración 5	Inscribir procesos electorales.....	22
Ilustración 6	Asignar procesos electorales.....	22
Ilustración 7	Inscribir candidatos.....	22
Ilustración 8	Cargar registro electoral.....	23
Ilustración 9	Crear padrón electoral.....	23
Ilustración 10	Asignar juntas receptoras de voto.....	23
Ilustración 11	Designar integrantes de JVR.....	24
Ilustración 12	Proclamar resultados generales.....	24
Ilustración 13	Verificar sufragante en padrón electoral.....	24
Ilustración 14	Emitir resultados parciales.....	25
Ilustración 15	Realizar votación.....	25
Ilustración 16	Diagrama de Actividad Sufragante.....	26
Ilustración 17	Diagrama de Actividad Integrante Junta Receptora de Voto.....	27
Ilustración 18	Diagrama de Actividad Administrador.....	28
Ilustración 19	Diagrama de Secuencia Sufragante.....	29
Ilustración 20	Diagrama de Secuencia Integrante Junta Receptora de Voto.....	30
Ilustración 21	Diagrama de Secuencia Administrador.....	31
Ilustración 22	Diagrama de Clases.....	32
Ilustración 23	Modelo Conceptual.....	32
Ilustración 24	Modelo Lógico.....	33
Ilustración 25	Interfaz principal.....	44
Ilustración 26	Interfaz de MySQL.....	44
Ilustración 27	Interfaz de NetBeans IDE.....	45
Ilustración 28	Interfaz Acceso a la aplicación web.....	46
Ilustración 29	Interfaz Proceso electoral.....	47
Ilustración 30	Interfaz Candidatos.....	48
Ilustración 31	Interfaz Registro electoral.....	48
Ilustración 32	Interfaz Junta receptora del voto.....	49
Ilustración 33	Interfaz Resultados Generales.....	49
Ilustración 34	Reporte de resultados generales.....	50
Ilustración 35	Interfaz Padrón electoral.....	51
Ilustración 36	Reporte de padrón electoral.....	51
Ilustración 37	Interfaz Resultados.....	52
Ilustración 38	Reporte de resultados parciales.....	52
Ilustración 39	Interfaz Votación en línea.....	53

INTRODUCCIÓN

Con el acontecer de los años la tecnología se ha ido apoderando de nuestro mundo con nuevas invenciones que automatizan diferentes procesos, que se los ha venido realizando de forma manual y por tal motivo han tomado mucho tiempo en la realización de los mismos.

Las instituciones tanto públicas y privadas en algún momento se han visto en la necesidad de optar por la adquisición de tecnología, para la automatización de diferentes procesos, salvaguardar la información o necesariamente para brindar un mejor servicio a sus usuarios finales, esto a su vez ayuda a ser una institución más competitiva en el ámbito en el cual se desempeñan.

La institución de educación superior Universidad Técnica de Babahoyo con el pasar de los años se ha visto en la necesidad de ir adoptando sistemas informáticos capaces de realizar múltiples tareas facilitando el trabajo diario y mejorando la interacción con la comunidad universitaria.

Se desarrolló una aplicación web para la automatización de los procesos electorales de la Universidad Técnica de Babahoyo en sus categorías: rector, vicerrectores, representantes docentes, representantes estudiantiles, representantes por los graduados y representantes por los trabajadores. De los cuales los representantes se eligen para formar parte del consejo universitario y consejo directivo de facultades. Estos procesos electorales se realizan mediante votaciones independientes, de acuerdo al Reglamento de Régimen Electoral de la Universidad Técnica de Babahoyo.

Para obtener claramente las necesidades que el sistema debe cumplir se realizó un análisis de contexto, el cual permite conocer las decadencias que tiene el proceso actual de elecciones en la universidad. En base a las necesidades encontradas se ha plasmado los requerimientos funcionales que el prototipo cubre según el usuario que conste en el sistema.

Para el desarrollo del prototipo se basó en la arquitectura cliente – servidor, ya que permite la interacción de manera eficaz entre varios clientes y el servidor al mismo tiempo, esto es posible gracias a las aplicaciones de tipo web. Se aplica metodología MVC que permite la separación de código en tres capas permitiendo a futuro la reutilización del mismo.

Entre los conceptos que se ha sido importante revisar para el desarrollo de la aplicación están herramientas del lado del cliente que son: HTML5, CSS y JavaScript. Mientras que las herramientas necesarias para procesar del lado del servidor se utiliza: PHP, MySQL y Apache.

Para obtener una idea clara de lo que se construyó fue importante plasmar el diseño mediante diagramas de casos de uso, donde se especifican las tareas que cada usuario deberá cumplir de acuerdo al perfil que se le haya asignado sea este: sufragante, integrante de junta receptora o administrador del sistema.

1. CAPITULO I. DIAGNOSTICO DE NECESIDADES Y REQUERIMIENTOS

1.1. Ámbito de Aplicación: Descripción del contexto y hechos de interés

El lugar hacia donde se encuentra dirigido la elaboración del prototipo es en la Universidad Técnica de Babahoyo situada en el cantón Babahoyo perteneciente a la provincia Los Ríos. Es una institución que se dedica a la formación de profesionales en educación de tercer nivel.

El prototipo denominado como aplicación web para la automatización de los procesos electorales de la institución, surge como una estrategia capaz de procesar información de manera inmediata, dando así solución a ciertos inconvenientes que se suscitan mediante la realización de dicho evento de la manera tradicional.

La forma tradicional de ejecutar los procesos electorales universitarios requiere la inclusión del factor económico, factor humano, y sobre todo de factor tiempo. Basado en esto, la realización de dicho proceso mediante una aplicación web lo hace más sencillo evitando contratiempos y la inversión que conlleva el mismo.

Tomando en cuenta el factor económico, en la actualidad para poner en ejecución un proceso electoral dentro de la universidad implica gastos para diferentes procesos, entre ellos podemos mencionar los siguientes: gastos en impresión de papeletas e impresión de padrón electoral y demás materiales.

En cuanto al factor humano se enfoca en reducir el trabajo del personal que está a cargo de realizar el conteo voto a voto ya que esta tarea la aplicación lo realizara de manera automática y sobre todo demostrando transparencia en los procesos en cuestión.

En cambio el factor tiempo, actualmente los votantes tienen que formar largas colas de espera para el cumplimiento de su votación, en cuanto a esto se reduce el tiempo de respuesta, porque basta presentarse con su identificación personal y se verificara su existencia en el padrón electoral, los miembros de la juntas receptoras del voto realizan

su trabajo de manera más efectiva y por otro lado para el tribunal electoral les ahorra mucho tiempo al obtener los resultados finales de forma inmediata al cierre del proceso electoral.

En la actualidad en nuestro medio el Consejo Nacional Electoral, CNE cuenta con un sistema de voto electrónico que es capaz de llevar a cabo el proceso electoral mediante el uso de máquinas de votación, papeletas y ánforas.

El voto electrónico admite que el votante maneje una sola papeleta y no como actualmente sucede una por cada dignidad que se vaya a elegir.

Cada sufragante recibe una papeleta en blanco, la cual contiene un chip que guarda toda la información del voto. Esta papeleta se fija en una ranura del artefacto de votaciones y el usuario elige en una pantalla táctil a los candidatos de su preferencia, estos a su vez están congregados en categorías: prefectos, viceprefectos y alcaldes, concejales y representantes de las juntas parroquiales.

Se agiliza la forma de llevar los procesos electorales, esto se da mediante una metodología fácil y sencilla por la cual el votante puede cumplir con su obligación al voto asistiendo a la respectiva junta receptora de votación con su identificación, y de esta manera se le habilita su respectivo usuario en el sistema, posterior a esto puede realizar su votación.

Estableciendo una breve relación entre el sistema de voto electrónico del CNE, y el prototipo que se denota la función principal de este proceso que es reducir costos, agilizar la proclamación de resultados y sobre todo que se cumpla bajo los principios de transparencia y derecho al voto secreto.

1.2. Establecimiento de requerimientos

Para definir los requerimientos fue importante basarse en la descripción del contexto ya que mediante este estudio se pueden obtener diferentes situaciones problemáticas que necesitan ser tratadas y para ello se citan a continuación las necesidades que se desean cubrir con el prototipo:

- Realizar los procesos electorales en sus categorías: Rector, Vicerrectores (Vicerrector Académico, Vicerrector de Investigación y Posgrado, Vicerrector Administrativo y Financiero), y representantes docentes, estudiantiles, graduados, de servidores y trabajadores.
- Permitir la inscripción de candidaturas en los procesos electorales de la Universidad Técnica de Babahoyo de acuerdo a los principios de equidad.
- Realizar el proceso electoral de manera transparente garantizando el voto secreto y la proclamación de resultados.
- Los miembros de la junta receptora del voto deben emitir documento con resultados parciales al cierre del proceso electoral.
- El tribunal electoral debe visualizar e imprimir los resultados finales del proceso electoral en cuestión.
- La valoración del voto debe ser considerada de acuerdo a la ponderación que se otorga según la categoría a la cual pertenece, sea este: docente, estudiante o trabajador para la elección de rector y vicerrectores.

De acuerdo a las funciones que cumple el Tribunal Electoral Universitario, Juntas Receptoras del Voto y Sufragantes se otorga permisos necesarios a los miembros de estas entidades para manipular la información correspondiente en la aplicación web. A continuación se citan las principales acciones que deben realizar cada usuario:

Administrador de la aplicación web:

- Encargado de inscribir procesos electorales.
- Inscribir a los candidatos a participar en el proceso.
- Cargar registro electoral al sistema.
- Crear padrón electoral con sus respectivos integrantes.
- Asignar el número de juntas receptoras del voto de acuerdo al número de votantes hábiles para el sufragio.
- Designar a los integrantes de las juntas receptoras del voto como mínimo tres vocales máximo cinco.
- Distribuir el padrón electoral de acuerdo al número de juntas receptoras del voto.
- Proclamar resultados del proceso electoral y emitir documento del mismo.

Integrantes de las Juntas Receptoras del voto

- Verificar la identidad del sufragante y habilitar la votación.
- Emitir acta de resultados parciales tomando en cuenta los votos válidos, nulos y en blanco.

Sufragante

- Elegir categoría de proceso electoral.
- Realizar proceso de votación. Puede directamente votar en blanco o simplemente votar entre lista según sea su preferencia.

1.2.1. Metodología

Fue necesario aplicar como método inductivo una entrevista con el presidente del Tribunal Electoral Universitario, y en base a ello se conoció la forma en que se ejecutan los procesos electorales y los métodos que se siguen para el cumplimiento del mismo.

La metodología aplicada en el desarrollo de software es el patrón Modelo Vista Controlador (MVC), ya que está basado en la separación de código de manera organizada de tal forma que a futuro se puedan implementar nuevos módulos con gran facilidad.

1.2.2. Arquitectura

Para la creación del prototipo se aplica la arquitectura Cliente – Servidor que permite a varios clientes realizar peticiones de información mediante la aplicación web y el servidor envía respuestas en base a la información que se encuentra almacenado en el mismo.

El prototipo hace énfasis a la línea de investigación denominado Desarrollo de Sistemas de la información, comunicación y emprendimientos empresariales y tecnológicos. Sobre todo se basa en la sublínea Desarrollo de Sistemas Informáticos.

1.2.3. Requerimientos de funcionalidad

Requerimiento equipo Cliente

Software

- Sistema Operativo Windows 10.
- Poseer navegador de internet. (Google Chrome Versión 63.0.3239.132, Mozilla Firefox Versión 58.0.2)

Hardware

- PROCESADOR AMD E1-2100 1.0GHz Dual-Core.
- Memoria 2GB.
- Monitor SAMSUNG 19.5" LED.

Requerimiento equipo Servidor

Software

- Sistema Operativo Windows 10.
- Poseer navegador de internet. (Google Chrome Versión 63.0.3239.132, Mozilla Firefox Versión 58.0.2)

Hardware

- Procesador Intel I3-6100 3.7 GHz 6ta generación
- Disco Duro 2 Tb Hitachi
- Memoria RAM 4gb Kinston
- Monitor Samsung 19.5" Led.

1.2.4. Requerimientos de desarrollo

Software

- Plataforma de desarrollo NetBeans.
- Lenguaje de programación PHP.
- Servidor (Apache V2.4.2, MySQL V5.5.24, PHP V5.4.3).
- Editor de plantillas Artister V4.

Hardware

- Monitor LED HP de 20".
- Procesador Intel Core i5-4210U (2.40 GHz)
- RAM 4 GB.
- Disco Duro 500GB. (mínimo espacio disponible)

1.3. Justificación del requerimiento a satisfacer

El desarrollo de la aplicación web nació con la idea de automatizar los procesos electorales que se realizan en la Universidad Técnica de Babahoyo, debido a que se llevan a cabo distintas categorías y en diferentes periodos de tiempo de acuerdo al Reglamento de Régimen Electoral (Universidad Técnica de Babahoyo, 2014)

Las elecciones acogen procesos tediosos como: esperar largo tiempo para ejercer el voto, y el conteo de votos para finalmente obtener a un ganador. El prototipo ocasiona disminución de los tiempos para la ejecución dicho proceso y realiza de forma automática el conteo y da un ganador de manera eficaz y en el tiempo preestablecido.

Los beneficiarios directos son: el Tribunal Electoral Universitario al ser los encargados de poner en marcha este proceso, además se beneficia la comunidad universitaria ya que ejerciendo el papel de sufragante se facilita la manera en la que van a ejercer su voto, esto se puede realizar mediante unos pasos muy sencillos como: asistir a la junta receptora del

voto, identificarse, el integrante de la junta le habilitara en el sistema y ya puede proceder con la votación respectiva.

Además al momento de definir los requerimientos del prototipo se basa en los artículos referentes a los procesos electorales del Reglamento de Régimen Electoral de la Universidad Técnica de Babahoyo, los cuales citaremos a continuación:

Art. 1. ÁMBITO.- Las normas de este Reglamento rigen la elección de Rector, Vicerrectores, y representantes docentes, estudiantiles, graduados, de servidores y trabajadores ante los organismos de cogobierno de la Universidad Técnica de Babahoyo.

Art. 6.- COMPETENCIAS Y ATRIBUCIONES.- Corresponde al Tribunal Electoral Universitario:

- d) Elaborar los padrones electorales...
- f) Determinar el número de Juntas Receptoras del Voto, designar a sus integrantes y señalar el lugar en el que se instalarán. La designación de los integrantes de la Junta Receptora del Voto es inexcusable.
- g) Dividir los padrones proporcionalmente para el número de Juntas establecidas.
- h) Realizar el escrutinio final, proclamar los resultados y elaborar el acta respectiva, la misma que serán enviada de inmediato al Rectorado, para conocimiento del H. Consejo Universitario;

Art. 7.- DE LAS JRV.- Las Juntas Receptoras del Voto son organismos de gestión electoral de carácter temporal que se encargan de recibir los sufragios y efectuar los escrutinios, de conformidad con el presente Reglamento.

Cada Junta Receptora del Voto estará integrada con un mínimo de tres vocales y un máximo de cinco, dependiendo de la complejidad del proceso electoral. Será presidida por un profesor titular. Otro profesor titular preferentemente actuará como secretario; el

resto de vocales principales y suplentes será designado de entre los diversos estamentos de la comunidad universitaria.

Art. 8.- NÚMERO DE JUNTAS RECEPTORAS DEL VOTO.- Habrá al menos una Junta Receptora del Voto para profesores, una para servidores y trabajadores; y, una por cada 300 estudiantes como máximo. Si existiera un excedente de estudiantes luego de conformadas las Juntas, se dará paso a constituir una última Junta con dicho excedente.

Art. 9.- OBLIGACIONES DE LAS JUNTAS RECEPTORAS DEL VOTO.- Son obligaciones de la Junta Receptora del Voto:

- b) Verificar la identidad del sufragante;
- d) Realizar el escrutinio parcial una vez concluida la votación, tomando en cuenta los votos válidos, nulos, en blanco y papeletas no utilizadas. Elaborará y firmará el acta respectiva.

2. CAPITULO II. DESARROLLO DEL PROTOTIPO

2.1. Definición del prototipo tecnológico

Se desarrolló una aplicación web capaz de automatizar los procesos electorales que se realizan en la Universidad Técnica de Babahoyo enfocado en minimizar labores de quienes organizan y en facilitar la accesibilidad de quienes ejercen el voto. Entre las funcionalidades más relevantes del prototipo podemos citar las siguientes:

- Inscripción de procesos electorales.
- Proceso de inscripción de candidaturas.
- Contar con registro actualizado de las personas que han cumplido con su respectiva votación.
- Mostrar resultados de las candidaturas ganadoras en el tiempo ya preestablecido.

Para que este prototipo sea funcional se ha realizado haciendo uso las tecnologías de desarrollo web con sus lenguajes respectivos tanto para el backend (lado del servidor) y frontend (lado del cliente). Del lado del servidor se utiliza: PHP, MySQL y Apache. Mientras tanto para el desarrollo del lado del cliente se utiliza: HTML, JavaScript y CSS. De esta manera se logra que la aplicación web se convierta en un entorno atractivo y sobre todo de fácil navegación para el usuario final.

2.2. Fundamentación teórica del prototipo

2.2.1. Página web

Una página web dinámica interactiva es la que permite variar el aspecto y comportamiento en función de decisiones que toma el usuario a través de la interfaz gráfica de la página. Para ello se varía la estructura de la página. (Lopez, Vara, & Verde, 2014)

Para lograr interactividad en una página web, esta envía una petición al servidor de tal manera que el código se ejecuta en el servidor y este responde a su vez con una nueva página web en el navegador. La página web dinámica trabaja conforme a la información ingresada por el cliente, el sitio web cambiara de acuerdo a la forma en que haya sido definido en la programación.

El código que se ejecuta en el navegador cuando se ha realizado una petición, responde con una página web con extensión html, en caso que se esté utilizando un lenguaje de servidor respondería con la extensión asp, jsp o php.

2.2.2. HTML

El HTML (Hyper Text Markup Language), lenguaje de marcado de hipertexto, es un lenguaje de marcas (utiliza etiquetas, como marcas para delimitar elementos del lenguaje), que sirve para describir el contenido y la estructura de las páginas web, que pueden ser interpretadas y visualizadas a través de los navegadores de Internet (clientes web: Firefox, Internet Explorer, Chrome, etc.). (Zofio, 2014)

Permite publicar información que contengan tablas, imágenes, textos, etc., obtener información por medio de enlaces que conllevan a otras páginas web. Además HTML permite diseñar formularios para inserción de datos, búsqueda de información, etc. Pero el código HTML por sí solo no resuelve esas peticiones y para ello es necesario programación del lado del servidor, en este caso aparecen los lenguajes de programación PHP, JSP, etc.

Ventajas:

- Lenguaje sencillo, muy fácil de comprender.
- Existen aplicaciones capaces de generar el código HTML automáticamente.
- Permite realizar páginas ligeras que se carguen rápidamente en el navegador.

Desventajas:

- Los sitios web realizados en HTML son reconocidos de diferentes maneras dependiendo del navegador.
- Su código es limitado en cuanto a su sintaxis.

2.2.3. Lenguaje CSS

Este lenguaje es el encargado de definir la presentación de los datos que se indican en el código HTML. Para incluir las hojas de estilos CSS en la página web, es recomendable incluir un enlace a la dirección donde se ubica el fichero CSS. El navegador interpreta las características CSS. Este lenguaje es enviado directamente desde el servidor. El navegador interpreta el código CSS y lo aplica a los elementos HTML. En el caso de que exista código CSS que no esté relacionado con ningún HTML, este no afectará a la página web. (Lopez, Vara, & Verde, 2014)

La hoja de estilos no es más que un grupo de especificaciones que indican el formato mediante el cual se van a presentar los elementos HTML de un determinado sitio web en nuestro navegador.

Ventajas:

- Permite tener un solo archivo CSS y se puede utilizar en diversas páginas web, haciendo referencia mediante las etiquetas.
- Ayuda a mejorar el tiempo de respuesta del sitio web.
- Permite detallar el aspecto de una página web.

Desventajas:

- Puede existir código CSS que no se llegue a utilizar en nuestro sitio web.
- En caso de existir inconvenientes o limitaciones de coincidencias, el navegador aplicará el formato predeterminado.

2.2.4. PHP

Es un lenguaje de programación interpretado, que se ejecuta en el lado del servidor. Este lenguaje tiene una licencia de software libre y de código abierto. Es un lenguaje rápido, con una gran librería de funciones y abundante documentación. El tipo de implementación con respecto a cómo se combina con el código, es similar a ASP. El código PHP se interpreta en el servidor y se envía al navegador en forma de HTML. (Lopez, Vara, & Verde, 2014)

Ventajas de PHP:

- Se considera un lenguaje sencillo de comprender.
- Es rápido a la hora de procesar cada línea de código.
- Es soportado por varias plataformas de entorno web.
- Posee características de lenguaje orientado a objetos.
- De fácil conexión con base de datos.
- Se puede alojar en hosting.
- Se puede encontrar fácilmente información referente a este lenguaje lo cual facilita el desarrollo de cualquier prototipo.

Desventajas de PHP:

- Para verificar lo que se está creando con el lenguaje es necesario contar en nuestra pc con un servidor web.
- Es interpretado en el servidor lo cual hace que durante varias peticiones colapse el sitio web.
- El contenido en muchas ocasiones no es compatible con alguno de las versiones de los navegadores

2.2.5. SQL y MySQL

SQL (Lenguaje de Consulta Estructurado, Structured Query Language en inglés) es el lenguaje de acceso a base de datos relacionales más extendido. Con este sistema, el cliente especifica las instrucciones para crear, borrar o dotar de contenido las tablas de la base de datos, además de permitir su interrogación. (Blanco, 2013)

MySQL, a su vez, es un gestor de base de datos relacionales multiplataforma basado en SQL. Existen interfaces entre MySQL y numerosos lenguajes de programación habitualmente utilizados para diseñar programas servidores de internet. (Blanco, 2013)

Para que una aplicación web funcione necesita de una base de datos. Se puede trabajar con MySQL y para trabajar conjuntamente con Apache existe una herramienta llamada PhpMyAdmin. Esta herramienta permite crear una base de datos de manera gráfica, y para acceder a la misma se lo hace mediante un navegador.

Ventajas:

- Fácil de usar y rápido a la vez.
- Muy utilizado para la realización de aplicaciones web, se integra muy bien con PHP.
- Capaz de trabajar en entorno cliente-servidor.
- Permite trabajar con replicación de base de datos.

Desventajas:

- Algunas utilidades no se encuentran documentadas
- No es intuitivo.

2.2.6. LENGUAJE UML

UML es un lenguaje estándar para modelar sistemas de información. Permite expresar mediante símbolos gráficos la semántica deseada y especificar modelos completos con una mínima ambigüedad. Se puede establecer una correspondencia desde este modelo a un modelo orientado a objetos (objeto – relacionales u objetos puros). (Hueso, Diseño Conceptual y Logico de Base de Datos, 2014)

UML define las siguientes técnicas:

- ✓ Los diagramas de clases representan la vista de diseño estática y de procesos en términos de clases, relaciones, interfaces y colaboraciones. (Hueso, Diseño Conceptual y Logico de Base de Datos, 2014)
- ✓ Los diagramas de actividades representan el comportamiento de una operación en términos de acciones. (Hueso, Diseño Conceptual y Logico de Base de Datos, 2014)
- ✓ Los diagramas de caso de uso representan las funciones del sistema desde el punto de vista del usuario mediante un conjunto de casos de uso, actores y relaciones. (Hueso, Diseño Conceptual y Logico de Base de Datos, 2014)
- ✓ Los diagramas de secuencia son una representación temporal de los objetos y sus interacciones. (Hueso, Diseño Conceptual y Logico de Base de Datos, 2014)

2.2.7. Servidor web

Un servidor Web es quien se encarga de responder a las peticiones que realizan los usuarios, esta respuesta esta conformada por todos los archivos necesarios para que se pueda presentar la pagina web que anteriormente haya solicitado el usuario. Para que el servidor web cumpla con dicha funcion es necesario que trabaje con el Protocolo de Transferencia de Hipertexto HTTP . Los ordenadores y los aparatos que se an configurado para esta funcion también se los conoce como servidores Web.

2.2.8. Apache

Apache HTTP Server se encarga de procesar peticiones y posterior a eso entregar documentos de tipo web. Las repuestas que apache envía en base a las peticiones está en formato HTML.

Una de sus grandes características que llama la atención al ser de código abierto y sobre todo gratuito, es soportada por varias plataformas que usan sistemas operativos robustos.

Apache tiene como oficio confirmar las peticiones de los diferentes sitios web, este software es el encargado de negar o permitir el acceso a la información de acuerdo a las normativas establecidas, ya que son los usuarios quienes intentan acceder al sitio web en un determinado momento.

Ventajas:

- Contiene un servidor web integrado.
- Software multiplataforma.
- Es software de código abierto, además es gratuito.

Desventajas:

- Contiene formatos de instalación que no son estándares.
- No posee una muy buena administración.

2.2.9. Arquitectura Cliente – Servidor

El modelo cliente – servidor describe el proceso que engloba la iteración entre un computador local denominado CLIENTE y un computador remoto llamado SERVIDOR. Muchos clientes pueden realizar distintas peticiones a lo cual el servidor está obligado a responder a cada petición con la debida información que ha sido solicitada. Para que esta comunicación tenga éxito tanto el cliente como el servidor debe estar conectado a una red local o una red mundial como lo es el Internet.

Dentro del modelo cliente – servidor existen dos procesos: los clientes (front - end), que se encargan de solicitar peticiones de conexión para obtener información, y los servidores (back – end), que tratan esas peticiones, obtienen la información y la envían a los procesos clientes. (Villada, 2015)

Las características básicas de una arquitectura cliente – servidor se pueden resumir en la siguiente lista:

- El proceso cliente proporciona la interacción con el usuario y el resto del sistema, mientras que el servidor gestiona los recursos compartidos. (Villada, 2015)
- Los procesos cliente y servidor pueden ejecutarse en el mismo nodo.
- Un servidor puede dar soporte a múltiples clientes. (Villada, 2015)
- Los clientes realizan peticiones y, por lo tanto, son agentes activos, mientras que los servidores actúan como pasivos en la comunicación. (Villada, 2015)
- La relación entre los clientes y los servidores se limita a la comunicación del mensaje. (Villada, 2015)
- Es un sistema escalable tanto horizontal (a nivel de clientes) como vertical (a nivel de servidores). (Villada, 2015)

2.2.10. Patrón de desarrollo de software MVC

MVC es una idea de diseño de software que se usa para realizar sistemas que demanda diseño de interfaces para su funcionamiento. Nace con la idea de suplantar el desarrollo de software de la forma que actualmente o mediante otros mecanismos que se está desarrollando, es decir, este patrón trata de separar en capas su código para que se pueda volver a utilizar, brindar seguridad en su codificación y sobre todo mayor comprensión en caso de necesitar dar mantenimiento al software.

El MVC define tres capas para desarrollo del software, organiza el código mediante formatos determinados utilizando un razonamiento enfocado en la funcionalidad y no en las particularidades del elemento en sí mismo.

2.2.10.1. Modelos

En esta capa se enfoca en el modelado de datos, por tal motivo es quien se encarga de acceder a la información y procesarla de tal forma como haya sido solicitada. En realidad la funcionalidad de dicha capa es la de realizar las respectivas inserciones, selecciones, eliminaciones y modificaciones de datos en las tablas correspondientes a la base de datos con la que se encuentre asociado el software. Para realizar aquellos procesos es esencial que en esta capa existan todos los métodos necesarios capaces de cumplir con cada una de las funcionalidades.

2.2.10.2. Vistas

Contiene el código de la aplicación que muestra para la visualización de las interfaces de usuario, es decir, el código que permite renderizar los estados de la aplicación en HTML. En cada una de las vistas muestra el código de HTML y PHP.

Para que las vistas o interfaces del software funcione necesita trabajar con la codificación, para lo cual obtendrá esta información mediante otra capa. Los modelos

reciben las peticiones de las vistas y estos generan lo que se va a mostrar, tal como la aplicación lo necesite.

2.2.10.3. Controladores

Contiene la codificación necesaria para responder de manera adecuada a las peticiones que se envían desde la aplicación, esto puede ser mostrar algún módulo, ejecutar una adquisición, una averiguación de información, etc.

Es la capa que se utiliza de conexión entre la capa modelo y vista para la respectiva implementación de sus funcionalidades en base a las necesidades de la aplicación. Pero, el compromiso es manejar las conexiones entre las dos capas: modelo y vistas, es decir, esta capa funciona de acuerdo a lo que se haya solicitado mediante las entradas de la aplicación en cuestión.

2.3. Objetivos del prototipo

- Agilizar los procesos electorales que se realizan en la Universidad Técnica de Babahoyo.
- Transparentar los procesos electorales, para mejorar la confianza en los procesos de selección de autoridades.

2.4. Diseño del prototipo

2.4.1. Fase de Análisis

El Tribunal Electoral Universitario es el encargado de organizar y ejecutar los procesos electorales en la Universidad Técnica de Babahoyo, en sus distintas secciones o categorías, a medida que ejecutan dichos procesos electorales requiere: invertir dinero para las impresiones del material electoral, tiempo adecuado y prudencial para la votación de cada sufragante y sobre todo una vez que se cierran las votaciones continua con el proceso más tedioso lo cual toma muchas horas de trabajo y es realizar el conteo de los votos para poder proclamar a los ganadores oficiales de tal proceso electoral.

2.4.2. Fase de Diseño

La aplicación web está diseñada de tal manera que es capaz de ejecutar los procesos electorales en sus diferentes categorías de manera rápida y oportuna, en relación con la forma de ejecutar estos procesos de manera tradicional.

Funcionalidad de la Aplicación Web					
Registrar procesos electorales	Inscripción de candidatos	Obtener registro electoral	Crear padrones electorales	Designar juntas receptoras	Proclamar resultados generales.

Ilustración 1 Funcionalidad del perfil Administrador
Desarrollado por: Amelia Alexandra Estrada Albiño

Funcionalidad de la Aplicación Web		
Verificar identificación de sufragante	Habilitar votación a sufragante	Emitir acta de resultados

Ilustración 2 Funcionalidad del perfil Integrantes de Junta Receptora del Voto
Desarrollado por: Amelia Alexandra Estrada Albiño

Funcionalidad de la Aplicación Web	
Elegir categoría de elecciones	Realizar votación

Ilustración 3 Funcionalidad del perfil Sufragante
Desarrollado por: Amelia Alexandra Estrada Albiño

Diagrama de Caso de Uso

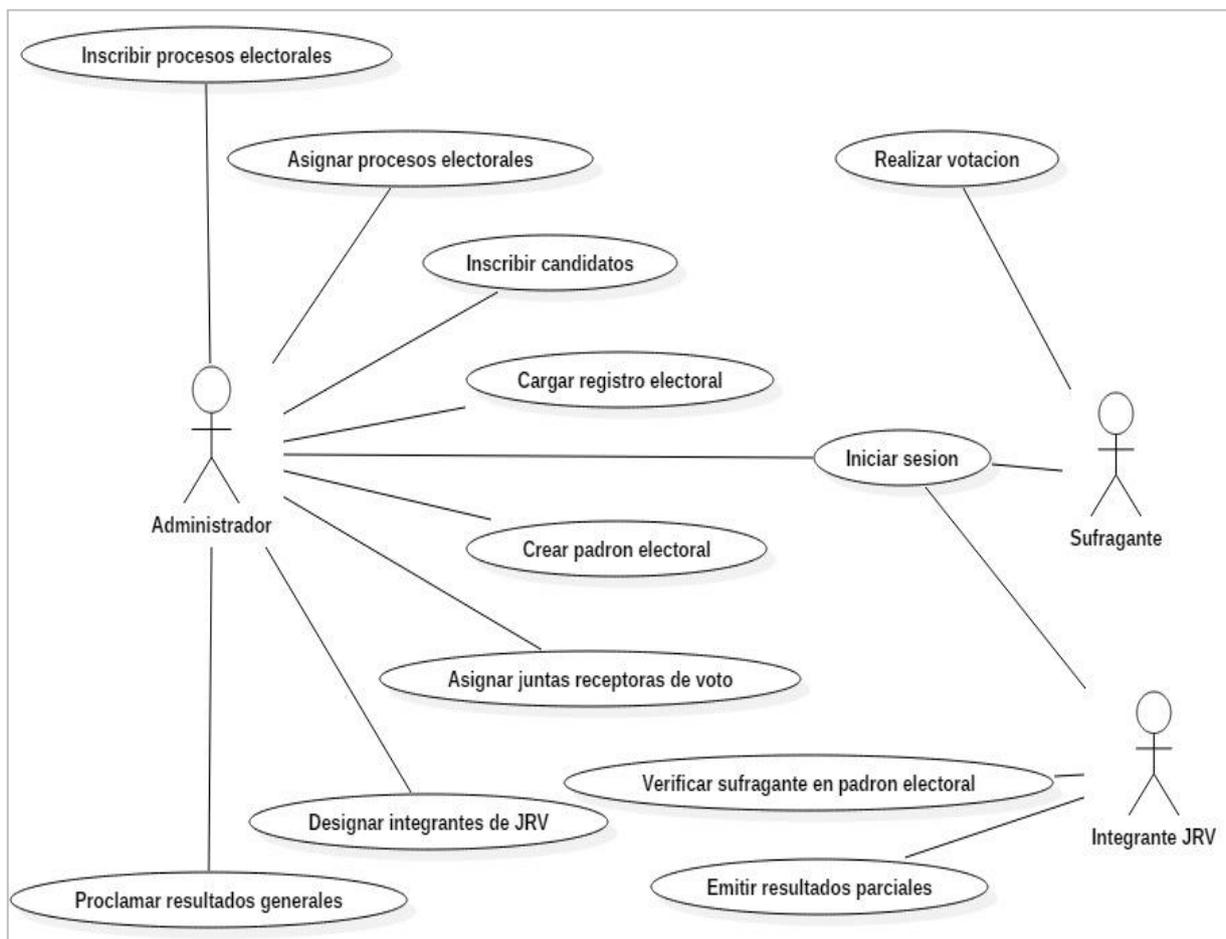


Ilustración 4 Diagrama de Caso de Uso General
Desarrollado por: Amelia Alexandra Estrada Albiño

Inscribir procesos electorales

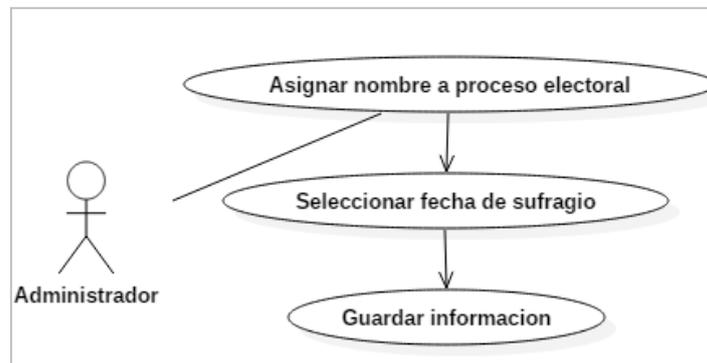


Ilustración 5 Inscribir procesos electorales
Desarrollado por: Amelia Alexandra Estrada Albiño

Asignar procesos electorales

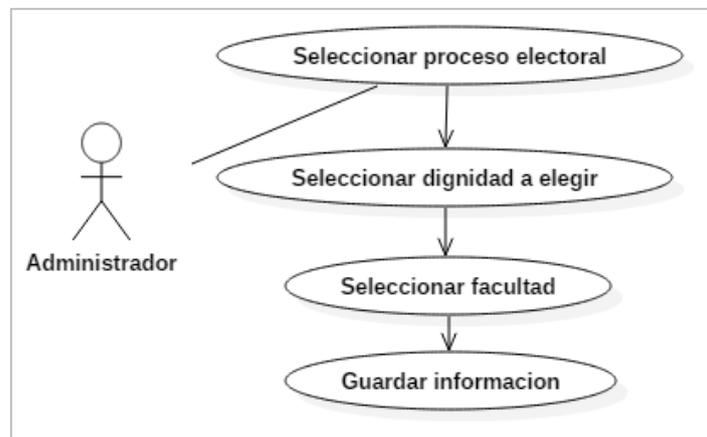


Ilustración 6 Asignar procesos electorales
Desarrollado por: Amelia Alexandra Estrada Albiño

Inscribir candidatos

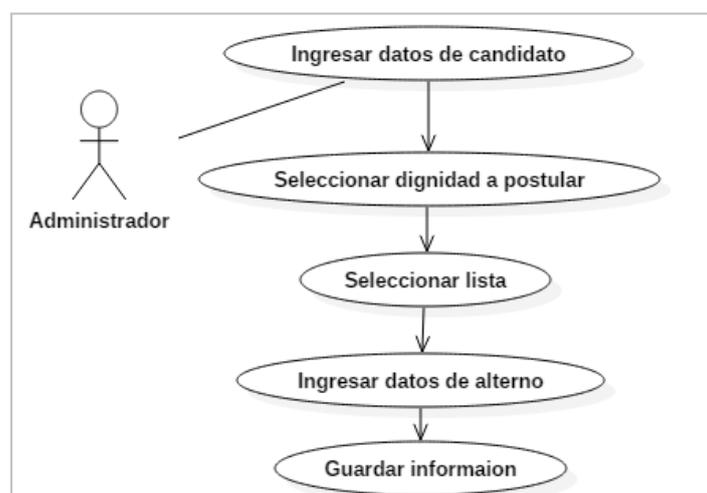


Ilustración 7 Inscribir candidatos
Desarrollado por: Amelia Alexandra Estrada Albiño

Cargar registro electoral

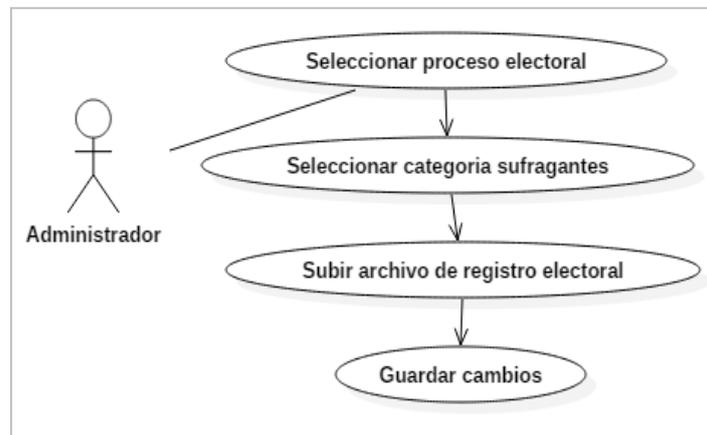


Ilustración 8 Cargar registro electoral
 Desarrollado por: Amelia Alexandra Estrada Albiño

Crear padrón electoral

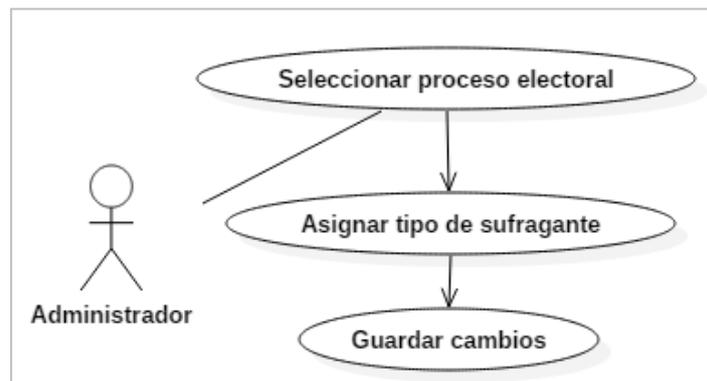


Ilustración 9 Crear padrón electoral
 Desarrollado por: Amelia Alexandra Estrada Albiño

Asignar juntas receptoras de voto

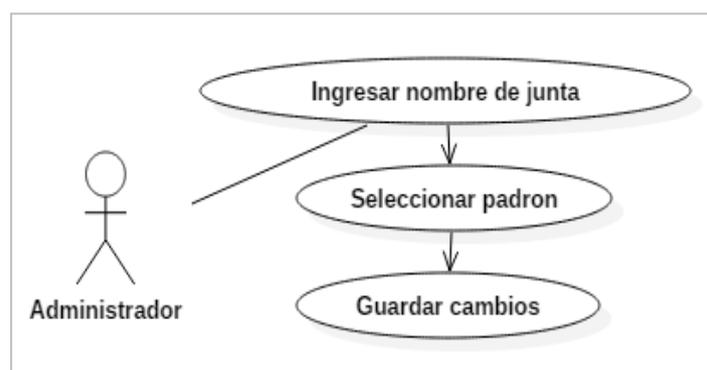


Ilustración 10 Asignar juntas receptoras de voto
 Desarrollado por: Amelia Alexandra Estrada Albiño

Designar integrantes de JVR

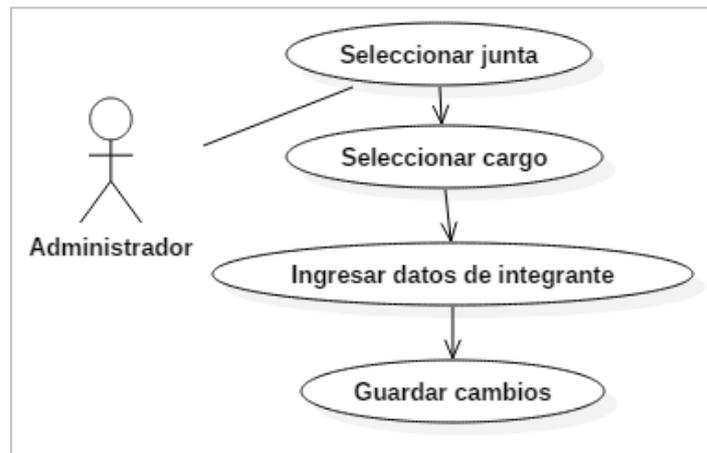


Ilustración 11 Designar integrantes de JVR
Desarrollado por: Amelia Alexandra Estrada Albiño

Proclamar resultados generales

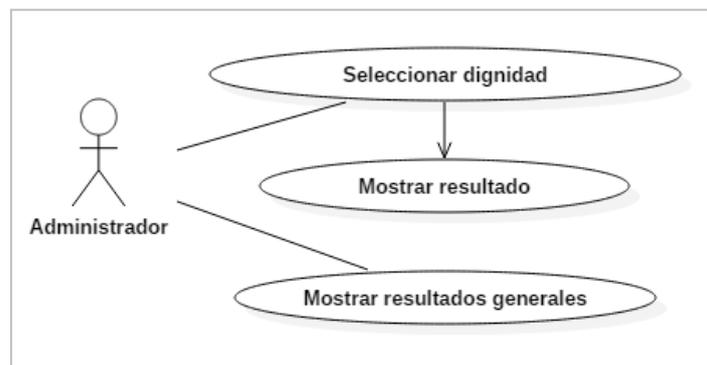


Ilustración 12 Proclamar resultados generales
Desarrollado por: Amelia Alexandra Estrada Albiño

Verificar sufragante en padrón electoral

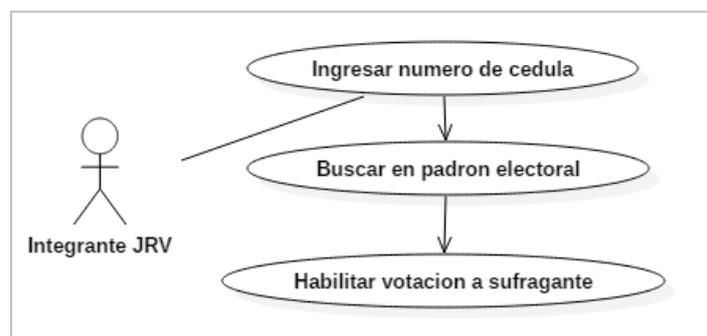


Ilustración 13 Verificar sufragante en padrón electoral
Desarrollado por: Amelia Alexandra Estrada Albiño

Emitir resultados parciales

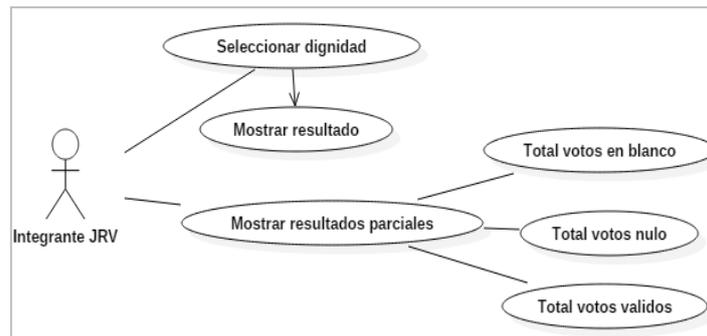


Ilustración 14 Emitir resultados parciales
Desarrollado por: Amelia Alexandra Estrada Albiño

Realizar votación

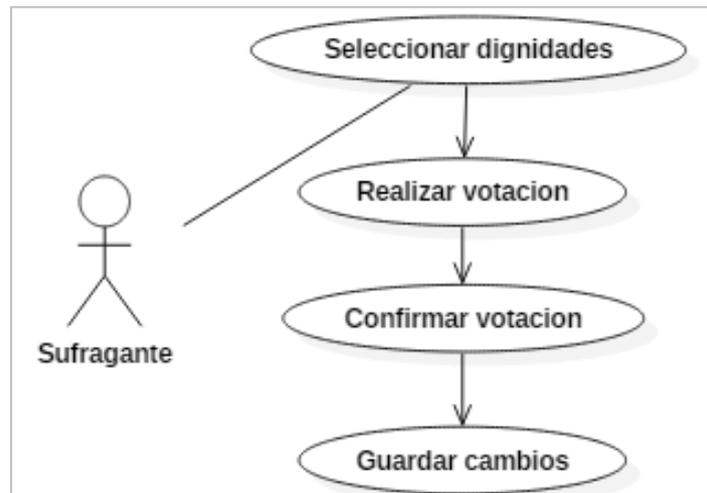


Ilustración 15 Realizar votación
Desarrollado por: Amelia Alexandra Estrada Albiño

Diagrama de Actividad
(Perfil Sufragante)

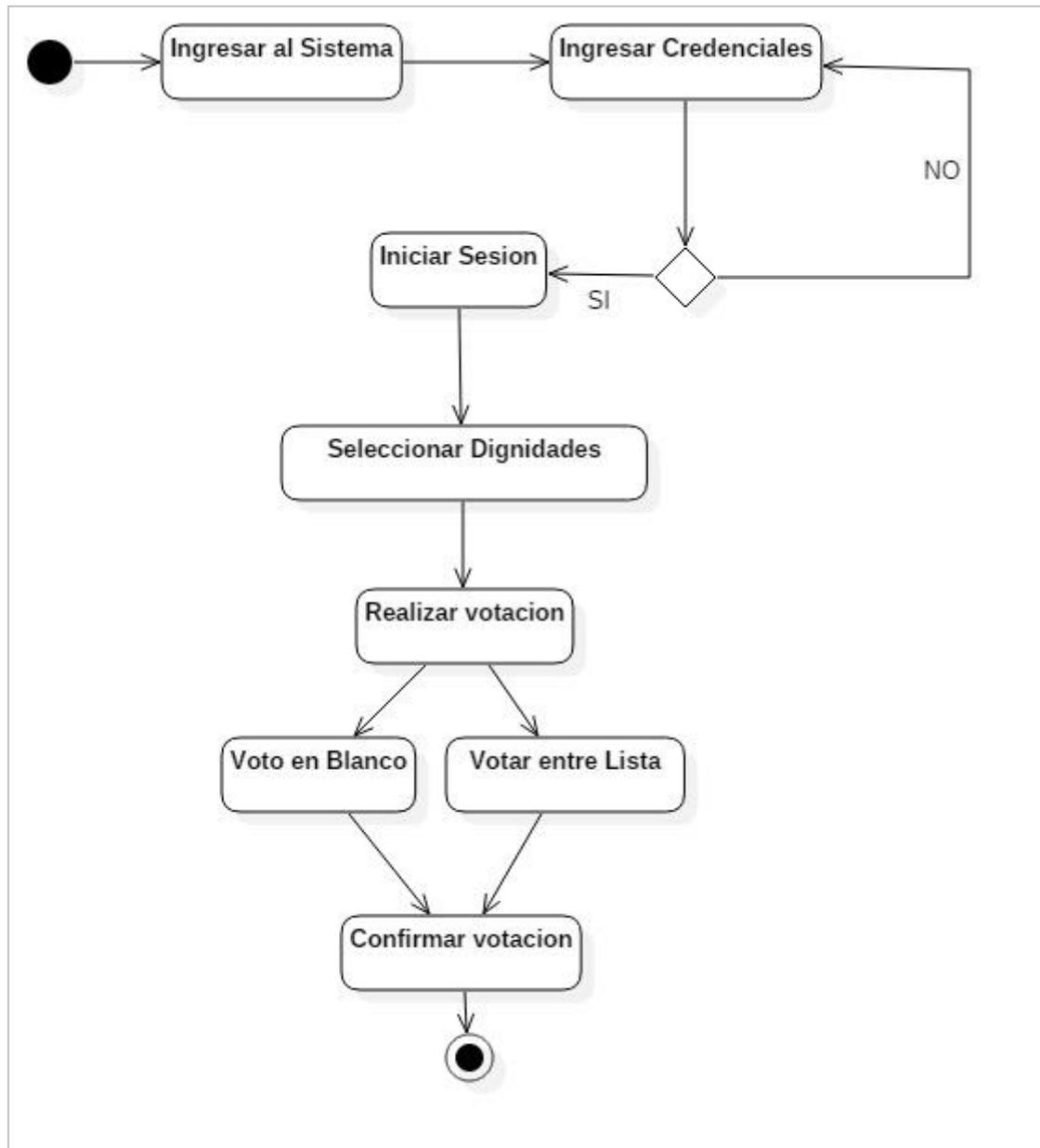


Ilustración 16 Diagrama de Actividad Sufragante
Desarrollado por: Amelia Alexandra Estrada Albiño

Diagrama de Actividad
(Perfil Integrante de Junta Receptora de Voto)

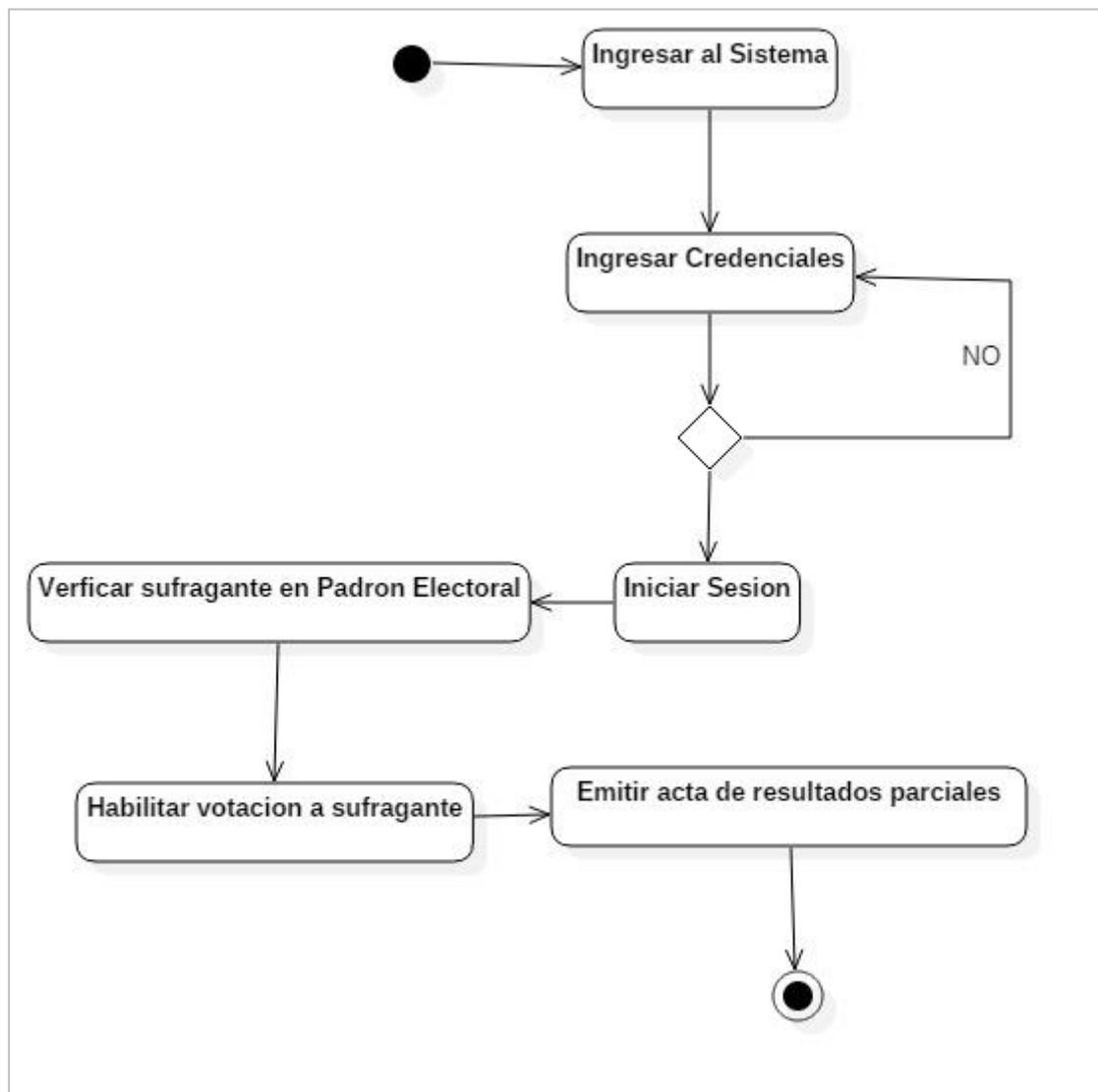


Ilustración 17 Diagrama de Actividad Integrante Junta Receptora de Voto
Desarrollado por: Amelia Alexandra Estrada Albiño

Diagrama de Actividad
(Perfil Administrador)

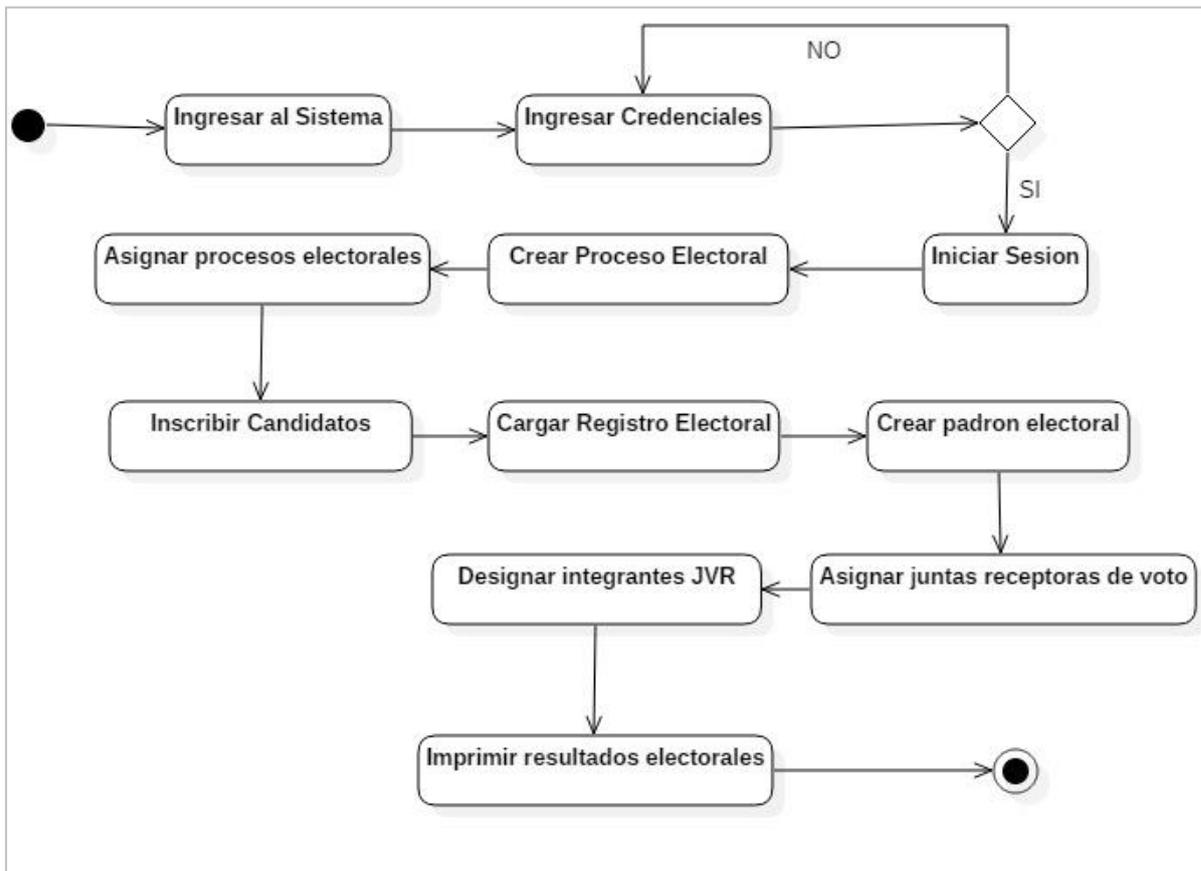


Ilustración 18 Diagrama de Actividad Administrador
Desarrollado por: Amelia Alexandra Estrada Albiño

Diagrama de Secuencia

(Perfil Sufragante)

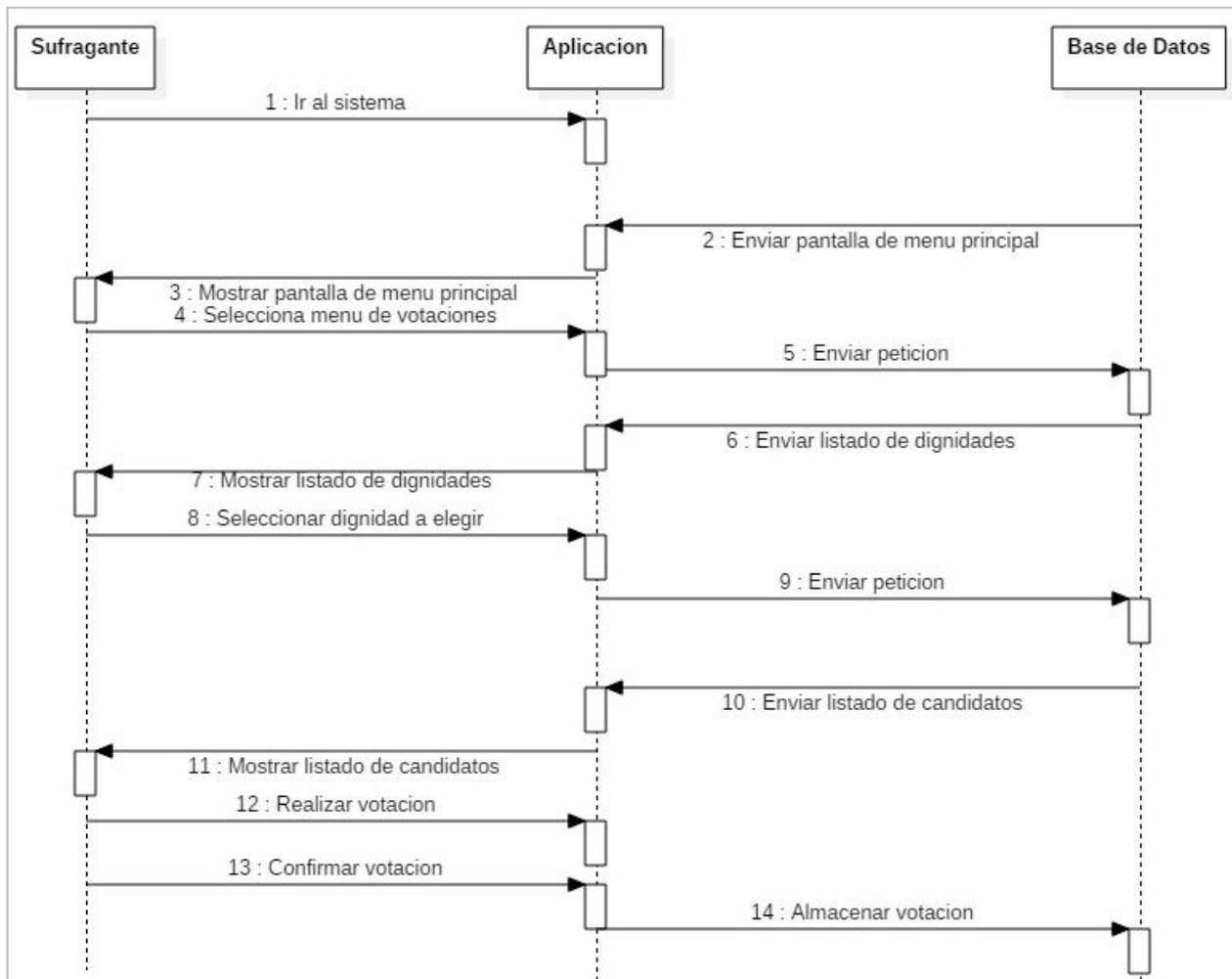


Ilustración 19 Diagrama de Secuencia Sufragante
Desarrollado por: Amelia Alexandra Estrada Albiño

Diagrama de Secuencia
(Perfil Integrante de Junta Receptora de Voto)

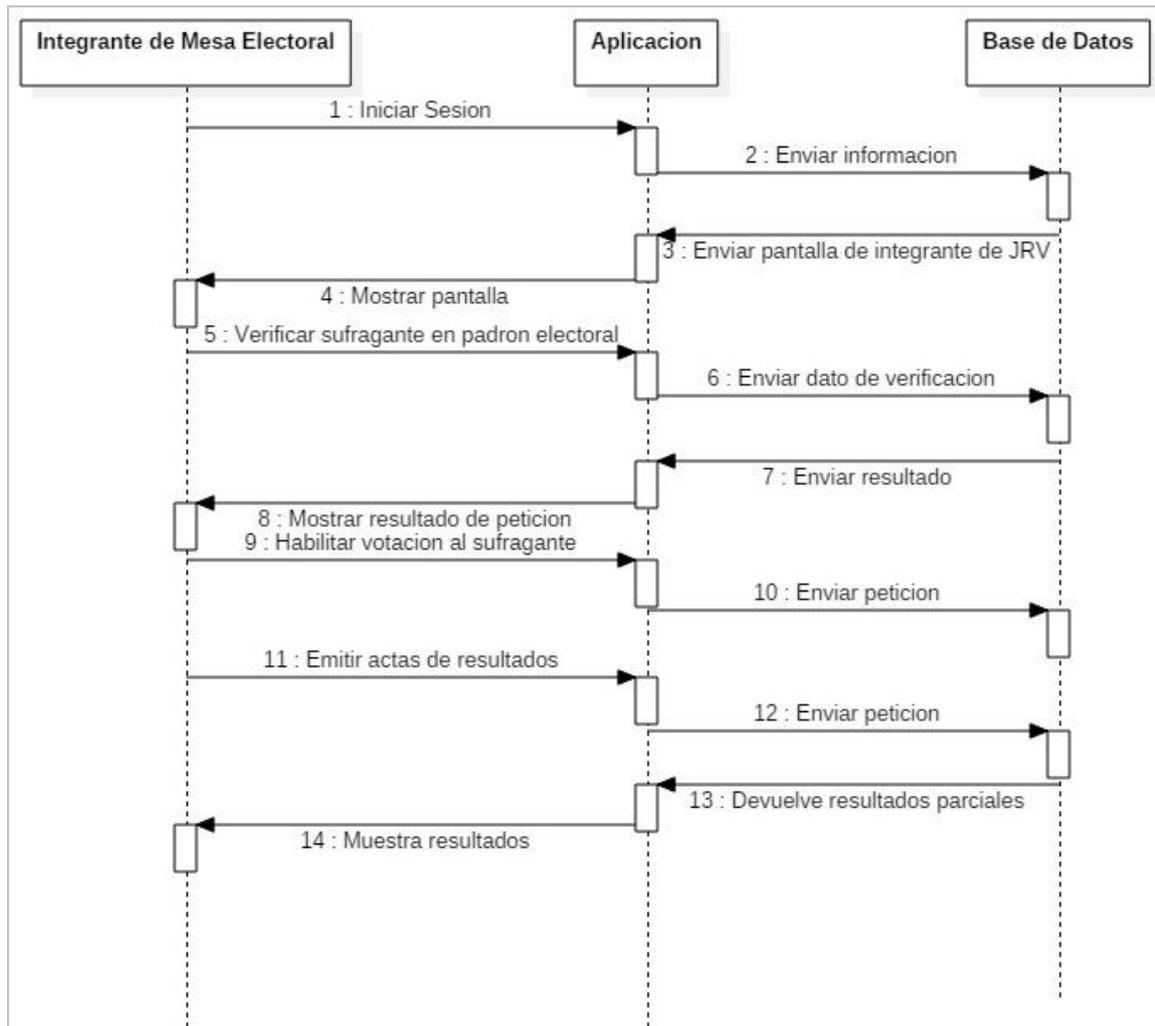


Ilustración 20 Diagrama de Secuencia Integrante Junta Receptora de Voto
Desarrollado por: Amelia Alexandra Estrada Albiño

Diagrama de Secuencia

(Perfil Administrador)

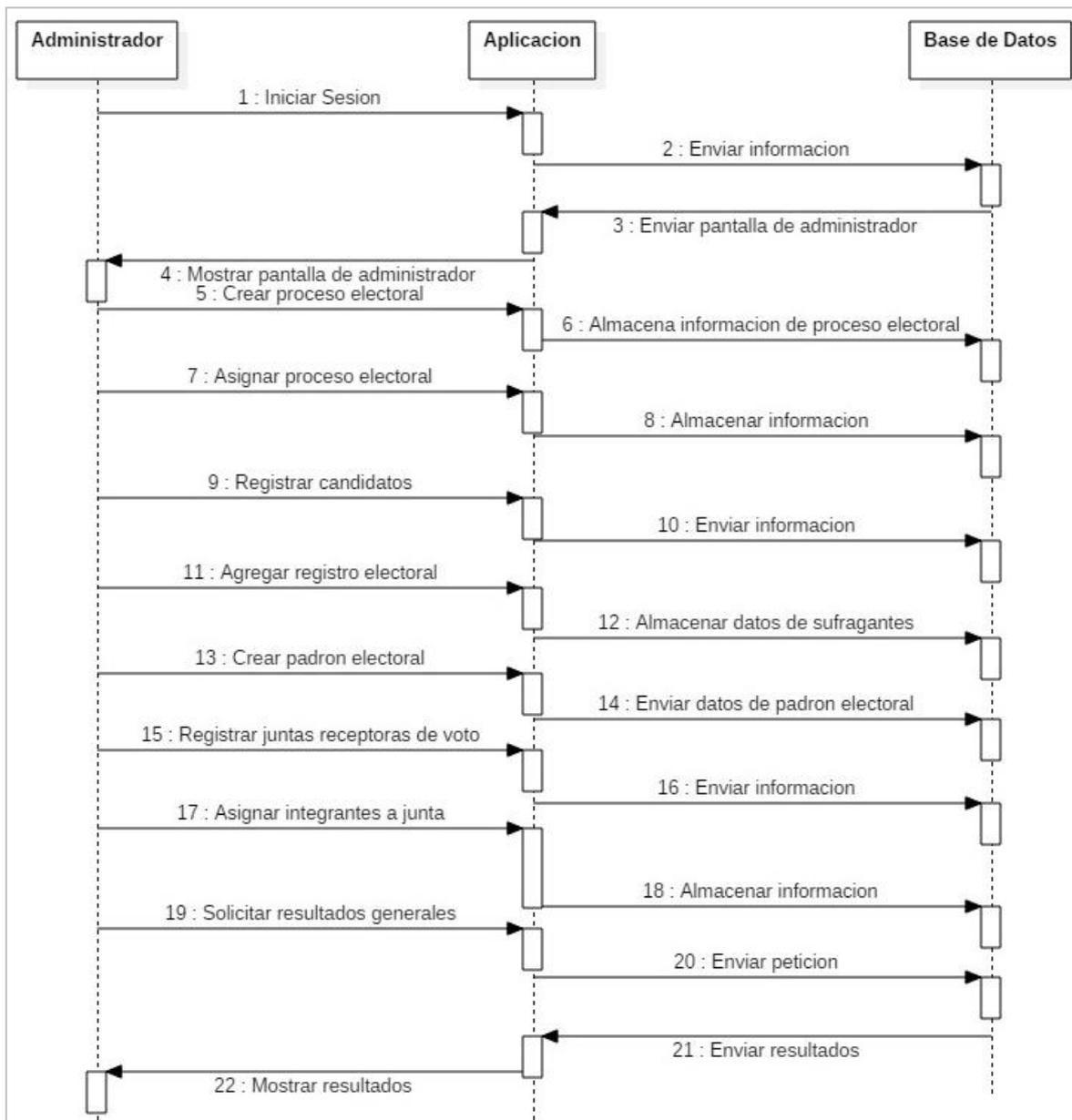


Ilustración 21 Diagrama de Secuencia Administrador
Desarrollado por: Amelia Alexandra Estrada Albiño

Diagrama de Clases

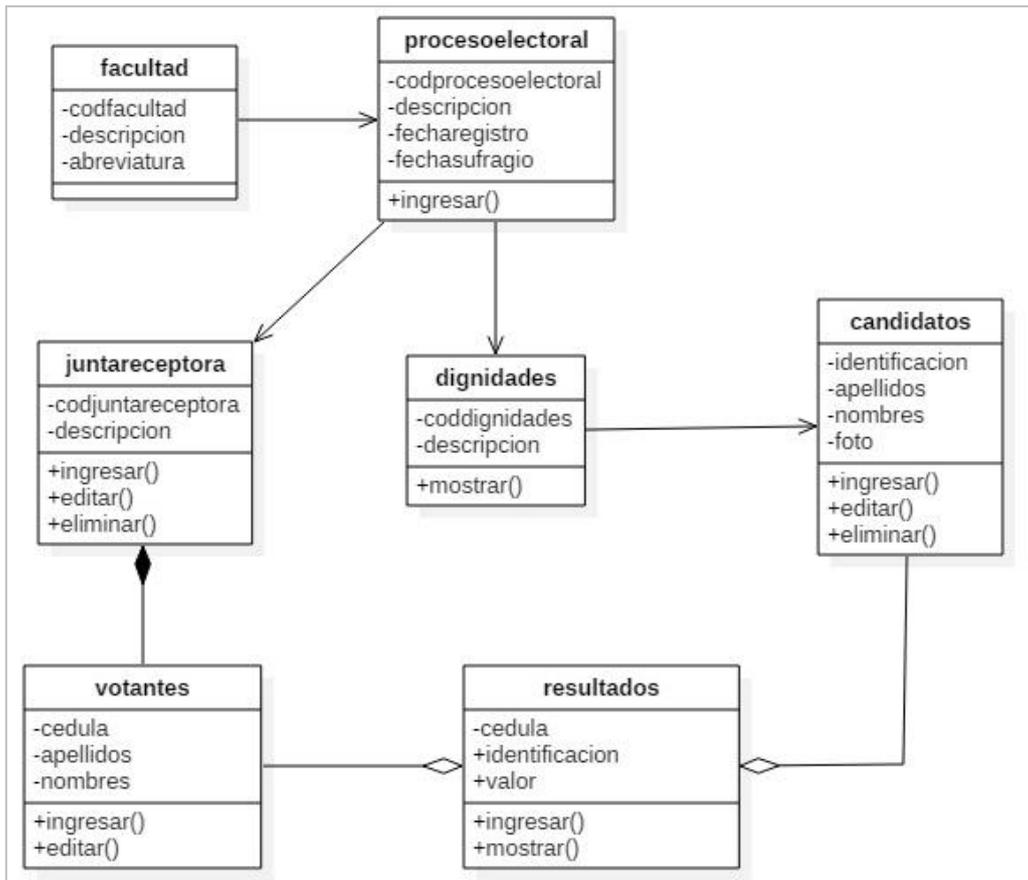


Ilustración 22 Diagrama de Clases
 Desarrollado por: Amelia Alexandra Estrada Albiño

Modelo Conceptual

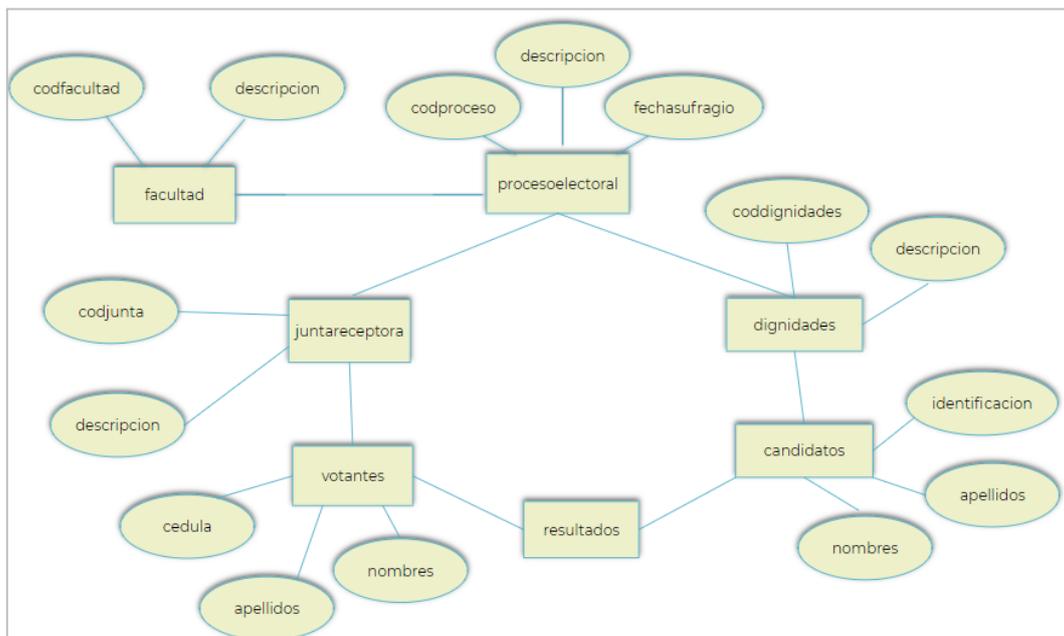


Ilustración 23 Modelo Conceptual
 Desarrollado por: Amelia Alexandra Estrada Albiño

Modelo Lógico

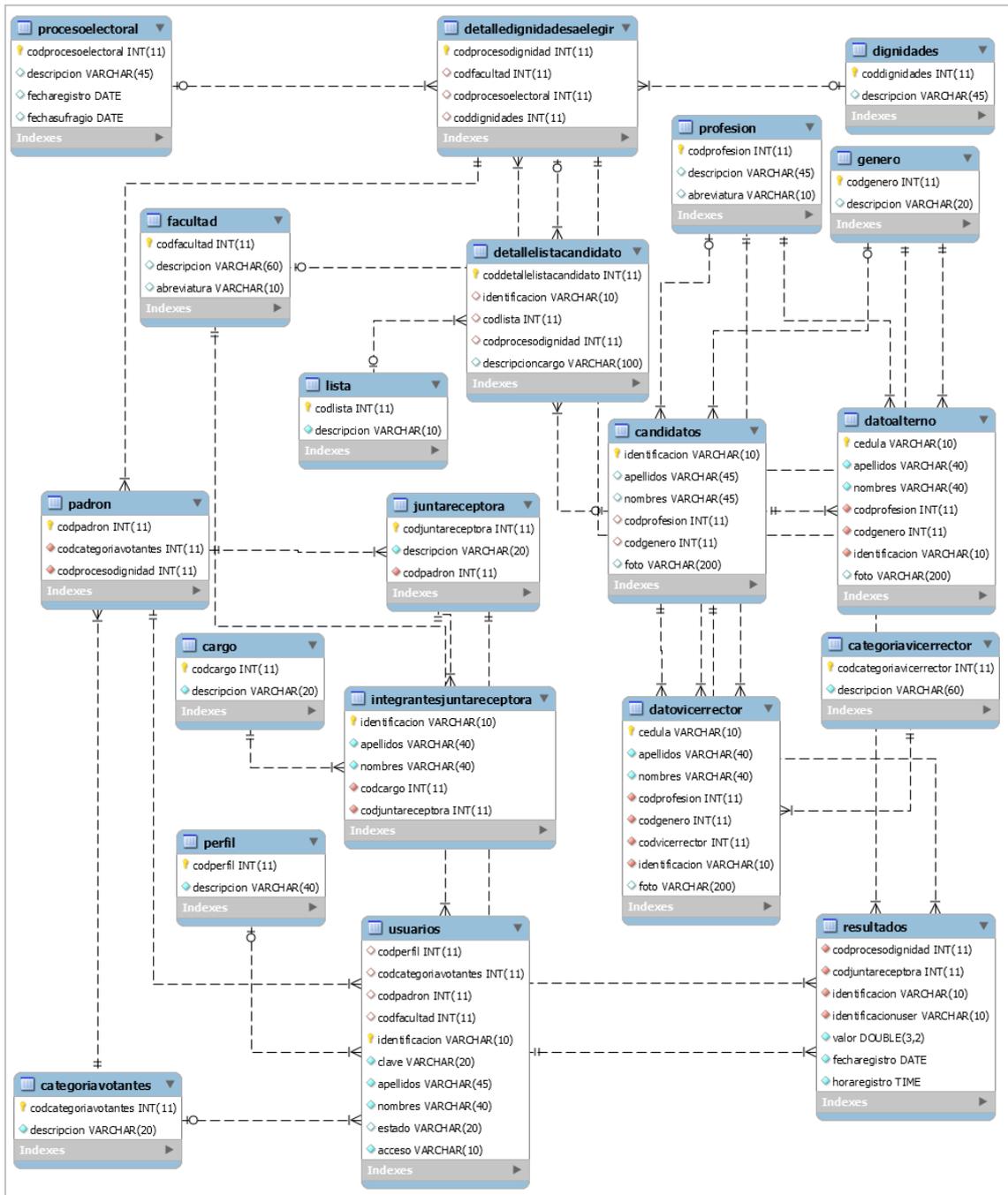


Ilustración 24 Modelo Lógico
 Desarrollado por: Amelia Alexandra Estrada Albiño

Script Base de Datos

```
CREATE TABLE IF NOT EXISTS `candidatos` (  
  `identificacion` varchar(10) NOT NULL,  
  `apellidos` varchar(45) DEFAULT NULL,  
  `nombres` varchar(45) DEFAULT NULL,  
  `codprofesion` int(11) DEFAULT NULL,  
  `codgenero` int(11) DEFAULT NULL,  
  `foto` varchar(200) DEFAULT NULL,  
  PRIMARY KEY (`identificacion`),  
  KEY `codprofesion_idx` (`codprofesion`),  
  KEY `codgenero_idx` (`codgenero`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `cargo` (  
  `codcargo` int(11) NOT NULL,  
  `descripcion` varchar(20) NOT NULL,  
  PRIMARY KEY (`codcargo`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `categoriavotantes` (  
  `codcategoriavotantes` int(11) NOT NULL,  
  `descripcion` varchar(20) NOT NULL,  
  PRIMARY KEY (`codcategoriavotantes`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```

CREATE TABLE IF NOT EXISTS `datoalterno` (
  `cedula` varchar(10) NOT NULL,
  `apellidos` varchar(40) NOT NULL,
  `nombres` varchar(40) NOT NULL,
  `codprofesion` int(11) NOT NULL,
  `codgenero` int(11) NOT NULL,
  `identificacion` varchar(10) NOT NULL,
  `foto` varchar(200) DEFAULT NULL,
  PRIMARY KEY (`cedula`),
  KEY `identificacion` (`identificacion`),
  KEY `codgenero` (`codgenero`),
  KEY `codprofesion` (`codprofesion`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE IF NOT EXISTS `detalledignidadesaelegir` (
  `codprocesodignidad` int(11) NOT NULL AUTO_INCREMENT,
  `codfacultad` int(11) DEFAULT NULL,
  `codprocesoelectoral` int(11) DEFAULT NULL,
  `coddignidades` int(11) DEFAULT NULL,
  PRIMARY KEY (`codprocesodignidad`),
  KEY `codprocesoelectoral_idx` (`codprocesoelectoral`),
  KEY `coddignidades_idx` (`coddignidades`),
  KEY `codfacultad_idx` (`codfacultad`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

```
CREATE TABLE IF NOT EXISTS `detallelistacandidato` (  
  `coddetallelistacandidato` int(11) NOT NULL AUTO_INCREMENT,  
  `identificacion` varchar(10) DEFAULT NULL,  
  `codlista` int(11) DEFAULT NULL,  
  `codprocesodignidad` int(11) DEFAULT NULL,  
  `descripcioncargo` varchar(50) DEFAULT NULL,  
  PRIMARY KEY (`coddetallelistacandidato`),  
  KEY `identificacion_idx` (`identificacion`),  
  KEY `codlista` (`codlista`),  
  KEY `codprocesodignidad` (`codprocesodignidad`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

```
CREATE TABLE IF NOT EXISTS `dignidades` (  
  `coddignidades` int(11) NOT NULL,  
  `descripcion` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`coddignidades`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `facultad` (  
  `codfacultad` int(11) NOT NULL AUTO_INCREMENT,  
  `descripcion` varchar(60) DEFAULT NULL,  
  `abreviatura` varchar(10) DEFAULT NULL,  
  PRIMARY KEY (`codfacultad`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

```
CREATE TABLE IF NOT EXISTS `genero` (  
  `codgenero` int(11) NOT NULL,  
  `descripcion` varchar(20) DEFAULT NULL,  
  PRIMARY KEY (`codgenero`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `integrantesjuntareceptora` (  
  `identificacion` varchar(10) NOT NULL,  
  `apellidos` varchar(40) NOT NULL,  
  `nombres` varchar(40) NOT NULL,  
  `codcargo` int(11) NOT NULL,  
  `codjuntareceptora` int(11) NOT NULL,  
  PRIMARY KEY (`identificacion`),  
  KEY `codcargo` (`codcargo`),  
  KEY `codjuntareceptora` (`codjuntareceptora`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `juntareceptora` (  
  `codjuntareceptora` int(11) NOT NULL AUTO_INCREMENT,  
  `descripcion` varchar(20) NOT NULL,  
  `codpadron` int(11) NOT NULL,  
  PRIMARY KEY (`codjuntareceptora`),  
  KEY `codpadron` (`codpadron`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

```
CREATE TABLE IF NOT EXISTS `lista` (  
  `codlista` int(11) NOT NULL AUTO_INCREMENT,  
  `descripcion` varchar(10) NOT NULL,  
  PRIMARY KEY (`codlista`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
```

```
CREATE TABLE IF NOT EXISTS `padron` (  
  `codpadron` int(11) NOT NULL AUTO_INCREMENT,  
  `codcategoriavotantes` int(11) NOT NULL,  
  `codprocesodignidad` int(11) NOT NULL,  
  PRIMARY KEY (`codpadron`),  
  KEY `codcategoriavotantes` (`codcategoriavotantes`),  
  KEY `codprocesodignidad` (`codprocesodignidad`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

```
CREATE TABLE IF NOT EXISTS `perfil` (  
  `codperfil` int(11) NOT NULL,  
  `descripcion` varchar(40) NOT NULL,  
  PRIMARY KEY (`codperfil`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `procesoelectoral` (  
  `codprocesoelectoral` int(11) NOT NULL AUTO_INCREMENT,  
  `descripcion` varchar(45) DEFAULT NULL,  
  `fecharegistro` date DEFAULT NULL,  
  `fechasufragio` date DEFAULT NULL,
```

```
PRIMARY KEY (`codprocesoelectoral`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

```
CREATE TABLE IF NOT EXISTS `profesion` (  
  `codprofesion` int(11) NOT NULL,  
  `descripcion` varchar(45) DEFAULT NULL,  
  `abreviatura` varchar(10) DEFAULT NULL,  
  PRIMARY KEY (`codprofesion`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `resultados` (  
  `codjuntareceptora` int(11) NOT NULL,  
  `identificacion` varchar(10) NOT NULL,  
  `identificacionuser` varchar(10) NOT NULL,  
  `valor` int(11) NOT NULL,  
  `fecharegistro` date NOT NULL,  
  `horaregistro` time NOT NULL,  
  KEY `identificacion` (`identificacion`),  
  KEY `identificacionuser` (`identificacionuser`),  
  KEY `codjuntareceptora` (`codjuntareceptora`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `usuarios` (  
  `codperfil` int(11) DEFAULT NULL,  
  `codcategoriavotantes` int(11) DEFAULT NULL,  
  `codpadron` int(11) DEFAULT NULL,  
  `identificacion` varchar(10) NOT NULL,  
  `clave` varchar(20) NOT NULL,  
  `apellidos` varchar(45) NOT NULL,  
  `nombres` varchar(40) NOT NULL,  
  `estado` varchar(20) DEFAULT NULL,  
  `acceso` varchar(10) NOT NULL,  
  PRIMARY KEY (`identificacion`),  
  KEY `codperfil` (`codperfil`),  
  KEY `codcategoriavotantes` (`codcategoriavotantes`),  
  KEY `padron` (`codpadron`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
ALTER TABLE `candidatos`  
  ADD CONSTRAINT `candidatos_ibfk_3` FOREIGN KEY (`codprofesion`)  
  REFERENCES `profesion` (`codprofesion`) ON DELETE CASCADE ON UPDATE  
  CASCADE,  
  ADD CONSTRAINT `candidatos_ibfk_4` FOREIGN KEY (`codgenero`) REFERENCES  
  `genero` (`codgenero`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `datoalterno`  
  ADD CONSTRAINT `datoalterno_ibfk_4` FOREIGN KEY (`codprofesion`)  
REFERENCES `profesion` (`codprofesion`) ON DELETE CASCADE ON UPDATE  
CASCADE,  
  ADD CONSTRAINT `datoalterno_ibfk_5` FOREIGN KEY (`codgenero`) REFERENCES  
`genero` (`codgenero`) ON DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `datoalterno_ibfk_6` FOREIGN KEY (`identificacion`)  
REFERENCES `candidatos` (`identificacion`) ON DELETE CASCADE ON UPDATE  
CASCADE;  
  
ALTER TABLE `detalledignidadesaelegir`  
  ADD CONSTRAINT `detalledignidadesaelegir_ibfk_1` FOREIGN KEY (`codfacultad`)  
REFERENCES `facultad` (`codfacultad`) ON DELETE CASCADE ON UPDATE  
CASCADE,  
  ADD CONSTRAINT `detalledignidadesaelegir_ibfk_2` FOREIGN KEY  
(`codprocesoelectoral`) REFERENCES `procesoelectoral` (`codprocesoelectoral`) ON  
DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `detalledignidadesaelegir_ibfk_3` FOREIGN KEY (`coddignidades`)  
REFERENCES `dignidades` (`coddignidades`) ON DELETE CASCADE ON UPDATE  
CASCADE;  
  
ALTER TABLE `detallelistacandidato`  
  ADD CONSTRAINT `detallelistacandidato_ibfk_1` FOREIGN KEY (`identificacion`)  
REFERENCES `candidatos` (`identificacion`) ON DELETE CASCADE ON UPDATE  
CASCADE,  
  ADD CONSTRAINT `detallelistacandidato_ibfk_2` FOREIGN KEY (`codlista`)  
REFERENCES `lista` (`codlista`) ON DELETE CASCADE ON UPDATE CASCADE,
```

```
ADD CONSTRAINT `detallelistacandidato_ibfk_4` FOREIGN KEY
(`codprocesodignidad`) REFERENCES `detalledignidadesaelegir` (`codprocesodignidad`)
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `integrantesjuntareceptora`
ADD CONSTRAINT `integrantesjuntareceptora_ibfk_1` FOREIGN KEY (`codcargo`)
REFERENCES `cargo` (`codcargo`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `integrantesjuntareceptora_ibfk_2` FOREIGN KEY
(`codjuntareceptora`) REFERENCES `juntareceptora` (`codjuntareceptora`) ON DELETE
CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `juntareceptora`
ADD CONSTRAINT `juntareceptora_ibfk_1` FOREIGN KEY (`codpadron`)
REFERENCES `padron` (`codpadron`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `padron`
ADD CONSTRAINT `padron_ibfk_2` FOREIGN KEY (`codcategoriavotantes`)
REFERENCES `categoriavotantes` (`codcategoriavotantes`) ON DELETE CASCADE ON
UPDATE CASCADE,
ADD CONSTRAINT `padron_ibfk_3` FOREIGN KEY (`codprocesodignidad`)
REFERENCES `detalledignidadesaelegir` (`codprocesodignidad`) ON DELETE CASCADE
ON UPDATE CASCADE;
```

```
ALTER TABLE `resultados`  
  
  ADD CONSTRAINT `resultados_ibfk_3` FOREIGN KEY (`identificacionuser`)  
REFERENCES `usuarios` (`identificacion`) ON DELETE CASCADE ON UPDATE  
CASCADE,  
  
  ADD CONSTRAINT `resultados_ibfk_4` FOREIGN KEY (`codjuntareceptora`)  
REFERENCES `juntareceptora` (`codjuntareceptora`) ON DELETE CASCADE ON  
UPDATE CASCADE,  
  
  ADD CONSTRAINT `resultados_ibfk_4` FOREIGN KEY (`identificacion`)  
REFERENCES `candidatos` (`identificacion`) ON DELETE CASCADE ON UPDATE  
CASCADE;
```

```
ALTER TABLE `usuarios`  
  
  ADD CONSTRAINT `usuarios_ibfk_1` FOREIGN KEY (`codperfil`) REFERENCES  
`perfil` (`codperfil`) ON DELETE CASCADE ON UPDATE CASCADE,  
  
  ADD CONSTRAINT `usuarios_ibfk_2` FOREIGN KEY (`codcategoriavotantes`)  
REFERENCES `categoriavotantes` (`codcategoriavotantes`) ON DELETE CASCADE ON  
UPDATE CASCADE,  
  
  ADD CONSTRAINT `usuarios_ibfk_3` FOREIGN KEY (`codpadron`) REFERENCES  
`padron` (`codpadron`) ON DELETE CASCADE ON UPDATE CASCADE;
```

2.4.3. Fase de Desarrollo

Se describen las tareas que realiza la aplicación web haciendo referencia en cada una de sus interfaces.



Ilustración 25 Interfaz principal
Desarrollado por: Amelia Alexandra Estrada Albiño

Entre las tecnologías web que se ha utilizado para el desarrollo del prototipo tenemos las siguientes:

MySQL

Mediante phpMyAdmin se puede trabajar en el diseño de la base de datos, ya que ofrece la posibilidad de trabajar de manera gráfica.

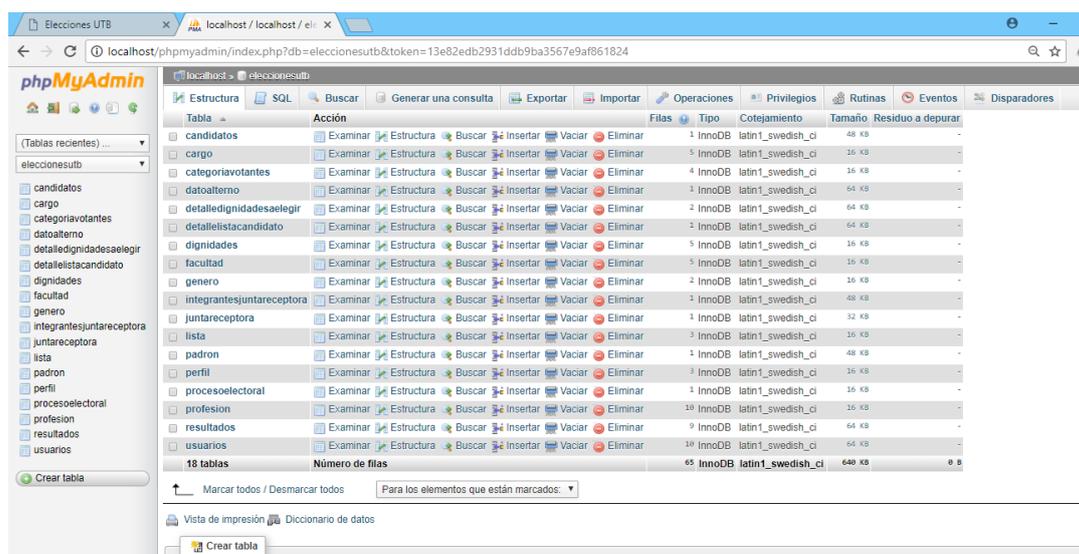


Ilustración 26 Interfaz de MySQL

NetBeans IDE

Utilizado por ser un entorno de desarrollo interactivo que permite adaptarse fácilmente y así realizar la aplicación web, en este caso haciendo uso del lenguaje PHP.

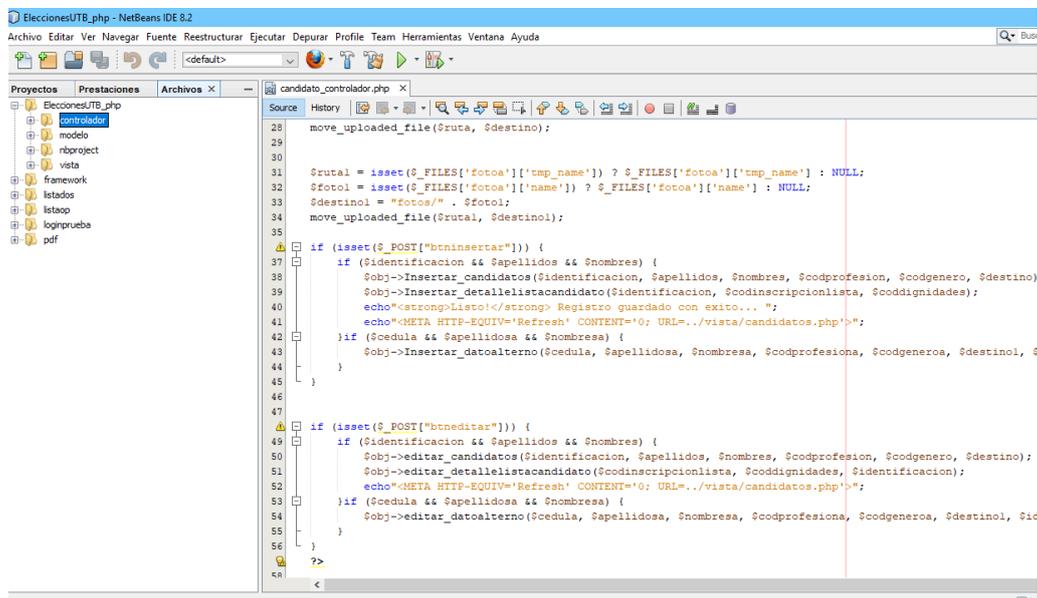


Ilustración 27 Interfaz de NetBeans IDE

2.5. Ejecución y/o ensamblaje del prototipo

Acceso a la aplicación web

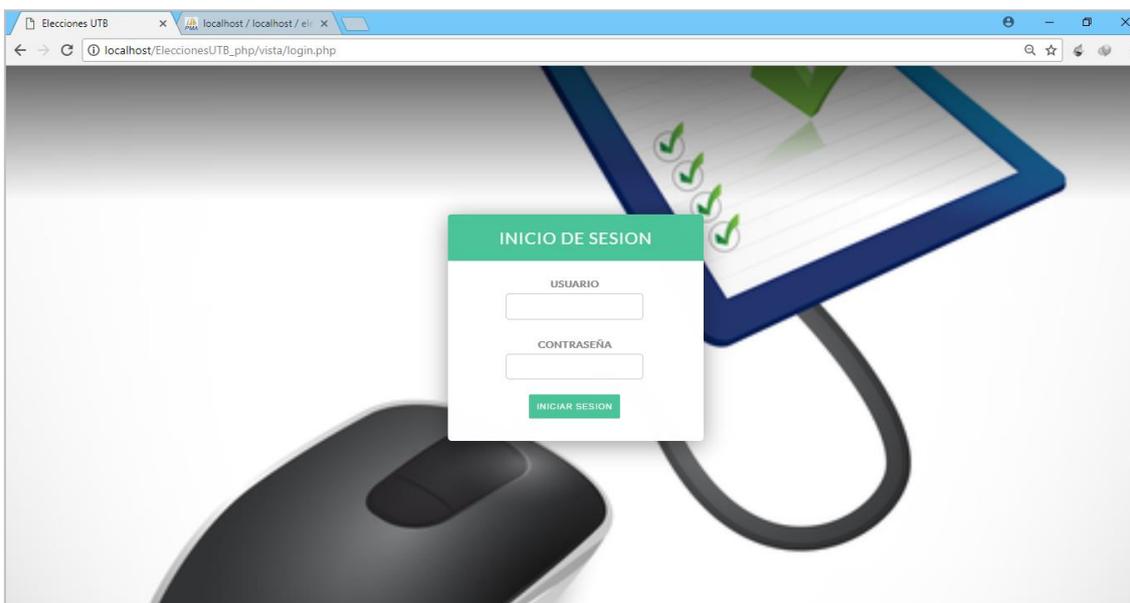


Ilustración 28 Interfaz Acceso a la aplicación web
Desarrollado por: Amelia Alexandra Estrada Albiño

Perfil Administrador

Encargado de gestionar la información correspondiente a los procesos electorales, entre sus principales funciones está administrar el siguiente menú de interfaces:

- Proceso electoral.
- Candidatos
- Registro electoral
- Juntas receptoras de voto
- Resultados Generales

Proceso electoral

Desde aquella pantalla puede gestionar la inscripción de los procesos electorales y la asignación de estos mismos procesos, de acuerdo a las dignidades que se vaya a elegir en dicho periodo ya preestablecido.



Ilustración 29 Interfaz Proceso electoral
Desarrollado por: Amelia Alexandra Estrada Albiño

Candidatos

En la siguiente interfaz puede realizar la inscripción de candidatos, y a su vez puede realizar acciones de edición o eliminación de algún registro, presentado en el listado.

FOTO	CEDULA	NOMBRES	CARGO	LISTA	ACCIONES
	1250093937	LCDA KERLY JUDITH GAIBOR MORALES Alterno: ANGEL AGUSTIN MORALES COBOS	REPRESENTANTE DOCENTES- FCS	LISTA A	
	1250093939	ING ANA LUCIA MORALES MAYORGA Alterno: ANGEL MANUEL LOPEZ ROMERO	REPRESENTANTE DOCENTES- FAFI	LISTA A	
	1250093934	MSC JUAN CARLOS SOLIZ LOPEZ Alterno: ANGELA ANDREA CONTRERAS PLUAS	REPRESENTANTE DOCENTES- FCJSE	LISTA A	
	1250093930	LCDA GLENDA ANAHI AGUIAR MORANTE Alterno: XAVIER DAVID CONTRERAS PINO	REPRESENTANTE DOCENTES- FCS	LISTA B	

Ilustración 30 Interfaz Candidatos
 Desarrollado por: Amelia Alexandra Estrada Albiño

Registro electoral

En esta sección puede cargar el registro electoral que debe estar en un archivo con extensión csv, y posteriormente cargar en el sistema luego este se almacena en la base de datos correspondiente, además puede gestionar el proceso de creación del padrón electoral.



Ilustración 31 Interfaz Registro electoral
 Desarrollado por: Amelia Alexandra Estrada Albiño

Junta receptora del voto

Crearé las juntas receptoras del voto necesarias una para cada padrón existente en el sistema, a su vez también puede asignar los integrantes de dicha junta.

ELECCIONES UTB

CERRAR SESION

Usuario: HUGO JAVIER GUERRERO TORRES
Perfil: ADMINISTRADOR

ACRECAR JUNTAS ACRECAR INTEGRANTES

CEDULA	NOMBRES	CARGO	JUNTA	ACCIONES
1200599650	GONZÁLEZ MAYORGA VICENTA RICARDINA	PRESIDENTE(A)	MESA ELECTORAL 1	
1204278517	ESCOBAR MAYORGA DIANA CAROLINA	PRIMER VOCAL	MESA ELECTORAL 1	
1706747217	VELA RIERA PABLO ALBERTO	SECRETARIO(A)	MESA ELECTORAL 1	

Realizado por:
Alexandra Estrada Albiño

Ilustración 32 Interfaz Junta receptora del voto
Desarrollado por: Amelia Alexandra Estrada Albiño

Resultados Generales

En esta interfaz puede visualizar los resultados generales del proceso electoral en cuestión y puede imprimir el respectivo archivo.

CERRAR SESION

Usuario: HUGO JAVIER GUERRERO TORRES
Perfil: ADMINISTRADOR

RESULTADOS

Dignidad a elegir: REPRESENTANTE DOCENTES-FAFI

FOTO	NOMBRES	LISTA	TOTAL VOTOS
	ING ANA LUCIA MORALES MAYORGA	LISTA A	3
	ING JULIO CESAR GOMEZ HOLGUIN	LISTA B	1

Votos Blancos: 1 Votos Nulos: 1 Votos Validos: 4

IMPRIMIR

Ilustración 33 Interfaz Resultados Generales
Desarrollado por: Amelia Alexandra Estrada Albiño



FOTO	NOMBRES	LISTA	TOTAL VOTOS
	LCDA. GAIBOR MORALES KERLY JUDITH	LISTA A	3
	LCDA. AGUIAR MORANTE GLENDA ANAHI	LISTA B	1

Ilustración 34 Reporte de resultados generales
Desarrollado por: Amelia Alexandra Estrada Albiño

Perfil Integrante de junta receptora del voto

Es el encargado de gestionar la tarea de verificación del sufragante en el sistema y emitir resultados parciales de su respectiva junta, por lo tanto administrara las siguientes interfaces:

- Padrón electoral.
- Resultados.

Padrón electoral

Una vez ingresado al sistema bajo este perfil el presidente de la junta será el único integrante de la junta receptora del voto que puede tener acceso a la aplicación web, por tanto el será quien se encargue de verificar la existencia del sufragante en el sistema y posterior a eso habilitar para que este pueda cumplir con su respectiva votación.



Ilustración 35 Interfaz Padrón electoral
 Desarrollado por: Amelia Alexandra Estrada Albiño

Al final del proceso electoral puede imprimir un reporte de quienes han cumplido con la votación y quiénes no.



Ilustración 36 Reporte de padrón electoral
 Desarrollado por: Amelia Alexandra Estrada Albiño

Resultados

En esta sección se muestra los resultados parciales, es decir, resultados pertenecientes a cada junta receptora del voto.



Ilustración 37 Interfaz Resultados
Desarrollado por: Amelia Alexandra Estrada Albiño

Además podemos imprimir los resultados por categoría y nos presentara de la siguiente forma.



Ilustración 38 Reporte de resultados parciales
Desarrollado por: Amelia Alexandra Estrada Albiño

Perfil Sufragante

Una vez que se ha presentado en la junta puede acceder al sistema y proceder con su función que es realizar la votación. Gestionará la siguiente interfaz:

- Votación en línea.

Votación en línea

El sufragante seleccionara la dignidad a elegir y procederá a la votación, puede directamente votar en blanco o elegir a uno de los candidatos de acuerdo a su elección.

VOTACION EN LINEA

Usuario: GEOVANNY VEGA VILLACIS
Perfil: SUFRAGANTE

Cedula: 0603005497 Junta Nro.: J-0001-1 Padron: PE-0001-2

Dignidad a elegir: REPRESENTANTE DOCENTES-FAFI

LISTADO DE CANDIDATOS

FOTO	NOMBRES	LISTA	VOTAR
	ING ANA LUCIA MORALES MAYORGA Alterno: ANGEL MANUEL LOPEZ ROMERO	LISTA A	<input type="button" value="VOTAR"/>
	ING JULIO CESAR GOMEZ HOLGUIN Alterno: ANGELA MARIA FERNANDEZ SALINAS	LISTA B	<input type="button" value="VOTAR"/>

VOTAR EN BLANCO GUARDAR

Ilustración 39 Interfaz Votación en línea
Desarrollado por: Amelia Alexandra Estrada Albiño

3. CAPITULO III. EVALUACION DEL PROTOTIPO

3.1. Evaluación Técnica

La Universidad Técnica de Babahoyo cuenta con equipamiento necesario, al contar con diversos laboratorios de computación, que poseen equipos que cumplen con las características necesarias para la ejecución de este prototipo.

Además, la institución en su respectivo departamento de sistemas cuenta con servidores capaces de alojar la información necesaria para el correcto funcionamiento de la aplicación web.

El prototipo ha sido diseñado de tal manera que este alojado en un servidor web para que distintos usuarios puedan acceder mediante una url en un tiempo que se considera prudencial para su respectiva votación o el cumplimiento de sus demás funciones según sea el usuario. Para ello, la forma más adecuada de asimilar esta funcionalidad se ejecuta la aplicación web en un servidor local. Y se denota que cumple con cada una de sus funcionalidades para la realización del proceso electoral que se desee.

3.2. Evaluación Económica

La aplicación web tiene la finalidad de reducir costos asociados a la puesta en marcha de los procesos electorales, y esto se cumple en un alto grado, de tal forma para el alojamiento de la información se puede utilizar los mismos servidores de la institución, y para la utilización del prototipo se necesita equipos de computación los cuales la institución si cuenta con ellos en sus respectivos laboratorios.

En el ámbito del proceso de elecciones, reduce costos de materiales, porque no es necesario tener impreso los padrones electorales mucho menos las papeletas para las dignidades a ser elegidas, esta información se puede alojar en el sistema, permitiendo así manejar dicha información digital de tal manera se lleva a cabo el cumplimiento de dichos procesos.

3.3. Evaluación de Funcionalidad

Para construir esta aplicación web se ha basado estrechamente en la forma que se ejecutan los procesos electorales en la institución, lo cual ha permitido la obtención de un prototipo capaz de realizar las tareas necesarias acorde a los procesos en cuestión.

Realizando una comparación en el tiempo de respuesta que toma a un sufragante votar se encuentra lo siguiente: la votación de manera tradicional toma su tiempo de tres a cinco minutos según el listado de dignidades a elegir. Mientras tanto la aplicación web reduce ese tiempo a un minuto con treinta segundos. Se considera este tiempo desde que el sufragante se ha presentado en la junta receptora del voto, hasta concluir con la votación del mismo.

A continuación se presenta una tabla comparativa de la realización de votaciones mediante la forma tradicional y la aplicación web.

SUFragANTE	VOTACION TIEMPO RESPUESTA		PORCENTAJE DE REDUCCION DE TIEMPO
	TRADICIONAL	APLICACIÓN WEB	
Sufragante 1	5 minutos	1 min 10 segundos	77%
Sufragante 2	3 minutos	50 segundos	73%
Sufragante 3	4 minutos	1 minuto	75%

4. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

La institución cuenta con equipamiento informático, que cumplen con los requerimientos previamente analizados para el correcto funcionamiento de la aplicación web.

En la institución existe personal capacitado para manipular equipos informáticos lo cual facilitara a la hora de relacionarse con el prototipo.

Los servidores de la institución, albergados en el respectivo departamento de sistemas soportan la carga que genera la ejecución de la aplicación web.

Al realizar las pruebas respectivas de funcionalidad se obtiene los tiempos de respuesta en base a la forma tradicional de realizar la votación, y mediante la aplicación web. El tiempo de respuesta se reduce en un 75% en la votación lo cual se considera un grado satisfactorio.

Para salvaguardar la información que transita por la red al momento de la ejecución del prototipo es necesario recurrir a la aplicación de métodos de encriptación.

4.2. Recomendaciones

Para explotar sus funcionalidades de manera adecuada es importante realizar una previa capacitación a las personas que tendrán acceso al sistema, especificando las actividades que puede realizar de acuerdo al perfil que el usuario haya sido asignado.

Previo a la ejecución de los procesos electorales, es necesario optimizar el ancho de banda, y dedicar solo a la ejecución de dicha actividad, esto es por cuestiones de seguridad y por la carga que genera.

Para obtener un eficiente aprovechamiento de la aplicación web, al momento de ser implementada debe estar alojado la información en un servidor propio de la institución.

La aplicación de métodos de encriptación, esta técnica ya está implementada en los servidores de la institución, por lo cual solo se necesitaría habilitar los protocolos correspondientes.

5. BIBLIOGRAFIA

- Alegsa, L. (5 de Julio de 2016). *ALEGSA.com.ar*. Obtenido de Definición de WAMP:
<http://www.alegsa.com.ar/Dic/wamp.php>
- Alvarez, M. (2 de Febrero de 2014). *Desarrolloweb.com*. Obtenido de Teoria de la Programacion Orientada a Objetos: <https://desarrolloweb.com/articulos/que-es-mvc.html>
- Bernabé, A. (2014). *Acceso a datos en aplicaciones web del entorno servidor*. Andalucía: IC Editorial.
- Blanco, E. (2013). Gestion de Datos con MySQL. En E. Blanco, *Fundamentos de Informatica en Entornos Bioinformaticos* (pág. 172). Barcelona: Editorial UOC. Obtenido de <https://ebookcentral.proquest.com/lib/utbabs/reader.action?docID=3214766&query=fundamentos%20de%20informatica%20en%20entornos%20bioinformaticos>
- Casado, C. (2014). Desarrollo de Software. En C. Casado, *Entornos de Desarrollo* (pág. 39). Madrid: RA-MA Editorial. Obtenido de <https://ebookcentral.proquest.com/lib/utbabs/detail.action?docID=3229487&query=entornos%20de%20desarrollo>
- Contreras, M. (2016). *Desarrollo de Aplicaciones Web Multiplataforma*. España: Ministerio de Educación de España. Obtenido de <https://ebookcentral.proquest.com/lib/utbabs/reader.action?docID=4536195&query=Ministerio%20de%20Educaci%C3%B3n%20de%20Espa%C3%B1a>
- Ferrer, M. (2014). *Implantaciones de Aplicaciones Web*. Madrid: RA-MA Editorial.
- Granados, R. (2014). *Desarrollo de aplicaciones web en el entorno servidor* (Primera ed.). Andalucía: IC Editorial.
- Hueso, L. (2014). *Base de Datos*. Madrid : Editorial RA-MA.
- Hueso, L. (2014). Diseño Conceptual y Logico de Base de Datos. En L. Hueso, *Gestion de Base de Datos* (Segunda ed., pág. 84). Madrid: RA-MA Editorial. Obtenido de <https://ebookcentral.proquest.com/lib/utbabs/reader.action?docID=3228876&query=gestion%20de%20base%20de%20datos>
- Ibarra, J. (2013). *Aplicaciones Informaticas de Base de Datos Relacionales*. Madrid: Editorial CEP S.L.
- Lopez, M., Vara, J., & Verde, J. (2014). Generacion Dinamica de Paginas Wew Interactivas. En M. Lopez, J. Vara, & J. Verde, *Desarrollo web en entorno servidor* (págs. 232 - 234). Madrid: RA-MA Editorial. Obtenido de <https://ebookcentral.proquest.com/lib/utbabs/detail.action?docID=3229695&query=desarrollo%20web%20entorno%20servidor>
- Ortiz, M. (25 de Noviembre de 2015). *ML*. Obtenido de Cómo funciona Apache HTTP Server: <http://miguelleonardortiz.com.ar/curso-arquitectura-web/como-funciona-apache-http-server/920>
- Perez, J. (19 de Noviembre de 2017). *Registro y registradores de dominio*. Obtenido de Principales lenguajes de programación web, ventajas y desventajas: <http://www.registrodominiosinternet.es/2013/08/lenguajes-programacion-web-ventajas.html>

- Rouse, M. (3 de Abril de 2017). *Servidor Web*. Obtenido de TechTarget:
<http://searchdatacenter.techtarget.com/es/definicion/Servidor-Web>
- Sanchez, F., Toharia, P., & Raya, L. (2014). *Lenguaje de Marcas y Sistemas de Gestion de la Informacion*. Madrid: Editorial RA-MA.
- Tama, V., & Galán, D. (10 de Septiembre de 2013). *Redaccion El Tiempo*. Obtenido de CNE presenta sistema de voto electrónico: <http://www.eltiempo.com.ec/noticias/cuenca/2/319909/cne-presenta-sistema-de-voto-electronico>
- Universidad Tecnica de Babahoyo. (16 de Diciembre de 2014). *REGLAMENTO INTERNO DE RÉGIMEN ELECTORAL*. Obtenido de Universidad Tecnica de Babahoyo Repositorio Digital:
<http://dspace.utb.edu.ec/>
- Valderrey, P. (2014). *Gestion de Base de Datos*. Madrid: RA-MA, S.A.
- Villada, J. (2015). Conceptos Basicos de Sistemas de Servidores. En J. Villada, *Instalacion y Configuracion del Software de Servidor Web* (págs. 37-38). Andalucia: IC Editorial. Obtenido de
<https://ebookcentral.proquest.com/lib/utbabsp/detail.action?docID=4310544&query=Instalacion%20y%20Configuracion%20del%20Software%20de%20Servidor%20Web>
- Zofio, J. (2014). Lenguajes de marcas, hojas de estilos y scripts. En J. Zofio, *Aplicaciones Web* (pág. 19). Madrid: Macmillan Iberia. S.A. Obtenido de
<https://ebookcentral.proquest.com/lib/utbabsp/reader.action?docID=3217129&query=aplicaciones%20web>