

# UNIVERSIDAD TÉCNICA DEL NORTE



**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS  
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO  
DE INGENIERO EN SISTEMAS COMPUTACIONALES**

**TEMA:**

**“SISTEMA DE VOTO ELECTRÓNICO CON PROTOCOLOS DE CURVAS  
ELÍPTICAS APLICADO EN ELECCIONES POPULARES”**

**AUTOR:**

**DIEGO XAVIER REA ENRÍQUEZ**

**DIRECTOR:**

**Msc. Mauricio Xavier Rea Peñafiel**

**Ibarra – Ecuador**

**2017**



## UNIVERSIDAD TÉCNICA DEL NORTE BIBLIOTECA UNIVERSITARIA

### AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

#### 1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital Institucional, determina la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información.

<b>DATOS DEL CONTACTO</b>	
<b>CÉDULA DE IDENTIDAD:</b>	1003063656-6
<b>NOMBRES Y APELLIDOS:</b>	REA ENRÍQUEZ DIEGO XAVIER
<b>DIRECCIÓN:</b>	CALLE BOLIVIA 3-35 Y COLOMBIA
<b>EMAIL:</b>	<a href="mailto:diegorea593@gmail.com">diegorea593@gmail.com</a>
<b>TELÉFONO MÓVIL:</b>	0999290170
<b>DATOS DE LA OBRA</b>	
<b>TÍTULO:</b>	“SISTEMA DE VOTO ELECTRÓNICO CON PROTOCOLOS DE CURVAS ELÍPTICAS APLICADO EN ELECCIONES POPULARES”
<b>AUTOR:</b>	REA ENRÍQUEZ DIEGO XAVIER
<b>FECHA:</b>	20 de Julio del 2017
<b>PROGRAMA:</b>	PREGRADO
<b>TÍTULO POR EL QUE OPTA:</b>	INGENIERO EN SISTEMAS COMPUTACIONALES
<b>DIRECTOR:</b>	Msc. Mauricio Xavier Rea Peñafiel

## **2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD**

Yo, Diego Xavier Rea Enríquez, con cédula de identidad Nro. 10030636566, en calidad de autor y titular de los derechos patrimoniales del trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación del trabajo en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.



## UNIVERSIDAD TÉCNICA DEL NORTE

### CESIÓN DE DERECHOS DE AUTOR

#### A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Diego Xavier Rea Enríquez, con cédula de identidad Nro. 1003063656, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5, 6, en calidad de autor del trabajo de grado denominado "SISTEMA DE VOTO ELECTRÓNICO CON PROTOCOLOS DE CURVAS ELÍPTICAS APLICADO EN ELECCIONES POPULARES" que ha sido desarrollado para optar por el título de Ingeniero en Sistemas Computacionales, en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes citada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte.

(Firma): 

Nombre: Diego Xavier Rea Enríquez

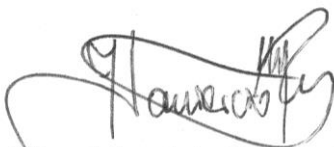
Cédula: 100306565

Ibarra, 20 de Julio del 2017

### CERTIFICACIÓN DEL ASESOR

Certifico que la Tesis previa a la obtención del título de Ingeniero en Sistemas Computacionales con el tema: "SISTEMA DE VOTO ELECTRÓNICO CON PROTOCOLOS DE CURVAS ELÍPTICAS APLICADO EN ELECCIONES POPULARES" ha sido desarrollada y terminada en su totalidad por el Sr. Diego Xavier Rea Enríquez con C.C. 100306365-6 bajo mi supervisión para lo cual firmo en constancia.

Atentamente,

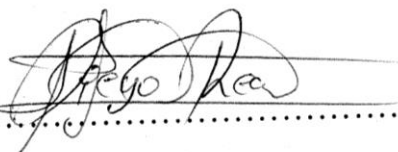


Msc. Mauricio Xavier Rea Peñafiel

DIRECTOR DE TESIS

### 3.- CONSTANCIAS

El autor manifiesta que la obra de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

(Firma):  .....

Nombre: Diego Xavier Rea Enríquez

Cédula: 1003063656

Ibarra, a los 20 días del mes de Julio del 2017

## **DEDICATORIA**

Primero a Dios, luego a mis padres que gracias a su apoyo incondicional hacen que hoy me encuentre en este punto trascendental de mi vida, ya que día a día fueron participes en el transcurso de este largo camino.

También a mis queridos hermanos que estuvieron en los buenos y malos momentos de mi vida.

Además a mis compañeros y amigos universitarios, que junto a ellos pude salir adelante en el ámbito académico.

Y no menos importante mi tutor de trabajo de grado y amigo Msc. Mauricio Rea, que gracias a su apoyo moral y profesional, supo encaminarme a lo largo de este largo camino.

## **AGRADECIMIENTOS**

Todos los éxitos conseguidos hasta ahora y los que se obtendrán en el futuro tendrán como primer responsable a la Universidad Técnica del Norte quién nos proporcionó las herramientas para desempeñarnos adecuadamente en nuestra profesión, para ella son los primeros agradecimientos.

Un agradecimiento muy especial al Msc. Mauricio Rea, Director de Trabajo de Grado por haberme brindado su guía en la elaboración del presente trabajo de grado.

Finalmente a todos mis amigos que han hecho de mi vida universitaria una grata experiencia.



## **RESUMEN**

En el presente trabajo de grado se plantea una solución a los sistemas de votación tradicional, para ser reemplazados con sistemas de voto electrónico.

Se enfoca principalmente al uso de seguridad web, utilizando protocolos de curvas elípticas, así mismo estos son más eficientes que los protocolos tradicionales caducos o más utilizados en sistemas web.

Este trabajo de grado va orientado para despejar dudas acerca de los temas relaciones en seguridad web, tipos de encriptación, programación java y base de datos relacionados con el tema principal.

Además puede considerarse una guía para elecciones populares de diferentes instituciones, ya que el sistema es versátil y se adapta a cualquier entorno.

## **ABSTRACT**

In the present work of degree poses a solution to the traditional voting systems, to be replaced with electronic voting systems.

It focuses mainly on the use of web security, using elliptical curve protocols, likewise these are more efficient than traditional protocols or more widely used in web systems.

This degree work is aimed at clearing doubts about the topics related to web security, types of encryption, java programming and database related to the main topic.

It can also be considered a guide for popular elections of different institutions, since the system is versatile and adapts to any environment.

## Contenido

<b>1. Introducción</b> .....	- 14 -
<b>1.1 Antecedentes</b> .....	- 14 -
<b>1.2 Situación actual</b> .....	- 15 -
<b>1.3 Prospectiva</b> .....	- 16 -
<b>1.4 Justificación</b> .....	- 16 -
<b>1.5 Planteamiento del problema</b> .....	- 17 -
<b>1.6 Objetivos</b> .....	- 17 -
<b>1.6.1 Objetivo general</b> .....	- 17 -
<b>1.6.2 Objetivos específicos</b> .....	- 17 -
<b>1.7 Justificación</b> .....	- 18 -
<b>1.8 Alcance</b> .....	- 18 -
<b>CAPÍTULO 2</b> .....	- 24 -
<b>2.1 Desarrollo ágil de software</b> .....	- 25 -
<b>2.1.1 Metodología SCRUM</b> .....	- 25 -
<b>2.1.2 Procesos SCRUM</b> .....	- 28 -
<b>2.1.3 Roles de SCRUM</b> .....	- 29 -
<b>2.2.1 Plataforma de desarrollo</b> .....	- 29 -
<b>2.2.1 Java</b> .....	- 29 -
<b>2.2.2 Primefaces</b> .....	- 30 -
<b>2.3.1 Protocolos de Curvas Elípticas</b> .....	- 32 -
<b>2.3.2 Ventajas de la Criptografía de Curvas Elípticas</b> .....	- 36 -
<b>2.4 Seguridad Informática</b> .....	- 36 -
<b>2.4.1 Criptografía</b> .....	- 38 -
<b>2.5 Gestor de Base de datos</b> .....	- 39 -
<b>2.5.1 PostgreSQL</b> .....	- 39 -
<b>2.6 Servidor</b> .....	- 40 -
<b>2.6.1 Windows Server 2012</b> .....	- 40 -
<b>2.6.2 Servidor de Aplicaciones</b> .....	- 40 -
<b>CAPÍTULO 3</b> .....	- 41 -
<b>3.1 Definición de Requerimientos</b> .....	- 41 -
<b>3.2 Análisis de Requerimientos</b> .....	- 44 -
<b>3.3 Diseño del Software</b> .....	- 44 -
<b>3.4 Codificación</b> .....	- 57 -
<b>3.5 Pruebas del Sistema</b> .....	- 66 -
	- 11 -

3.6 Implementación .....	- 67 -
<b>CAPÍTULO 4</b> .....	- 68 -
4.1 Conclusiones .....	- 69 -
4.2 Recomendaciones .....	- 69 -
4.3 Impacto Económico.....	- 69 -
4.4 Impacto Ambiental.....	- 71 -
4.5 Impacto Social .....	- 71 -
Bibliografía .....	- 72 -
<b>ANEXOS</b> .....	- 74 -

# **CAPÍTULO 1**

## **INTRODUCCIÓN**

### **CONTENIDO DEL CAPÍTULO**

- 1.1.** Antecedentes
- 1.2.** Situación actual
- 1.3.** Prospectiva
- 1.4.** Planteamiento del Problema
- 1.5.** Objetivos
  - 1.5.1.** Objetivo General
  - 1.5.2.** Objetivos Específicos
- 1.6.** Justificación
- 1.7.** Alcance

## **Introducción**

### **1.1 Antecedentes**

Durante muchos años el derecho al sufragio ha sido fundamental en la sociedad, ya que es el método por el cual se elige gobernantes públicos, referéndums populares aprobatorios, y al mismo tiempo un país que realiza votaciones es considerado un gobierno democrático.

En el Ecuador, se ha implementado a lo largo de la historia en dos oportunidades el voto electrónico más reciente fue en las elecciones seccionales del 23 de Febrero del 2014 en donde el Consejo Nacional Electoral realizó un plan piloto para la automatización del sufragio electoral en las provincias de Santo Domingo de los Tsáchilas, Azuay y en una zona rural de Pichincha (sector la Morita, parroquia Tumbaco, cantón Quito) en donde aproximadamente el número de electores asciende a un millón que ejercen su derecho al voto mediante dispositivos facilitados por Rusia, Venezuela y Argentina.

En la constitución de la República del Ecuador del 2008 contempla en el art. 219 que el Consejo Nacional Electoral tendrá las funciones: "organizar, dirigir, vigilar y garantizar, de manera transparente, los procesos electorales, convocar a elecciones, realizar los cómputos electorales, proclamar los resultados, y posesionar a los ganadores de las elecciones". (Asamblea Constituyente, Constitución de la República del Ecuador, 2008).

Para el año 2009, entra en vigencia el “Código de la Democracia” en el cual manifiesta expresamente que el CNE podrá decidir la utilización de métodos electrónicos de votación y/o escrutinio en forma total o parcial en los procesos electorales, dejando así la potestad al Órgano Electoral la implementación y desarrollo de la votación electrónica,

misma que está siendo estudiada por las áreas operativas del CNE para determinar la factibilidad o no de su implementación (Asamblea Nacional, 2009).

Al ser tan importante el derecho al voto, se ha realizado en algunos establecimientos del país este tipo de sufragio como es: la universidad de Guayaquil y unidades educativas como son: “Federico Gonzales Suarez” y “Medardo Alfaro” ubicadas en la ciudad de Guayaquil; para la elaboración en este sufragio se empleó un sistema realizado por funcionarios del Consejo Nacional Electoral de Santo Domingo con maquinaria propia de la administración, esto se ejecuta debido a la dependencia existente con otros países como son Venezuela, Rusia y Argentina, quienes colaboraban en este tipo de procesos de voto electrónico.

## **1.2 Situación actual**

En la actualidad en la mayoría de instituciones se debe realizar votaciones populares, ya sea para la designación de representantes, como para un mandato aprobatorio o consulta popular, además de ser un derecho de todos los ecuatorianos al ser Ecuador un país democrático.

Es difícil encontrar en el presente instituciones que cuenten con sistemas de voto electrónico seguro, al momento se lo realiza dirigiéndose a sufragar a los recintos electorales, que son lugares autorizados para poder ejercer el derecho al voto, en cada uno de estos recintos se encuentran mesas electorales, en los que se realiza de forma ordenada el sufragio persona a persona, o voto a voto, para luego ser proceder a entregar un certificado de votación, una vez culminado el horario establecido se realiza el conteo de voto a voto, para luego plasmarlo en una acta, para posteriormente ser enviada para un conteo global.

Las urnas son trasladadas por personal de seguridad y acompañados de fiscalizadores de los partidos políticos, para controlar la manipulación de las mismas, también existen veedores en cada mesa electoral para evitar posibles errores humanos, o fraudes electorales.

### **1.3 Prospectiva**

El incremento de ciudadanos, el avance de la tecnología dejará en desuso el común y rutinario sistema de votación que ya está caduco en otros países del mundo, por esto es que se innova, para no quedarse estancado tecnológicamente como institución, provincia y país. Se puede ser un referente para el norte del Ecuador al usar un sistema de voto electrónico seguro, aplicando protocolos que permiten evitar manipulación de datos, o errores humanos, que se han dado en elecciones populares pasadas.

### **1.4 Justificación**

El presente proyecto será pionero en el Norte del país, ya que en la actualidad no se cuenta con un sistema de voto electrónico, además de eficiente, con la seguridad que garantiza un proceso de transparencia en el derecho al voto que la constitución ampara, además la espera después del proceso para conocer resultados oficiales será disminuida de manera drástica, como lo es con el conteo voto a voto que se encuentra actualmente en desuso.

Una vez culminado este proyecto espero se pueda utilizar para todo tipo de elecciones populares, para el uso del público en general, y con todos los conocimientos aprendidos del mismo, poder asesorar, tanto en sistemas especializados de voto electrónico, como en seguridad informática, además de incursionar en temas relativamente nuevos para el crecimiento del país, y moral de los ciudadanos.



## **1.5 Planteamiento del problema**

En las elecciones populares al ser un pilar importante en el funcionamiento de la democracia, se debe contar con un sistema de elecciones electrónicas seguro, rápido y fiable que ayude al medio ambiente ya que en la actualidad no existen un sistema de control dentro de un sistema de democracia, y existen dificultades como son la imprecisión de los datos, fraudes electorales, ya que dentro de cada mesa electoral, existe gente afiliadas a cierto partidos, también un aspecto importante a destacar es la demora en la presentación de los resultados, y el costo que representa para el país cada vez que existen votaciones populares o de referéndum, además del impacto ambiental que tiene, tanto la creación de urnas, como papeletas para todos los votantes en donde consta cada partido con sus respectivos representantes.

## **1.6 Objetivos**

### **1.6.1 Objetivo general**

- Elaborar un sistema de voto electrónico para ser usado en elecciones populares que ayude a garantizar transparencia y seguridad en elecciones de dignidades usando protocolos de curvas elípticas.

### **1.6.2 Objetivos específicos**

- Estudiar protocolos de curvas elípticas para adicionar seguridad al sistema web.
- Comprobar el funcionamiento del sistema, aplicándolo en una elección popular en un curso de estudiantes que necesiten realizar elecciones populares.
- Desarrollo de un sistema de gestión de votación electrónica.

- Generar resultados oficiales después de que todos los votantes hayan ejercido el derecho al voto.
- Generar comprobantes de voto para cada votante, una vez realizado el voto electrónico.

### **1.7 Justificación**

En el importante desarrollo que ha tenido nuestro país en los últimos años y creciente demanda por parte de las personas que sufragan, se toma la iniciativa del desarrollo de un sistema de votación electrónica que sea capaz de adaptarse a las necesidades de las diferentes instituciones ya sean privadas o públicas utilizando como seguridad para el mismo protocolo de curvas elípticas que en la actualidad garantiza un cifrado más completo y capaz de realizarlo en menor tiempo.

También se ahorra recurso humano y tiempo para realizar estas votaciones ya que con este sistema agilizo el tiempo de espera, lo que causará satisfacción en los votantes ya que en la actualidad existen numerosos inconvenientes en las votaciones como son: largas filas en espera de poder sufragar, un desorden por el desconocimiento del proceso, largas y angustiosas horas de espera en el conteo voto a voto, en donde puede suceder un fraude electoral, ya que existe la posibilidad de error humano.

### **1.8 Alcance**

Este proyecto estará limitado por el uso de protocolos de seguridad informática como es el de curvas elípticas, y se lo realizará de manera que el sistema sea más rápido, ya que es un protocolo que usa mayor cifrado pero menor espacio de memoria, así como además se limita el aspecto de la base de datos, únicamente a ser pre- diseñada para un(os) curso superior de la unidad educativa “Ciudad de Ibarra” que necesite del servicio del mismo.

El proyecto también está limitado por un módulo de auditoría o bitácora, en donde se almacenará todas las personas que ingresen al sistema, ya sea como administradores o como usuarios votantes, así mismo una vez culminado el proceso electoral se imprimirá un comprobante de voto, que es relativamente más amigable para el medio ambiente.

Además, para un mayor acceso de los usuarios del sistema, se tomó en cuenta la disponibilidad de navegadores web, entre los que tenemos son: Internet Explorer, Mozilla Firefox, Google Chrome, Safari.

Para un buen funcionamiento y correcto uso de la aplicación se realizará guías técnicas de cómo utilizar dicho protocolo, y una guía de cómo se realizará el voto, que estará también disponible en físico, así como en el módulo de ayuda del ingreso a la página de voto.



*Fuente Propia*

*Figura 1.1 Diagrama de arquitectura*

## **Funciones del aplicativo a Desarrollar:**

Existe dos tipos de Roles:

- Votante
- Administrador
- Consultor

Para que el votante, pueda ejercer su derecho, debe estar empadronado, para esto el Administrador, se encargará de insertar directamente en la base de datos, los datos de todas las personas empadronadas, estas personas tendrán que ingresar su número de cédula acompañado de la clave proporcionada por el Administrador, que se generará de manera automática y tendrá por defecto la letra (e) además de integrar un número aleatorio de 6 dígitos que será notificado mediante correo electrónico para poder cambiar la contraseña a una de su preferencia, o mantener la misma, con el objeto de que únicamente el usuario conozca su clave.

El Administrador tiene acceso total a funciones operativas del sistema como son buscar, editar, eliminar e ingresar usuarios de tipo votantes con los datos más relevantes y necesarios para las respectivas notificaciones que se las realizarán vía correo electrónico, sin importar el proveedor del mismo.

El usuario votante tendrá la capacidad de votar por quien decida, y al final podrá generar un comprobante que certificará que esa persona ya realizó el voto y será notificado vía correo electrónico en donde se le proporcionará un código de 6 dígitos para brindar mayor seguridad al voto.

Los datos se estarán almacenando en la base de datos de PostgreSQL, la cual está montada en un servidor Windows Server 2012 r2 y solo tendrá acceso el administrador al

mismo. Vale recalcar, que el administrador también será registrado en la Bitácora del Sistema.

El rol consultor tiene acceso total a mirar resultados, así como también realizar búsquedas de personas empadronadas, y a inspeccionar la bitácora del sistema, pero con la condición de que no podrá manipular ninguno de estos datos, además de tener único acceso a mirar los resultados finales del proceso, y poder exportar los mismos.

### **Módulos del aplicativo a Desarrollar:**

Usuarios: Votar, Imprimir y un módulo de Ayuda para realizar el voto de manera ilustrada.

Conteo de Votos: Tiene acceso únicamente el administrador, se presentará mediante un reporte, todos los votos realizados hasta el momento que se decida, incluyendo en este reporte, personas que aún falta por votar, así como votos nulos, en blanco y por candidatos de la preferencia.

Ingreso de Usuarios votantes: para el ingreso dinámico de usuarios votante se presenta una pantalla en donde hace la inserción directamente en la base de datos y se genera automáticamente la contraseña así como se habilita para votar.

Bitácora: el ingreso a la bitácora, la puede revisar únicamente el administrador, en esta bitácora se presenta un reporte general con fecha y hora de ingreso de cada usuario votante así como del administrador, conjuntamente con la actividad que realizó y así mismo ingresos que posean contraseñas erróneas.

Administrador: para ingresar como administrador, se ha realizado una página web extra, de las mismas características del votante, pero con diferente dirección URL, esto se lo realiza por motivos de seguridad.

De igual manera existirá el modulo que empleará el consultor, en donde podrá acceder únicamente a funciones de revisión, tanto de resultados, como bitácoras del sistema.

La única persona autorizada a la impresión de resultados oficiales, será el consultor, que tendrá la potestad legal de hacer público los resultados, así como la vista de bitácora para fines pertinentes.

Como parte de un servicio seguro, existirá un computador totalmente independiente, en el que se podrá realizar cambios de contraseña, o restauración contraseñas que se lo realizará mediante correo electrónico personal, para que la atención sea más rápida y eficiente.



Fuente Propia

Figura 1.2 Diagrama de bloques

# CAPÍTULO 2

## MARCO TEÓRICO

### CONTENIDO DEL CAPÍTULO

#### **2.1** Desarrollo ágil de software

##### 2.1.1 Metodología SCRUM

##### 2.1.2 Procesos de SCRUM

##### 2.1.3 Roles de SCRUM

#### **2.2** Plataforma de desarrollo Java – Primefaces

#### **2.3** Curvas elípticas

##### 2.3.1 Protocolos de Curvas Elípticas

##### 2.3.2 Ventajas de la Criptografía de curvas elípticas

#### **2.4** Seguridad Informática

##### 2.4.1 Criptografía

##### 2.4.2 Seguridad Web

#### **2.5** PostgreSQL

#### **2.6** Servidor

##### 2.6.1 Windows Server

##### 2.6.2 Servidor de aplicaciones



## **2.1 Desarrollo ágil de software**

El desarrollo ágil de software rodea un enfoque basado en la toma de decisiones para proyectos de software que se basa principalmente en desarrollo iterativo e incremental donde los requisitos y las soluciones del mismo evolucionan con el pasar del tiempo, dependiente de las necesidades de mismo.

Al haber desarrollado el sistema de voto electrónico que se aplica para cualquier tipo de votación popular, se eligió la metodología scrum para la realización y post modificación del mismo, pudiendo acoplarse a las necesidades de cualquier entidad que requiera de este sistema. (Duarte, 2009)

El desarrollo ágil toma como filosofía la entrega rápida, las necesidades satisfechas del cliente, simplicidad y una buena comunicación entre el cliente y el desarrollador es por esto que las metodologías ágiles presentan una real ayuda para superar debilidades que existen en el desarrollo del software tradicional. (Madrid, <http://www.criptored.upm.es>, 2015)

### **2.1.1 Metodología SCRUM**

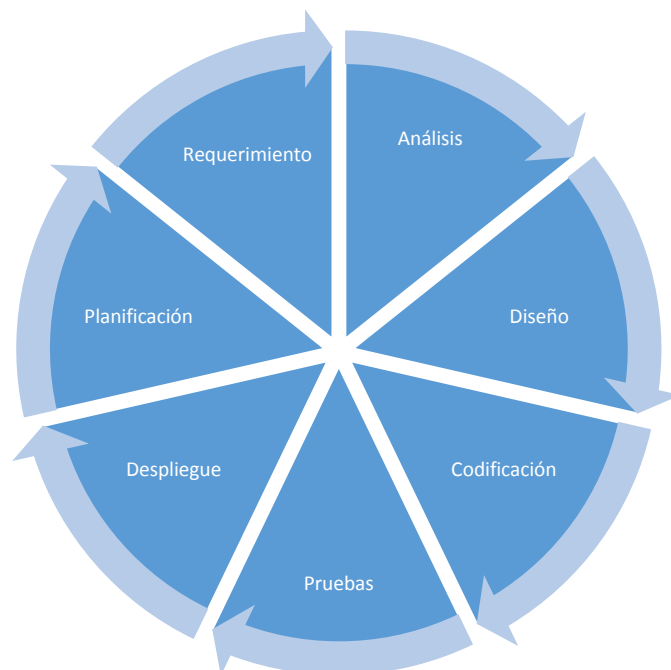
Al ser una metodología de desarrollo ágil para el desarrollo de proyectos y conjuntamente con las nuevas tendencias, requerimientos y adaptabilidad que necesita este proyecto, se ha seleccionado este tipo de metodología ya que tiene la gran ventaja de tener flexibilidad para cambios y presentar a los usuarios conjuntamente con avances de tipo funcional.

Este tipo de metodología busca entregar un software de calidad y totalmente funcional, respetando cinco etapas tradicionales para un proyecto en donde se permita entregas parciales que se irán complementando e incrementando mientras se avanza con la realización del proyecto. (Sanchez, 2016)

Las etapas de esta metodología ágil son:

- Definición de requerimientos.
- Análisis
- Diseño de Software
- Construcción, codificación.
- Pruebas de sistema.
- Implementación

A estas etapas se integra el despliegue que es una característica esencial para pasar siguiente etapa o llegar al fin del proyecto, a continuación en el gráfico se entiende de mejor manera.



*Figura 2. 1*

*Fuente Propia*

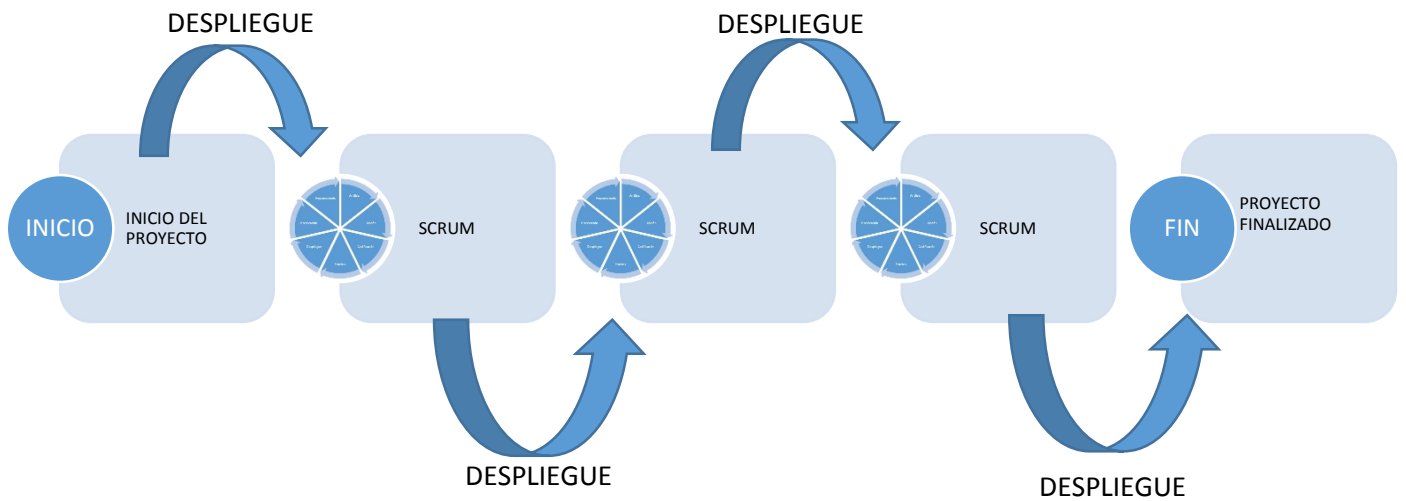


Figura 2. 2

Fuente Propia

Es una metodología de desarrollo ágil, cuyo principal objetivo es maximizar el rendimiento de la empresa para quien desarrolla el software, en donde el cliente se compromete con el sistema al mismo tiempo que se entusiasma ya que puede integrarse al equipo de trabajo para dar soluciones viables y cambios funcionales al sistema.

Esta metódica de trabajo promueve la innovación, motivación y compromiso del equipo que forma parte del proyecto, por lo que los profesionales encuentran un ámbito propicio para desarrollar sus capacidades. (Sanchez, 2016)

Los principales beneficios que obtenemos al aplicar este tipo de metodología ágil son: (Willis, 2016)

- Cumplimiento de expectativas
- Flexibilidad a cambios
- Reducción del tiempo para el mercado
- Mayor Calidad de Software
- Mayor Productividad
- Predicciones cercanas al tiempo de entrega
- Reducción de riesgos.

### **2.1.2 Procesos SCRUM**

El desarrollo se realiza de forma iterativa e incremental. Cada iteración, denominada Sprint, tiene una duración preestablecida de entre 2 y 4 semanas, obteniendo como resultado una versión del software con nuevas prestaciones listas para ser usadas. En cada nuevo Sprint, se va ajustando la funcionalidad ya construida y se añaden nuevas prestaciones priorizándose siempre aquellas que aporten mayor valor de negocio. (Sanchez, 2016).

La finalidad de realizar estos Sprint es poseer un feedback sobre lo que se está haciendo en cada momento con el proyecto, esto quiere decir que en el caso de posibles fallas, podemos volver a la versión anterior del mismo, también reducimos sustancialmente el riesgo de fallas en el software, ya que cualquier fallo afecta únicamente a la programación que se lleva a cabo en ese momento. (Singh, 2014)

Al principio de cada Sprint a realizarse se lleva a cabo una planificación con las tareas, objetivos y actividades que se va a desarrollar, conjuntamente con una fecha tentativa de entrega del mismo.

Para el control adecuado de este tipo de metodología se recomienda al menos reuniones pasando un día, ya que es importante el conocimiento del desarrollo y posibles problemas del mismo.

Al culminar cada Sprint se debe revisar y presentar la parte desarrollada en donde se analiza la mejora y funcionamiento de la parte desarrollada.

### **2.1.3 Roles de SCRUM**

En este tipo de desarrollo ágil se involucran algunas personas, entre ellas tenemos los siguientes roles: (Byous, 2005)

- SCRUM MASTER, es la persona que se encarga de liderar el proyecto, guiándolo para que cumpla las reglas y también con los procesos propios de la metodología.
- PRODUCT OWNER, o su traducción al español que es el dueño del producto, este rol pertenece al cliente que usa el software, traslada la visión para el uso del software al equipo.
- Team, es el equipo de profesionales que desarrollan el proyecto de manera conjunta.

### **2.2.1 Plataforma de desarrollo**

#### **2.2.1 Java**

A la plataforma Java se le conoce como entorno de computación que tiene la capacidad de ejecutar aplicaciones desarrolladas que han usado el lenguaje Java, para esto se utiliza un conjunto de herramientas que son llamadas herramientas para el desarrollo en Java (JDK).

La lenguaje Java fue desarrollado con el propósito de ser usado en combinación con la plataforma antes mencionada, para el buen uso del lenguaje se compone de varios

componentes y tecnologías, una de estas tecnologías o framework es Java Server Faces (JSF), dicho framework se lo usa ya que nos permite desarrollar de mejor manera aplicaciones de negocio de una forma dinámica en donde nuestra lógica se pueda implementar en el lenguaje java.

Java se creó como una herramienta de programación para ser usada en un proyecto de set-top-box en una pequeña operación denominada the Green Project en Sun Microsystems en el año 1991. El equipo (Green Team), compuesto por trece personas y dirigido por James Gosling, trabajó durante 18 meses en Sand Hill Road en Menlo Park en su desarrollo. (Byous, 2005)

La gran diferencia de Java con otros lenguajes de programación es que no es estrictamente dirigido a objetos, como es Ruby, por otra parte resulta que el lenguaje java resulta en muchas oportunidades ser redundante en código, lo que en la actualidad están tratando de solucionar con J2SE 5.0 que trata de reducir la redundancia en el código utilizando una nueva construcción en bucles.

### **2.2.2 Primefaces**

Una de las principales razones por las que se usó JSF en nuestro proyecto de voto electrónico, es que permite el ingreso de JavaScript en las páginas web, esto se lo realiza para acelerar la respuesta del cliente desde el navegador del mismo, además de que podemos usar la herramienta dispuesta para código HTML, que es PrimeFaces.

PrimeFaces integra novedosos componentes para la programación como es un completo editor de código HTML, o autocompletado de código, además de poseer gráficas para la integración en páginas web, también se lo toma muy en cuenta por su sencillez en la instalación, sin necesidad de complejas configuraciones, además de poseer una amplia difusión ya que en la actualidad existe una comunidad que se encarga de

respaldar a este novedoso y vistoso framework que cuenta con múltiples apariencias o temas listos para ser usados.

En específico PrimeFaces es una implementación de JSF (Pierini, 2014), también existen componentes para la interfaz del usuario en lo que compete a los dispositivos móviles, lo que además de integrar una forma organizada del código, facilita en gran proporción la programación al programador.

Primefaces al ser una librería de componentes visuales de código abierto activo, posee varias versiones que son estables entre cada versión lanzada, esta librería es desarrollada y mantenida por una empresa de tecnología de información de origen Turco denominada Prime Techonology.

Para los efectos de tipo visual que utiliza Primefaces es el soporte de jQuery, además que dispone de 26 temas ya prediseñados y más de 90 componentes en código abierto que son mantenidos por la empresa antes mencionada.



*Fuente: [www.primefaces.org](http://www.primefaces.org)*

*Figura 2. 3*

### 2.3.1 Protocolos de Curvas Elípticas

Una curva elíptica es una curva que en el plano se ubica de tal manera que cada línea que la corta es en dos puntos y la corta exactamente en un tercer punto de esta manera:

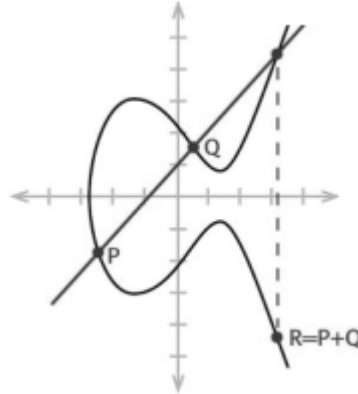


Figura 2. 4.1

Fuente: <http://filosofiahacker.blogspot.com/cambiar> fuent

En las últimas décadas, la criptografía con curvas elípticas ha adquirido una creciente importancia, llegando a formar parte de los estándares industriales. Su principal logro se ha conseguido en los criptosistemas basados en el problema del logaritmo discreto, como los de tipo ElGamal. Estos criptosistemas planteados en el grupo de puntos de una curva elíptica garantizan la misma seguridad que los construidos sobre el grupo multiplicativo de un cuerpo finito, pero con longitudes de clave mucho menor. (Josep M. Miret Javier Valera & Magda Valls, 2015)

¿Cómo es la aplicación?

Dentro de los tipos de criptografía que tenemos dos tipos como son:

- Simétrica
- Asimétrica

Criptografía simétrica, utiliza un sola clave para cifrar y descifrar un mensaje, para poder usar este tipo de criptografía es necesario poseer una clave previamente, lo que el nivel de seguridad es más bajo. (Josep M. Miret Javier Valera & Magda Valls, 2015)



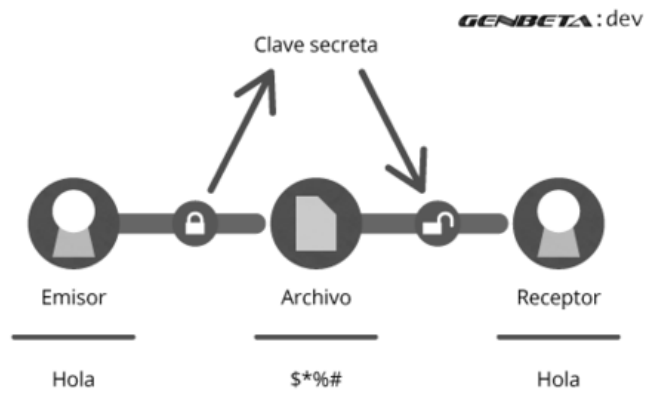


Figura 2. 4.2

Fuente: <http://www.genbetadev.com/>

Criptografía Asimétrica, este tipo de criptografía tiene como principal requisito el uso de dos tipos de clave: (Data, 2016)

- Clave Pública, difundida sin ningún problema.
- Clave Privada, es una clave secreta para poder mirar el archivo.

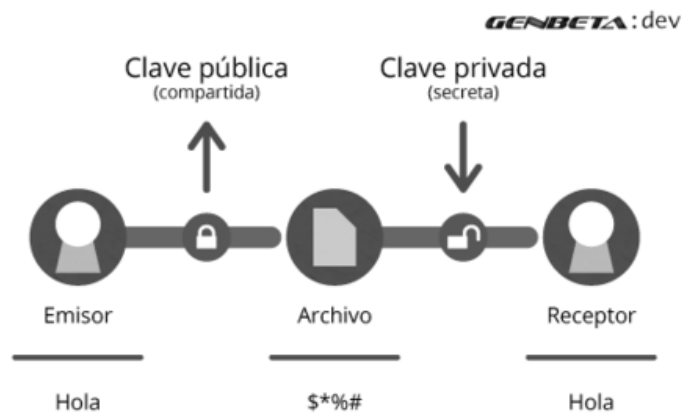


Figura 2. 4.3

Fuente: <http://www.genbetadev.com/>

La criptografía de curvas elípticas es una variante la criptografía asimétrica ya que se necesita dos tipos de clave, una que es pública y privada, la clave pública, no posee toda la información necesaria para descifrar la clave privada.

La criptografía con curvas elípticas aparece como una alternativa a los criptosistemas de clave pública clásicos como el RSA y el ElGamal, tanto por la disminución del tamaño de las claves que se requieren como por el abanico de grupos que ofrecen en el mismo cuerpo base. Su implantación en algunos sistemas de comunicaciones es un hecho constatable y su uso aumenta día a día debido a sus ventajas. Por ejemplo, se usa en tarjetas inteligentes, sistemas de identificación por radio frecuencia, sistemas de votación electrónica, etc. (Josep M. Miret Javier Valera & Magda Valls, 2015)

En nuestra aplicación se utiliza la criptografía de curva elíptica ya que es más rápida y las claves de cifrado son más cortas que otras como tenemos a la RSA y al mismo tiempo este brindan una mayor seguridad. (Interactiva, 2011)

Además se vio la necesidad de estudiar este tipo de método, ya que es el último método de cifrado, a forma de comparación he realizado este cuadro:

Tabla 2.4.1

*Comparativa ECC vs.Otras claves criptográficas*

ECC (ELLIPTIC CURVE CRYPTOGRAPHY)	OTRAS CLAVES CRIPTOGRÁFICAS
256 Bits	3072 Bits

*Fuente: Propia*

Así se puede apreciar en la tabla 2.1 que la ECC ofrece la misma seguridad que otras claves criptográficas, como podemos mirar, la clave es más corta, esto se traduce a que se necesita menor potencia informática para obtener conexiones más rápidas y seguras, lo que es ideal ya que además cuenta con el respaldo de la Agencia de Seguridad Nacional de Estados Unidos.

Para este tipo de cifrado disponemos tanto en Windows como en Linux, el uso inmediato de dichos protocolos mediante los navegadores, que se hace el intercambio de claves mediante certificaciones de tipo ECDSA para esto usamos el software denominado Open SSL.



Figura 2. 4.4

Fuente: <http://www.en.xolido.com/>

ECDSA. Conocido como Elliptic Curve Digital Signature Algorithm. Es más reciente que el RSA. Es más seguro que el RSA ya que con claves más pequeñas se consigue el mismo nivel de seguridad que claves más grandes en el RSA. Al tener claves más pequeñas también es mejor algoritmo ya que mejora el rendimiento de las aplicaciones de cifrado, descifrado y firma digital. (Anónimo, 2015)

Open SSL, es una biblioteca que posee soporte para los protocolos tanto de intercambio de claves (Diffie Hellman), así como algoritmos de firma digital que integran curvas elípticas (ECDSA), de este modo, esta biblioteca se la emplea para la correcta utilización de los protocolos estudiados. (Data, 2016)

La principal ventaja de este tipo de protocolos es en la reducción de tamaño de la clave, y por lo tanto esto influye en la velocidad de transferencia de los datos, para hacerlo de manera segura, para esto el siguiente cuadro explica de mejor manera.

Longitud de la llave simétrica	Longitud de clave asimétrica estándar	Longitud de la curva elíptica
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Figura 2. 4.2

Fuente: <http://www.openssl.org>

Como podemos apreciar en el gráfico, para una clave simétrica de 256 bits, en el caso de usar una clave asimétrica estándar, llevaría 15360 bits, lo que no sería práctico en el asunto de optimizar los recursos, mientras que la clave aplicando las curvas elípticas, es completamente práctico con 512 bits. (Interactiva, 2011)

### 2.3.2 Ventajas de la Criptografía de Curvas Elípticas

Las principales ventajas al usar ECC son:

- Mínimos requerimientos de memoria y tasa de transmisión, estos requerimientos en comparación con otras criptografías, son considerablemente más reducidas.
- Escalabilidad, el nivel de seguridad que presta las curvas elípticas es elevado, con poco costo de producción y mayor optimización de recursos para conseguir el mismo resultado.

## 2.4 Seguridad Informática

Como el nombre del mismo manifiesta en asegurar los recursos del sistema de información para garantizar esto debe cumplir los tres pilares:

- Confidencialidad
- Integridad
- Disponibilidad

Al haber desarrollado una página web con nivel de impacto alto, en donde la confidencialidad, integridad y disponibilidad de los datos, se encuentran expuestos a posibles ataques por lo que existen diferentes técnicas de seguridad. (Encrypt, 2016)

La disponibilidad de los datos refiere a los datos de un sistema sea accesible, y utilizable para que se recupere dicha información cuando se lo necesite.

La integridad de los datos es una cualidad, que indica que los datos no han sido manipulados ni alterados desde el documento original.

La confidencialidad es la privacidad de los datos, mediante métodos que protejan el sistema de invasiones de personas no deseadas, que solo esperan para realizar un fraude electoral.

Así mismo para evitar este tipo de ataques se posee mecanismo de seguridad, todo estos mecanismos depende del tipo de sistemas, del tipo de función y los factores de riesgos que amenazan al sistema o al desarrollo del mismo.

- Preventivos
- Correctivos
- Detectivos

Todos los sistemas son programados, así que cualquier persona que tenga el conocimiento y el deseo de reprogramar nuestro sistema lo puede hacer, por lo que puede tener acceso a información delicada e importante. (Data, 2016)

Las contraseñas son las herramientas más utilizadas para restringir el acceso a los sistemas informáticos. Sin embargo, sólo son efectivas si se escogen con cuidado, la mayor parte de los usuarios de computadoras escogen contraseñas que son fáciles de

adivinar: El nombre de la pareja, el de un hijo o el de una mascota, palabras relacionadas con trabajos o aficiones o caracteres consecutivos del teclado. (Rios, 2010)

Los hackers conocen y explotan estas vulnerabilidades, por lo que un usuario precavido no debe utilizarlos. Muchos sistemas de seguridad no permiten que los usuarios utilicen palabras reales o nombres como contraseñas, evitando así que los hackers puedan usar diccionarios para adivinarlas. Incluso la mejor contraseña debe cambiarse periódicamente. (Rios, 2010).

#### **2.4.1 Criptografía**

La criptografía responde a la necesidad de codificar mensajes que sólo pueda descifrar el destinatario y se ha aplicado tanto a defensa, como a secretos industriales y, en los últimos años, sobre todo, al comercio electrónico. Esto es así porque actualmente la seguridad de los sistemas informáticos se ve debilitada por el fuerte crecimiento de las redes y cuando se trata este tema hay que tener en cuenta un aspecto tan importante como la privacidad e integridad de los datos. (Mateo, 2015)

Los Objetivos principales de la criptografía son:

- Privacidad, es asegurar mediante cifrado que nadie es capaz de observar la información que se envía de un extremo a otro.
- Integridad de los datos, se debe asegurar que el mensaje en ningún momento ha sido alterado.
- Autenticación, se debe asegurar que tanto el que envía el mensaje como el que lo recibe son la personas que dicen ser.
- No repudio, es la imposibilidad de negar cualquier tipo de mensaje por parte del emisor.

## 2.5 Gestor de Base de datos

### 2.5.1 PostgreSQL

PostgreSQL es un sistema de gestión de base de datos de código abierto, el desarrollo del mismo, como la mayoría de proyectos de código abierto, es manejado por una comunidad de desarrolladores.

Las principales ventajas del uso de PostgreSQL son:

- Seguridad en Términos Generales.
- Integridad en Base de datos.
- Disparadores (Triggers)
- Conexión a DBMS
- Transacciones y Respaldos

A lo largo de la historia se ha ido implementando código y comentarios al desarrollo del mismo, por lo que PostgreSQL ha tenido un cambio desde su primera versión, hasta llegar a la versión 9 que usamos para nuestro sistema de voto electrónico como principal gestor de base de datos.

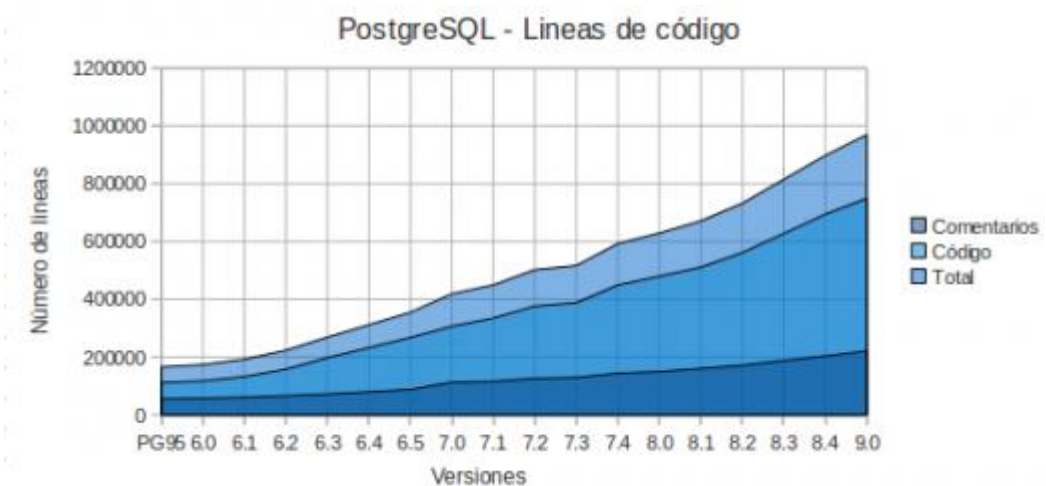


Figura 2.6. 1

Fuente: [www.postgresql.org.es/](http://www.postgresql.org.es/)

## **2.6 Servidor**

### **2.6.1 Windows Server 2012**

Es un sistema operativo para optimizar las funciones de programación que se emplearon en el desarrollo de este proyecto, y se presenta como una versión similar a Microsoft Windows 8 que posee una interfaz de tipo Modern UI y viene por defecto Windows Powershell con más comandos que su versión anterior.

Nuestro servidor posee una Ip pública que identifica de manera lógica y jerárquica a una interfaz de dispositivo y que se encarga de tener un punto de enlace con internet.

Conjuntamente se realiza la configuración para que mediante un puerto por defecto de tipo “S” seguro, pueda tener enlace a internet y poder mirar directamente el desarrollo de la aplicación aquí realizada.

### **2.6.2 Servidor de Aplicaciones**

#### **2.6.2.1 Wildfly**

Wildfly es la aplicación mejorada de JBoss que es más ligera y flexible que su predecesora, al ser una aplicación de código abierto se implementó en este proyecto, y está disponible para ser usado en muchas plataformas, se basa en Java EE.

Este servidor de aplicación se implementó con una red de colaboradores que posee a nivel mundial, mediante sus portales web, que comparten información y mejoran a esta herramienta, lo que lo hace perfecto para el uso en empresas grandes, así como en la elaboración de este proyecto.



# **CAPÍTULO 3**

## **DESARROLLO e implementación**

### **CONTENIDO DEL CAPÍTULO**

- 3.1 desarrollo de la solución
- 3.2 Definición de Requerimientos(son genéricos)
- 3.3 Análisis de Requerimientos
- 3.4 Diseño de Software
- 3.5 Codificación
- 3.6 Pruebas del Sistema
  
- 3.7 Implementación

#### **3.1 Definición de Requerimientos**

La unidad educativa “Ciudad de Ibarra” que se encuentra en la ciudad de Ibarra, provincia de Imbabura tiene la necesidad de contar un sistema para elecciones populares que permita resultados confiables, y con rapidez, que a su vez sea fácil de manejar tanto por votantes, así como por las personas que se encargarán del buen uso del sistema.

### **Requerimientos Funcionales**

- El sistema debe permitir mediante una página de login controlar el ingreso de los 3 roles: Administrador, Consultor y Votante.
- El Administrador se debe encargar del ingreso de nuevos votantes o usuarios empadronados para el sistema de votación.
- La persona que está Administrando el Sistema se encargará de crear, los procesos electorales vigentes, así como también, las listas y dignidades presentes en cada proceso electoral.
- El sistema integra una bitácora de sistema, en ella se refleja todas las personas que ingresan al sistema con sus respectivas cualidades importantes.
- El votante iniciará el proceso de votación realizando una comprobación de su identidad mediante el envío de un correo electrónico.
- El Consultor exclusivamente tiene acceso a toda la información como son: registro de los procesos de elecciones vigentes, lista de empadronados, un listado con todas las listas inscritas para el proceso, así también las dignidades seleccionadas con el detalle que no podrá modificarla, ni eliminarla, de los registros.
- Comprobar que tanto listas y empadronados pertenezcan a un proceso electoral.

### Requerimientos No funcionales

- Las partes del menú principal están en lugares visibles para los usuarios del sistema, sin importar el rol que tengan.
- El rendimiento del sistema será óptimo y con un alto nivel de seguridad web.
- La facilidad de uso para este sistema es que se lo pueda mirar en todo el mundo mediante un portal desde cualquier navegador.
- El sistema entregará resultados al instante, mientras va transcurriendo la votación.

### Requerimientos Externos:

- El sistema de voto electrónico es funcional para los siguientes navegadores, ya que estos tienen la capacidad de aceptar la criptografía de curvas elípticas.

Web client	Minimum version required
Mozilla Firefox	2.0
Google Chrome	1.0 on ECC compatible OS
Microsoft IE	7 on ECC compatible OS
Apple Safari	4 on ECC compatible OS

- El sistema de voto electrónico se encarga de comprobar la validez del usuario al enviar un correo electrónico al mail antes ingresado por el administrador que pertenece al usuario, de esta manera el usuario realiza

el ingreso de una nueva contraseña para tener acceso a votar, el correo electrónico puede ser institucional o privado.

### 3.2 Análisis de Requerimientos

Una vez analizados todos los requerimientos que se necesitan para este proyecto, se decide realizarlos todos, ya que todos cumplen al ser:

- Óptimos
- Adaptables
- Aplicados a situaciones reales
- No poseen contradicciones
- Necesarios y/o correctos
- Viables y/o Factibles
- Consistentes
- Demostrable

### 3.3 Diseño del Software

Los valores que deben ser practicados por todos los miembros involucrados en el desarrollo y que hacen posible que la metodología Scrum tenga éxito son:

- Autonomía del equipo
- Respeto en el equipo
- Responsabilidad y auto-disciplina
- Foco en la tarea
- Información transparencia y visibilidad.

Roles del proyecto

<b>Persona</b>	<b>Contacto</b>	<b>Rol</b>
Msc. Mauricio Rea	0986099536	SCRUM MASTER

Universidad Técnica del Norte	062997800	PRODUCT OWNER
Diego Rea	0999290170	EQUIPO TÉCNICO
Ing. Antonio Quiña	0999208083	SCRUM MASTER

## PLANIFICACIÓN DE TRABAJOS DE DESARROLLO

<b>Development:</b>	Diego Rea	
<b>Coordinador:</b>	Antonio Quiña	Mauricio Rea
<b>Sprint</b>	1	
<b>Fechas</b>	01/08/2016	
	19/08/2016	
<b>Total horas</b>	87	

PROYECTO	ACTIVIDAD	Fase Desarrollo	Tipo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO SUGERIDO (Horas)	TIEMPO REAL (Horas)	ESTADO
VOTO E	AMBIENTE	Desarrollo	Nuevo	Instalacion de Programas	4	4	4	DONE
		Desarrollo	Nuevo	Instalación de Gestor de Base de datos	4	4	4	DONE
		Desarrollo	Nuevo	Información Relevante Votaciones	4	4	4	DONE
		Desarrollo	Nuevo	Adición de código de consultas de compañeros	3	3	3	DONE
		Desarrollo	Nuevo	Metodo de carga de consulta instituciones	1	1	1	DONE
		Desarrollo	Nuevo	Revisión de reportes generados	2	2	2	DONE
		Desarrollo	Nuevo	Adición de de login local	6	6	6	DONE
		Desarrollo	Nuevo	Cambios Sis_Colegio	2	2	2	DONE
	Formateo de Computadora	Desarrollo	Nuevo	Formateo de Computadora	3	3	3	DONE
		Desarrollo	Nuevo	Instalación de Programas	4	4	4	DONE
		Desarrollo	Nuevo	Carga de Proyectos anteriores	1	1	1	DONE
	BASE DE DATOS	Desarrollo	Nuevo	Reunion de sistema de Voto Electrónico	3	3	3	DONE
		Desarrollo	Nuevo	Determinación de BDD	3	3	3	DONE
		Desarrollo	Nuevo	Carga de Ingresos de Procesos Electorales	2	2	2	DONE
		Desarrollo	Nuevo	Prueba de la base de datos v1	3	3	3	DONE
		Desarrollo	Nuevo	Modelamiento v1, de la base de datos	4	4	4	DONE
		Desarrollo	Nuevo	Generación de la base de datos	2	2	2	DONE
	General	Reuniones	Planificación	Nuevo	Reunión de Socialización Scrum	4	4	4
Planificación			Nuevo	Reunión de planificación.	4	4	4	DONE
Revisión			Nuevo	Reunión de Revisión de desarrollos.	2	2	2	DONE
Logística en territorio		Territorio	Nuevo	Logística en territorio.	10	10	10	DONE
Tareas no planificadas				0	0	0	DONE	
<b>TOTAL</b>					<b>71</b>	<b>71</b>	<b>71</b>	

## PLANIFICACIÓN DE TRABAJOS DE DESARROLLO

<b>Development:</b>	Diego Rea								
<b>Coordinador:</b>	Antonio Quiña	Mauricio Rea							
<b>Sprint</b>	2								
<b>Fechas</b>	22/08/2016								
	09/09/2016								
<b>Total horas</b>	88								

PROYECTO	ACTIVIDAD	Fase Desarrollo	Tipo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO SUGERIDO (Horas)	TIEMPO REAL (Horas)	ESTADO
VOTO ELECTRO	Ambiente de producción	Desarrollo	Nuevo	Cambios en base de datos	3	3	3	DONE
		Desarrollo	Nuevo	Configuración Servidor de aplicaciones	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas funcionales v1	2	2	2	DONE
VOTO ELECTRO	Reportes	Desarrollo	Nuevo	Prototipos de Reportes	2	2	2	DONE
		Desarrollo	Nuevo	Metodos de reporte de resultados totales	2	2	2	DONE
		Desarrollo	Nuevo	Vincular desarrollo	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas funcionales del primer módulo	2	2	2	DONE
	Gestión de Ingreso	Desarrollo	Nuevo	Creación de módulo de Votante	6	6	6	DONE
		Desarrollo	Nuevo	Cambios en base de datos sobre Votantes Empadronados	2	2	2	DONE
		Desarrollo	Nuevo	Cambio en base de datos acerca de ID y Clave	2	2	2	DONE
		Desarrollo	Nuevo	Edición de Ingresos tabla Persona	2	2	2	DONE
		Desarrollo	Nuevo	Cambios en base de datos	2	2	2	DONE
		Desarrollo	Nuevo	Edición de imágenes de Ingreso	2	2	2	DONE
		Desarrollo	Nuevo	Subir cambios a producción	1	1	1	DONE
	Segunda Fase	Pruebas	Nuevo	Pruebas funcionales	4	4	4	DONE
		Desarrollo	Nuevo	Creación de scripts para preguntas para segunda fase de evaluaciones	4	4	4	DONE
Desarrollo		Nuevo	Creación de imágenes Voto electrónico v1	2	2	2	DONE	
Análisis Sitio Web	Análisis	Desarrollo	Nuevo	Creación de métodos para evaluación de Química	2	2	2	DONE
		Análisis	Nuevo	Análisis de sitio web pruebas de Rendimiento	2	2	2	DONE
		Análisis	Nuevo	Elaborar informe resultante Tesis v1	3	3	3	DONE
GESTIÓN PROCESOS ELECTORALES	Investigación	Análisis	Nuevo	Editar formato de informe de análisis de sitio web	2	2	2	DONE
		Investigación	Nuevo	Investigación sobre tipos de votaciones	2	2	2	DONE
CONSULTAS	Investigación	Investigación	Nuevo	Instalación y revisión de sistemas de gestión de Procesos Electorales	2	2	2	DONE
		Investigación	Nuevo	Definición de Primefaces	3	3	3	DONE
		Investigación	Nuevo	Utilización de Primefaces	3	3	3	DONE
		Investigación	Nuevo	Busqueda de web service para consumo de Primefaces	3	3	3	DONE
		Investigación	Nuevo	Consulta de utilización de componentes	2	2	2	DONE
		Investigación	Nuevo	Consulta sobre Wildfly	2	2	2	DONE
General	Reuniones	Investigación	Nuevo	Investigación sobre instalación del gestor de bases de dato	2	2	2	DONE
		Planificación	Nuevo	Reunión de Socialización Scrum, Retrospectiva Sprint	4	4	4	DONE
		Planificación	Nuevo	Reunión de planificación.	2	2	2	DONE
	Revisión	Nuevo	Reunión de Revisión de desarrollos.	2	2	2	DONE	
	Logística en territorio	Territorio	Nuevo	Logística en territorio.	10	10	10	DONE
	Tareas no planificadas				0	0	0	
<b>TOTAL</b>					<b>88</b>	<b>88</b>	<b>88</b>	

## PLANIFICACIÓN DE TRABAJOS DE DESARROLLO

<b>Development:</b>	Diego Rea								
<b>Coordinador:</b>	Antonio Quiña	Mauricio Rea							
<b>Sprint</b>	3								
<b>Fechas</b>	12/09/2016								
	30/09/2016								
<b>Total horas</b>	89								

PROYECTO	ACTIVIDAD	Fase Desarrollo	Tipo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO SUGERIDO (Horas)	TIEMPO REAL (Horas)	ESTADO
VOTO ELECTRO	Ambiente de producción	Desarrollo	Nuevo	Cambios en base de datos	3	3	3	DONE
		Desarrollo	Nuevo	Constante Limpiezas de proyecto y codigo redundante	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas funcionales v1,1	2	2	2	DONE
VOTO ELECTRO	Reportes	Desarrollo	Nuevo	Reporte para Consultor y Administrador de Proceso Elecciones	2	2	2	DONE
		Desarrollo	Nuevo	Ingreso en bitácora de Cosultores	2	2	2	DONE
		Desarrollo	Nuevo	Vincular desarrollo	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas funcionales del primer módulo	2	2	2	DONE
	Gestión de Ingreso	Desarrollo	Nuevo	Creación de módulo de Votante	6	6	6	DONE
		Desarrollo	Nuevo	Cambios en base de datos sobre tipo de usuarios	2	2	2	DONE
		Desarrollo	Nuevo	Cambio en base de datos de los tipos de datos	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas de Base de datos mediante archivos excel	2	2	2	DONE
		Desarrollo	Nuevo	Cambios en base de datos	2	2	2	DONE
		Desarrollo	Nuevo	Edición de imágenes de Ingreso	2	2	2	DONE
		Desarrollo	Nuevo	Subir cambios a producción	1	1	1	DONE
		Pruebas	Nuevo	Pruebas funcionales v1,1	4	4	4	DONE
	Segunda Fase	Desarrollo	Nuevo	Creación de scripts para funcionamiento del Modulo Votantes	4	4	4	DONE
		Desarrollo	Nuevo	Creación de imágenes Voto electrónico v1,1 Modulo Votantes	2	2	2	DONE
Análisis Sitio Web	Análisis	Análisis	Nuevo	Análisis de sitio web pruebas de Rendimiento	2	2	2	DONE
		Análisis	Nuevo	Elaborar informe resultante Tesis v1	3	3	3	DONE
		Análisis	Nuevo	Editar formato de informe de análisis de sitio web	2	2	2	DONE
VOTANTES EMPADRONADOS	Investigación	Investigación	Nuevo	Investigación sobre requisitos para empadronar Votantes	2	2	2	DONE
		Investigación	Nuevo	Investigación de envío de resultados y cambios de contraseña mediante mail	2	2	2	DONE
CONSULTAS	Investigación	Investigación	Nuevo	Enlace de base de datos Postgres y Java	3	3	3	DONE
		Investigación	Nuevo	Utilización de componente de Primefaces	3	3	3	DONE
		Investigación	Nuevo	Consulta sobre comparativas de Login	4	4	4	DONE
		Investigación	Nuevo	Consulta de utilización de componentes	2	2	2	DONE
		Investigación	Nuevo	Consulta sobre Wildfly y posibles fallos	2	2	2	DONE
General	Reuniones	Planificación	Nuevo	Reunión de Socialización Scrum, Retrospectiva Sprint	4	4	4	DONE
		Planificación	Nuevo	Reunión de planificación.	2	2	2	DONE
		Revisión	Nuevo	Reunión de Revisión de desarrollos.	2	2	2	DONE
	Logística en territorio	Territorio	Nuevo	Logística en territorio.	10	10	10	DONE
	Tareas no planificadas				0	0	0	
<b>TOTAL</b>					89	89	89	



## PLANIFICACIÓN DE TRABAJOS DE DESARROLLO

<b>Development:</b>	Diego Rea								
<b>Coordinador:</b>	Antonio Quiña	Mauricio Rea							
<b>Sprint</b>	4								
<b>Fechas</b>	03/10/2016								
	21/10/2016								
<b>Total horas</b>	90								

PROYECTO	ACTIVIDAD	Fase Desarrollo	Tipo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO SUGERIDO (Horas)	TIEMPO REAL (Horas)	ESTADO
VOTO ELECTRO	Ambiente de producción	Desarrollo	Nuevo	Cambios en base de datos	3	3	3	DONE
		Desarrollo	Nuevo	Constante Limpiezas de proyecto y codigo redundante	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas funcionales v1,2	2	2	2	DONE
VOTO ELECTRO	Reportes	Desarrollo	Nuevo	Reportes actualizados para Consultor y Administrador de Proceso Elecciones	2	2	2	DONE
		Desarrollo	Nuevo	Ingreso en bitácora de Cosultores, y Administradores	2	2	2	DONE
		Desarrollo	Nuevo	Vincular desarrollo con la v1,2	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas funcionales v1,1	2	2	2	DONE
	Gestión de Ingreso	Desarrollo	Nuevo	Creación de módulo de Persona	6	6	6	DONE
		Desarrollo	Nuevo	Cambios en base de datos sobre tipo de usuarios	2	2	2	DONE
		Desarrollo	Nuevo	Cambio en base de datos de los tipos de datos	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas de Base de datos mediante archivos excel	2	2	2	DONE
		Desarrollo	Nuevo	Cambios en base de datos	2	2	2	DONE
		Desarrollo	Nuevo	Edición de imágenes de Roles	2	2	2	DONE
		Desarrollo	Nuevo	Subir cambios a producción	1	1	1	DONE
	Segunda Fase	Pruebas	Nuevo	Pruebas funcionales v1,1	4	4	4	DONE
		Desarrollo	Nuevo	Creación de scripts para funcionamiento del Modulo Votantes	4	4	4	DONE
Desarrollo		Nuevo	Creación de imágenes Voto electrónico v1,2 Modulo Control	2	2	2	DONE	
Análisis Sitio Web	Análisis	Desarrollo	Nuevo	Creación de métodos de acceso al sistema	2	2	2	DONE
		Análisis	Nuevo	Análisis de sitio web pruebas de Rendimiento	2	2	2	DONE
		Análisis	Nuevo	Elaborar informe resultante Tesis v1	3	3	3	DONE
VOTANTES EMPADRONADOS	Investigación	Análisis	Nuevo	Editar formato de informe de análisis de sitio web	2	2	2	DONE
		Investigación	Nuevo	Investigación sobre requisitos para poder ser Candidatos y Listas registradas	2	2	2	DONE
CONSULTAS	Investigación	Investigación	Nuevo	Investigación de envío de resultados y cambios de contraseña mediante mail	2	2	2	DONE
		Investigación	Nuevo	Enlace de base de datos Postgres y Java	3	3	3	DONE
		Investigación	Nuevo	Utilización de componente de Primefaces	5	5	5	DONE
		Investigación	Nuevo	Consulta sobre metodos de ingreso de datos	2	2	2	DONE
		Investigación	Nuevo	Consulta de utilización de componentes	2	2	2	DONE
General	Reuniones	Investigación	Nuevo	Consulta sobre Wildfly y posibles fallos	2	2	2	DONE
		Investigación	Nuevo	Investigación sobre archivos csv y archivos planos	3	3	3	DONE
		Planificación	Nuevo	Reunión de Socialización Scrum, Retrospectiva Sprint	4	4	4	DONE
	Logística en territorio	Planificación	Nuevo	Reunión de planificación.	2	2	2	DONE
Revisión		Nuevo	Reunión de Revisión de desarrollos.	2	2	2	DONE	
Tareas no planificadas	Territorio	Nuevo	Logística en territorio.	10	10	10	DONE	
<b>TOTAL</b>					90	90	90	

## PLANIFICACIÓN DE TRABAJOS DE DESARROLLO

<b>Development:</b>	Diego Rea								
<b>Coordinador:</b>	Antonio Quiña	Mauricio Rea							
<b>Sprint</b>	5								
<b>Fechas</b>	24/10/2016								
	18/11/2016								
<b>Total horas</b>	91								

PROYECTO	ACTIVIDAD	Fase Desarrollo	Tipo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO SUGERIDO (Horas)	TIEMPO REAL (Horas)	ESTADO
VOTO ELECTRO	Ambiente de producción	Desarrollo	Nuevo	Cambios en base de datos	3	3	3	DONE
		Desarrollo	Nuevo	Constante Limpiezas de proyecto y codigo redundante	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas funcionales v2,0	2	2	2	DONE
VOTO ELECTRO	Reportes	Desarrollo	Nuevo	Reportes actualizados para Consultor y Administrador de Listas	2	2	2	DONE
		Desarrollo	Nuevo	Ingreso en bitácora de Cosultores, y Administradores	2	2	2	DONE
		Desarrollo	Nuevo	Vincular desarrollo con la v1,2	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas funcionales v1,2	2	2	2	DONE
	Gestión de Ingreso	Desarrollo	Nuevo	Creación de módulo de Consultor	6	6	6	DONE
		Desarrollo	Nuevo	Creacion de módulo Administrador	2	2	2	DONE
		Desarrollo	Nuevo	Creación de módulo de Persona Votnte	2	2	2	DONE
		Desarrollo	Nuevo	Registro de Candidatos	2	2	2	DONE
		Desarrollo	Nuevo	Cambios en base de datos	2	2	2	DONE
		Desarrollo	Nuevo	Edición de imágenes de Candidatos	2	2	2	DONE
		Desarrollo	Nuevo	Subir cambios a producción	1	1	1	DONE
		Pruebas	Nuevo	Pruebas funcionales v1,2	4	4	4	DONE
	Segunda Fase	Desarrollo	Nuevo	Elaboracion de Informes de las pruebas funcionales	4	4	4	DONE
		Desarrollo	Nuevo	Creación de imágenes Voto electrónico v1,2 Modulo Control	2	2	2	DONE
Desarrollo		Nuevo	Creación de métodos de acceso al sistema	2	2	2	DONE	
Análisis Sitio Web	Análisis	Análisis	Nuevo	Análisis de sitio web pruebas de Rendimiento	2	2	2	DONE
		Análisis	Nuevo	Elaborar informe resultante Tesis v1	3	3	3	DONE
		Análisis	Nuevo	Editar formato de informe de análisis de sitio web	2	2	2	DONE
MODULO CANDIDATOS	Investigación	Investigación	Nuevo	Investigación sobre requisitos para poder ser Candidatos y asignar una lista y proceso	2	2	2	DONE
		Investigación	Nuevo	Investigación de envio de resultados y cambios de contraseña mediante mail	2	2	2	DONE
CONSULTAS	Investigación	Investigación	Nuevo	Elaboración de informes con metodología Ágiles	3	3	3	DONE
		Investigación	Nuevo	Elaboración de documentos sprints y versionamiento	5	5	5	DONE
		Investigación	Nuevo	Consulta sobre metodos de ingreso de datos	2	2	2	DONE
		Investigación	Nuevo	Consulta de utilización de componentes	2	2	2	DONE
		Investigación	Nuevo	Consulta sobre Wildfly y posibles fallos	2	2	2	DONE
General	Reuniones	Planificación	Nuevo	Reunión de Socialización Scrum, Retrospectiva Sprint	4	4	4	DONE
		Planificación	Nuevo	Reunión de planificación.	2	2	2	DONE
		Revisión	Nuevo	Reunión de Revisión de desarrollos.	2	2	2	DONE
	Logística en territorio	Territorio	Nuevo	Logística en territorio.	10	10	10	DONE
	Tareas no planificadas				0	0	0	
<b>TOTAL</b>					91	91	91	

## PLANIFICACIÓN DE TRABAJOS DE DESARROLLO

<b>Development:</b>	Diego Rea								
<b>Coordinador:</b>	Antonio Quiña	Mauricio Rea							
<b>Sprint</b>	6								
<b>Fechas</b>	21/11/2016								
	09/12/2016								
<b>Total horas</b>	91								

PROYECTO	ACTIVIDAD	Fase Desarrollo	Tipo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO SUGERIDO (Horas)	TIEMPO REAL (Horas)	ESTADO
VOTO ELECTRO	Ambiente de producción	Desarrollo	Nuevo	Cambios en base de datos	3	3	3	DONE
		Desarrollo	Nuevo	Constante Limpiezas de proyecto y codigo redundante	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas funcionales v2,1	2	2	2	DONE
VOTO ELECTRO	Reportes	Desarrollo	Nuevo	Reportes actualizados para Consultor y Administrador de Listas	2	2	2	DONE
		Desarrollo	Nuevo	Ingreso en bitácora de Cosultores, y Administradores	2	2	2	DONE
		Desarrollo	Nuevo	Vincular desarrollo con la v1,2	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas funcionales v1,2	2	2	2	DONE
	Gestión de Ingreso	Desarrollo	Nuevo	Creación de módulo de Consultor	6	6	6	DONE
		Desarrollo	Nuevo	Creacion de módulo Administrador	2	2	2	DONE
		Desarrollo	Nuevo	Creación de módulo de Persona Votnte	2	2	2	DONE
		Desarrollo	Nuevo	Registro de Candidatos	2	2	2	DONE
		Desarrollo	Nuevo	Cambios en base de datos	2	2	2	DONE
		Desarrollo	Nuevo	Edición de imágenes de Candidatos	2	2	2	DONE
		Desarrollo	Nuevo	Subir cambios a producción	1	1	1	DONE
		Pruebas	Nuevo	Pruebas funcionales v1,2	4	4	4	DONE
	Segunda Fase	Desarrollo	Nuevo	Elaboracion de Informes de las pruebas funcionales	4	4	4	DONE
		Desarrollo	Nuevo	Creación de imágenes Voto electrónico v1,2 Modulo Control	2	2	2	DONE
Desarrollo		Nuevo	Creación de métodos de acceso al sistema	2	2	2	DONE	
Análisis Sitio Web	Análisis	Análisis	Nuevo	Análisis de sitio web pruebas de Rendimiento	2	2	2	DONE
		Análisis	Nuevo	Elaborar informe resultante Tesis v1	3	3	3	DONE
		Análisis	Nuevo	Editar formato de informe de análisis de sitio web	2	2	2	DONE
MODULO CANDIDATOS	Investigación	Investigación	Nuevo	Investigación sobre requisitos para poder ser Candidatos y asignar una lista y proceso	2	2	2	DONE
		Investigación	Nuevo	Investigación de envio de resultados y cambios de contraseña mediante mail	2	2	2	DONE
CONSULTAS	Investigación	Investigación	Nuevo	Elaboración de informes con metodología Ágiles	3	3	3	DONE
		Investigación	Nuevo	Elaboración de documentos sprints y versionamiento	5	5	5	DONE
		Investigación	Nuevo	Consulta sobre metodos de ingreso de datos	2	2	2	DONE
		Investigación	Nuevo	Consulta de utilización de componentes	2	2	2	DONE
		Investigación	Nuevo	Consulta sobre Wildfly y posibles fallos	2	2	2	DONE
General	Reuniones	Planificación	Nuevo	Reunión de Socialización Scrum, Retrospectiva Sprint	4	4	4	DONE
		Planificación	Nuevo	Reunión de planificación.	2	2	2	DONE
		Revisión	Nuevo	Reunión de Revisión de desarrollos.	2	2	2	DONE
	Logística en territorio	Territorio	Nuevo	Logística en territorio.	10	10	10	DONE
	Tareas no planificadas				0	0	0	
<b>TOTAL</b>					91	91	91	

## PLANIFICACIÓN DE TRABAJOS DE DESARROLLO

<b>Development:</b>	Diego Rea								
<b>Coordinador:</b>	Antonio Quiña	Mauricio Rea							
<b>Sprint</b>	7								
<b>Fechas</b>	12/12/2016								
	06/01/2017								
<b>Total horas</b>	91								

PROYECTO	ACTIVIDAD	Fase Desarrollo	Tipo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO SUGERIDO (Horas)	TIEMPO REAL (Horas)	ESTADO
VOTO ELECTRO	Ambiente de producción	Desarrollo	Nuevo	Cambios en base de datos	3	3	3	DONE
		Desarrollo	Nuevo	Constante Limpiezas de proyecto y codigo redundante	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas funcionales v2,1	2	2	2	DONE
VOTO ELECTRO	Reportes	Desarrollo	Nuevo	Reportes actualizados para Consultor y Administrador de Listas	2	2	2	DONE
		Desarrollo	Nuevo	Ingreso en bitácora de Cosultores, y Administradores	2	2	2	DONE
		Desarrollo	Nuevo	Vincular desarrollo con la v1,2	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas funcionales v1,2	2	2	2	DONE
	Gestión de Ingreso	Desarrollo	Nuevo	Creación de módulo de Consultor	6	6	6	DONE
		Desarrollo	Nuevo	Creacion de módulo Administrador	2	2	2	DONE
		Desarrollo	Nuevo	Creación de módulo de Persona Votnte	2	2	2	DONE
		Desarrollo	Nuevo	Registro de Candidatos	2	2	2	DONE
		Desarrollo	Nuevo	Cambios en base de datos	2	2	2	DONE
		Desarrollo	Nuevo	Edición de imágenes de Candidatos	2	2	2	DONE
		Desarrollo	Nuevo	Subir cambios a producción	1	1	1	DONE
		Pruebas	Nuevo	Pruebas funcionales v1,2	4	4	4	DONE
	Segunda Fase	Desarrollo	Nuevo	Elaboracion de Informes de las pruebas funcionales	4	4	4	DONE
		Desarrollo	Nuevo	Creación de imágenes Voto electrónico v1,2 Modulo Control	2	2	2	DONE
Desarrollo		Nuevo	Creación de métodos de acceso al sistema	2	2	2	DONE	
Análisis Sitio Web	Análisis	Análisis	Nuevo	Análisis de sitio web pruebas de Rendimiento	2	2	2	DONE
		Análisis	Nuevo	Elaborar informe resultante Tesis v1	3	3	3	DONE
		Análisis	Nuevo	Editar formato de informe de análisis de sitio web	2	2	2	DONE
MODULO CANDIDATOS	Investigación	Investigación	Nuevo	Investigación sobre requisitos para poder ser Candidatos y asignar una lista y proceso	2	2	2	DONE
		Investigación	Nuevo	Investigación de envio de resultados y cambios de contraseña mediante mail	2	2	2	DONE
CONSULTAS	Investigación	Investigación	Nuevo	Elaboración de informes con metodología Ágiles	3	3	3	DONE
		Investigación	Nuevo	Elaboración de documentos sprints y versionamiento	5	5	5	DONE
		Investigación	Nuevo	Consulta sobre metodos de ingreso de datos	2	2	2	DONE
		Investigación	Nuevo	Consulta de utilización de componentes	2	2	2	DONE
		Investigación	Nuevo	Consulta sobre Wildfly y posibles fallos	2	2	2	DONE
General	Reuniones	Planificación	Nuevo	Reunión de Socialización Scrum, Retrospectiva Sprint	4	4	4	DONE
		Planificación	Nuevo	Reunión de planificación.	2	2	2	DONE
		Revisión	Nuevo	Reunión de Revisión de desarrollos.	2	2	2	DONE
	Logística en territorio	Territorio	Nuevo	Logística en territorio.	10	10	10	DONE
	Tareas no planificadas				0	0	0	
<b>TOTAL</b>					91	91	91	

## PLANIFICACIÓN DE TRABAJOS DE DESARROLLO

<b>Development:</b>	Diego Rea								
<b>Coordinador:</b>	Antonio Quiña	Mauricio Rea							
<b>Sprint</b>	8								
<b>Fechas</b>	09/01/2017								
	10/02/2017								
<b>Total horas</b>	92								

PROYECTO	ACTIVIDAD	Fase Desarrollo	Tipo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO SUGERIDO (Horas)	TIEMPO REAL (Horas)	ESTADO
VOTO ELECTRO	Ambiente de producción	Desarrollo	Nuevo	Cambios en base de datos	3	3	3	DONE
		Desarrollo	Nuevo	Constante Limpiezas de proyecto y codigo redundante	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas funcionales ejecutadas por tutor	2	2	2	DONE
VOTO ELECTRO	Reportes	Desarrollo	Nuevo	Reportes actualizados para Consultor y Administrador de Listas	2	2	2	DONE
		Desarrollo	Nuevo	Ingreso en bitácora de Cosultores, y Administradores	2	2	2	DONE
		Desarrollo	Nuevo	Vincular desarrollo con la v1,2	2	2	2	DONE
		Pruebas	Nuevo	Pruebas funcionales v1,2	2	2	2	DONE
	Gestión de Ingreso	Desarrollo	Nuevo	Creación de módulo de Consultor	6	6	6	DONE
		Desarrollo	Nuevo	Creacion de módulo Administrador	2	2	2	DONE
		Desarrollo	Nuevo	Creación de módulo de Persona Votnte	2	2	2	DONE
		Desarrollo	Nuevo	Registro de Candidatos	2	2	2	DONE
		Desarrollo	Nuevo	Cambios en base de datos	2	2	2	DONE
		Desarrollo	Nuevo	Edición de imágenes de Candidatos	2	2	2	DONE
		Desarrollo	Nuevo	Subir cambios a producción	1	1	1	DONE
		Pruebas	Nuevo	Pruebas funcionales v1,2	4	4	4	DONE
	Segunda Fase	Desarrollo	Nuevo	Elaboración de Oficios a entidad Auspiciante para pruebas	4	4	4	DONE
		Desarrollo	Nuevo	Preparación con imágenes institucionales	2	2	2	DONE
Desarrollo		Nuevo	Creación de métodos de acceso al sistema	2	2	2	DONE	
Análisis Sitio Web	Análisis	Análisis	Nuevo	Análisis de sitio web pruebas de Rendimiento	2	2	2	DONE
		Análisis	Nuevo	Elaborar informe resultante Tesis v1	3	3	3	DONE
		Análisis	Nuevo	Editar formato de informe de análisis de sitio web	2	2	2	DONE
MODULO CANDIDATOS	Investigación	Investigación	Nuevo	Investigación sobre requisitos para poder tener acceso a institucion Auspiciante	2	2	2	DONE
		Investigación	Nuevo	Elaboración de formatos prediseñados para la subida de archivos	2	2	2	DONE
CONSULTAS	Investigación	Investigación	Nuevo	Elaboración de formatos de subida de información a la base de datos	3	3	3	DONE
		Investigación	Nuevo	Elaboración de documentos sprints y versionamiento	5	5	5	DONE
		Investigación	Nuevo	Consulta sobre metodos de ingreso de datos	2	2	2	DONE
		Investigación	Nuevo	Consulta de utilización de protocolos y certificados	2	2	2	DONE
		Investigación	Nuevo	Consulta sobre disponibilidad de Servidores	2	2	2	DONE
General	Reuniones	Planificación	Nuevo	Reunión de Socialización Scrum, Retrospectiva Sprint	4	4	4	DONE
		Planificación	Nuevo	Reunión de planificación.	2	2	2	DONE
		Revisión	Nuevo	Reunión de Revisión de desarrollos.	2	2	2	DONE
	Logística en territorio	Territorio	Nuevo	Logística en territorio.	10	10	10	DONE
	Tareas no planificadas				0	0	0	
<b>TOTAL</b>					92	92	92	

## PLANIFICACIÓN DE TRABAJOS DE DESARROLLO

<b>Development:</b>	Diego Rea								
<b>Coordinador:</b>	Antonio Quiña	Mauricio Rea							
<b>Sprint</b>	9								
<b>Fechas</b>	11/02/2017								
	24/02/2017								
<b>Total horas</b>	92								

PROYECTO	ACTIVIDAD	Fase Desarrollo	Tipo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO SUGERIDO (Horas)	TIEMPO REAL (Horas)	ESTADO
VOTO ELECTRO	Ambiente de producción	Desarrollo	Nuevo	Cambios en base de datos	3	3	3	DONE
		Desarrollo	Nuevo	Constante Limpiezas de proyecto y codigo redundante	2	2	2	DONE
		Desarrollo	Nuevo	Pruebas funcionales ejecutadas por tutor	2	2	2	DONE
VOTO ELECTRO	Reportes	Desarrollo	Nuevo	Reportes actualizados para Consultor y Administrador de Listas	2	2	2	DONE
		Pruebas	Nuevo	Ingreso en bitácora de Cosultores, y Administradores	2	2	2	DONE
		Pruebas	Nuevo	Vincular desarrollo con la v1,2	2	2	2	DONE
		Pruebas	Nuevo	Pruebas funcionales v1,2	2	2	2	DONE
	Gestión de Ingreso	Pruebas	Nuevo	Creación de módulo de Consultor	6	6	6	DONE
		Pruebas	Nuevo	Creacion de módulo Administrador	2	2	2	DONE
		Finalizado	Nuevo	Creación de módulo de Persona Votante	2	2	2	DONE
		Pruebas	Nuevo	Registro de Candidatos	2	2	2	DONE
		Pruebas	Nuevo	Cambios en base de datos	2	2	2	DONE
		Finalizado	Nuevo	Edición de imágenes de Candidatos	2	2	2	DONE
		Pruebas	Nuevo	Subir cambios a producción	1	1	1	DONE
	Segunda Fase	Pruebas	Nuevo	Pruebas funcionales v1,2	4	4	4	DONE
		Finalizado	Nuevo	Elaboración de Oficios a entidad Auspiciante para pruebas	4	4	4	DONE
Pruebas		Nuevo	Preparación con imágenes institucionales	2	2	2	DONE	
Análisis Sitio Web	Análisis	Desarrollo	Nuevo	Creación de métodos de acceso al sistema	2	2	2	DONE
		Análisis	Nuevo	Análisis de sitio web pruebas de Rendimiento	2	2	2	DONE
		Análisis	Nuevo	Elaborar informe resultante Tesis v1	3	3	3	DONE
MODULO CANDIDATOS	Investigación	Análisis	Nuevo	Editar formato de informe de análisis de sitio web	2	2	2	DONE
		Investigación	Nuevo	Investigación sobre requisitos para poder tener acceso a institucion Auspiciante	2	2	2	DONE
CONSULTAS	Investigación	Investigación	Nuevo	Elaboración de formatos prediseñados para la subida de archivos	2	2	2	DONE
		Investigación	Nuevo	Elaboración de formatos de subida de información a la base de datos	3	3	3	DONE
		Investigación	Nuevo	Elaboración de documentos sprints y versionamiento	5	5	5	DONE
		Investigación	Nuevo	Consulta sobre metodos de ingreso de datos	2	2	2	DONE
		Investigación	Nuevo	Consulta de utilización de protocolos y certificados	2	2	2	DONE
General	Reuniones	Investigación	Nuevo	Consulta sobre disponibilidad de Servidores	2	2	2	DONE
		Investigación	Nuevo	Investigación sobre VPN en cliente	5	5	5	DONE
		Planificación	Nuevo	Reunión de Socialización Scrum, Retrospectiva Sprint	4	4	4	DONE
	Logística en territorio	Planificación	Nuevo	Reunión de planificación.	2	2	2	DONE
Revisión		Nuevo	Reunión de Revisión de desarrollos.	2	2	2	DONE	
Tareas no planificadas	Territorio	Nuevo	Logística en territorio.	10	10	10	DONE	
<b>TOTAL</b>					92	92	92	

## PLANIFICACIÓN DE TRABAJOS DE DESARROLLO

<b>Development:</b>	Diego Rea								
<b>Coordinador:</b>	Antonio Quiña	Mauricio Rea							
<b>Sprint</b>	10								
<b>Fechas</b>	27/02/2017								
	07/03/2017								
<b>Total horas</b>	87								

PROYECTO	ACTIVIDAD	Fase Desarrollo	Tipo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO SUGERIDO (Horas)	TIEMPO REAL (Horas)	ESTADO
VOTO ELECTRO	Ambiente de producción	Finalizado	Nuevo	Revisión de Base de dato	3	3	3	DONE
		Finalizado	Nuevo	Limpiezas de proyecto y codigo redundante	2	2	2	DONE
		Finalizado	Nuevo	Pruebas funcionales ejecutadas por tutor	2	2	2	DONE
VOTO ELECTRO	Reportes	Finalizado	Nuevo	Reportes actualizados para Consultor y Administrador de Listas	2	2	2	DONE
		Finalizado	Nuevo	Ingreso en bitácora de Cosultores, y Administradores	2	2	2	DONE
		Finalizado	Nuevo	Puesta a punto de Servidor previa produccion	2	2	2	DONE
		Finalizado	Nuevo	Pruebas funcionales v1,3	2	2	2	DONE
	Gestión de Ingreso	Finalizado	Nuevo	Creación de módulo de Consultor	6	6	6	DONE
		Finalizado	Nuevo	Creacion de módulo Administrador	2	2	2	DONE
		Finalizado	Nuevo	Creación de módulo de Persona Votante	2	2	2	DONE
		Finalizado	Nuevo	Registro de Candidatos	2	2	2	DONE
		Finalizado	Nuevo	Cambios en base de datos	2	2	2	DONE
		Finalizado	Nuevo	Edición de imágenes de Candidatos	2	2	2	DONE
		Finalizado	Nuevo	Subir cambios a producción	1	1	1	DONE
		Finalizado	Nuevo	Pruebas funcionales v1,3	4	4	4	DONE
	Segunda Fase	Finalizado	Nuevo	Elaboración de Resultados oficiales	4	4	4	DONE
		Desarrollo	Nuevo	Preparación de informe final	2	2	2	DONE
Desarrollo		Nuevo	Creación de métodos de acceso al sistema	2	2	2	DONE	
Análisis Sitio Web	Análisis	Análisis	Nuevo	Análisis de sitio web pruebas de Rendimiento	2	2	2	DONE
		Análisis	Nuevo	Elaborar informe resultante Tesis Final	3	3	3	DONE
		Análisis	Nuevo	Editar formato de informe de análisis de sitio web	2	2	2	DONE
MODULO CANDIDATOS	Investigación	Investigación	Nuevo	Investigación sobre requisitos para poder tener acceso a institucion Auspiciante	2	2	2	DONE
		Investigación	Nuevo	Elaboración de formatos prediseñados para la subida de archivos	2	2	2	DONE
CONSULTAS	Investigación	Investigación	Nuevo	Elaboración de formatos de Finalización scrum	3	3	3	DONE
		Investigación	Nuevo	Elaboración de documentos sprints y versionamiento	5	5	5	DONE
		Investigación	Nuevo	Consulta sobre metodos de ingreso de datos	2	2	2	DONE
		Investigación	Nuevo	Consulta de utilización de protocolos y certificados	2	2	2	DONE
		Investigación	Nuevo	Consulta sobre disponibilidad de Servidores	2	2	2	DONE
General	Reuniones	Planificación	Nuevo	Reunión de Socialización Scrum, Retrospectiva Sprint	4	4	4	DONE
		Planificación	Nuevo	Reunión de planificación.	2	2	2	DONE
		Revisión	Nuevo	Reunión de Revisión de desarrollos.	2	2	2	DONE
	Logística en territorio	Territorio	Nuevo	Logística en territorio.	10	10	10	DONE
	Tareas no planificadas				0	0	0	
<b>TOTAL</b>					87	87	87	

## PLANIFICACIÓN DE TRABAJOS DE DESARROLLO

PROYECTO	ACTIVIDAD	Fase Desarrollo	Tipo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO SUGERIDO (Horas)	TIEMPO REAL (Horas)	ESTADO
<b>Development:</b> Diego Rea <b>Coordinador:</b> Antonio Quiña      Mauricio Rea <b>Sprint:</b> 11 <b>Fechas:</b> 08/03/2017 10/06/2017 <b>Total horas:</b> 87								
VOTO ELECTRO	Ambiente de producción	Finalizado	Nuevo	Revisión de Base de dato	3	3	3	DONE
		Finalizado	Nuevo	Limpiezas de proyecto y codigo redundante	2	2	2	DONE
		Finalizado	Nuevo	Pruebas funcionales ejecutadas por tutor	2	2	2	DONE
VOTO ELECTRO	Reportes	Finalizado	Nuevo	Reportes actualizados para Consultor y Administrador de Listas	2	2	2	DONE
		Finalizado	Nuevo	Ingreso en bitácora de Cosultores, y Administradores	2	2	2	DONE
		Finalizado	Nuevo	Puesta a punto de Servidor previa produccion	2	2	2	DONE
		Finalizado	Nuevo	Pruebas funcionales v1,3	2	2	2	DONE
	Gestión de Ingreso	Finalizado	Nuevo	Creación de módulo de Consultor	6	6	6	DONE
		Finalizado	Nuevo	Creacion de módulo Administrador	2	2	2	DONE
		Finalizado	Nuevo	Creación de módulo de Persona Votante	2	2	2	DONE
		Finalizado	Nuevo	Registro de Candidatos	2	2	2	DONE
		Finalizado	Nuevo	Cambios en base de datos	2	2	2	DONE
		Finalizado	Nuevo	Edición de imágenes de Candidatos	2	2	2	DONE
		Finalizado	Nuevo	Subir cambios a producción	1	1	1	DONE
	Segunda Fase	Finalizado	Nuevo	Pruebas funcionales v1,3	4	4	4	DONE
		Finalizado	Nuevo	Elaboración de Resultados oficiales	4	4	4	DONE
Finalizado		Nuevo	Preparación de informe final	2	2	2	DONE	
Análisis Sitio Web	Análisis	Finalizado	Nuevo	Entrega Final del Proyecto a autoridades del plantel	2	2	2	DONE
		Análisis	Nuevo	Análisis de sitio web pruebas de Rendimiento	2	2	2	DONE
		Análisis	Nuevo	Elaborar informe resultante Tesis Final	3	3	3	DONE
MODULO CANDIDATOS	Investigación	Análisis	Nuevo	Editar formato de informe de análisis de sitio web	2	2	2	DONE
		Investigación	Nuevo	Investigación sobre requisitos para poder tener acceso a institucion Auspiciante	2	2	2	DONE
CONSULTAS	Investigación	Investigación	Nuevo	Elaboración de formatos prediseñados para la subida de archivos	2	2	2	DONE
		Investigación	Nuevo	Elaboración de formatos de Finalización scrum	3	3	3	DONE
		Investigación	Nuevo	Elaboración de documentos sprints y versionamiento	5	5	5	DONE
		Investigación	Nuevo	Consulta sobre metodos de ingreso de datos	2	2	2	DONE
		Investigación	Nuevo	Consulta de utilización de protocolos y certificados	2	2	2	DONE
General	Reuniones	Investigación	Nuevo	Consulta sobre disponibilidad de Servidores	2	2	2	DONE
		Planificación	Nuevo	Reunión de Socialización Scrum, Retrospectiva Sprint	4	4	4	DONE
		Planificación	Nuevo	Reunión de planificación.	2	2	2	DONE
	Revisión	Nuevo	Reunión de Revisión de desarrollos.	2	2	2	DONE	
	Logística en territorio	Territorio	Nuevo	Logística en territorio.	10	10	10	DONE
	Tareas no planificadas				0	0	0	
<b>TOTAL</b>					<b>87</b>	<b>87</b>	<b>87</b>	



## Desarrollo por meses del proyecto

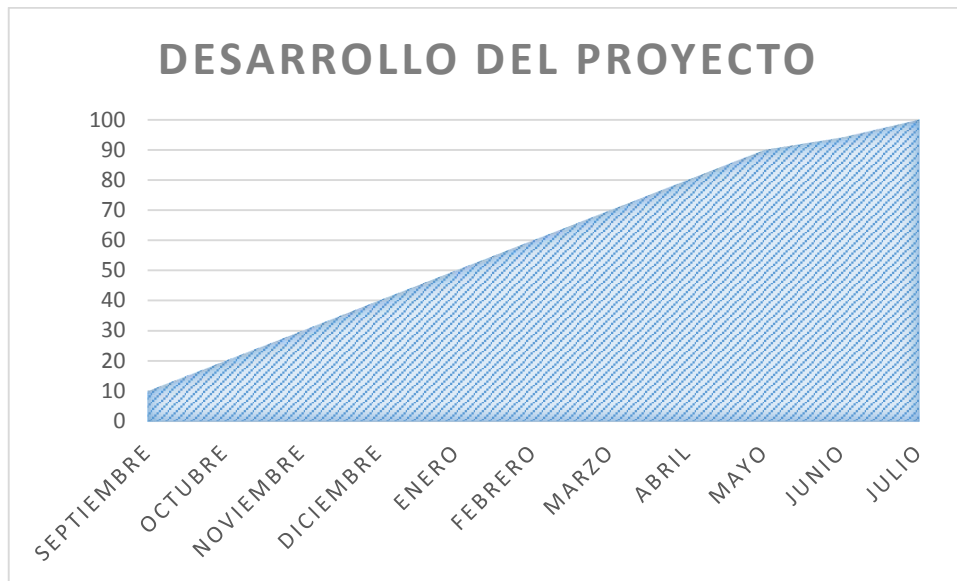


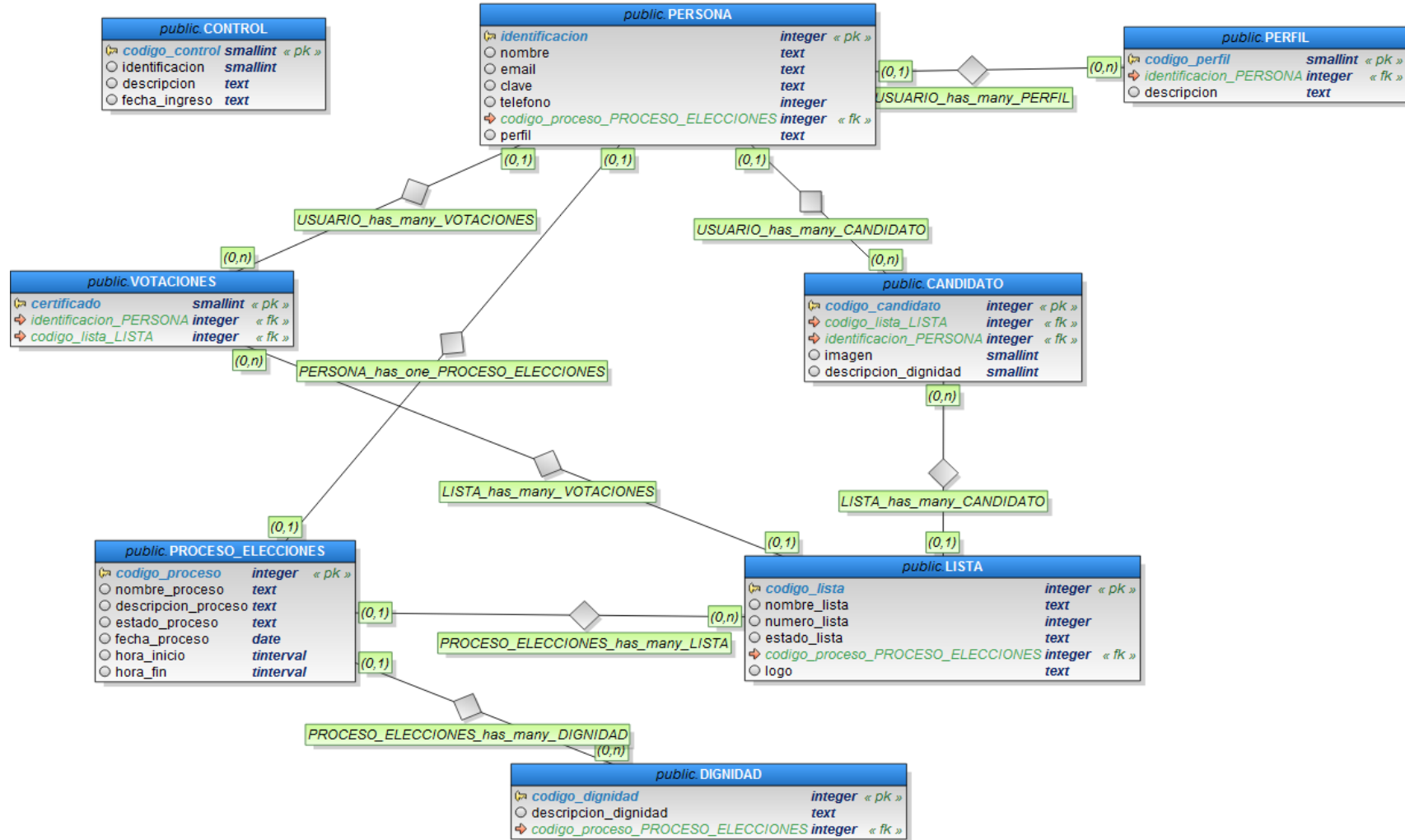
Figura 3.1. 2

Fuente: Propia

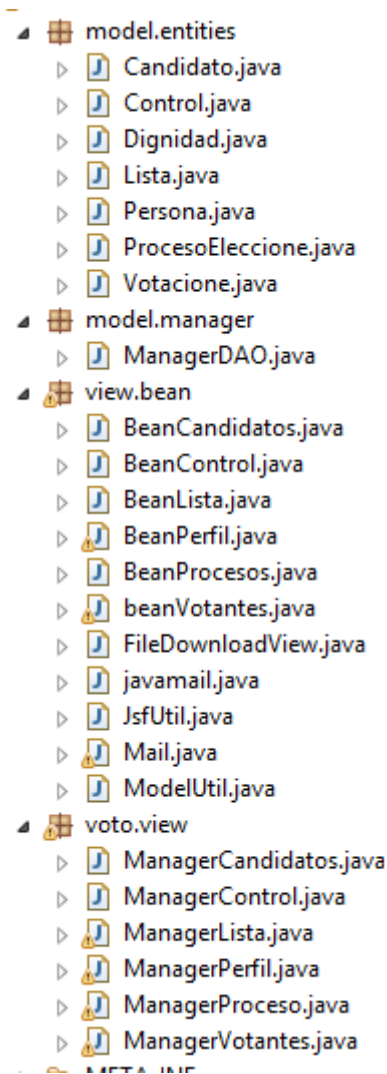
### 3.4 Codificación

La codificación aquí presentada, es Java-Primefaces, utilizando como herramienta para desarrollo a eclipse mars, nuestro servidor de aplicaciones es Wildfly en su versión 9.0, como gestor de base de datos PostgreSQL, a continuación se detalla los métodos que se emplearon para el buen desarrollo del sistema.

## DISEÑO DE BASE DE DATOS



## Codificación del Proyecto Voto Tesis



## Clase ManagerDAO

```
@Stateless(mappedName = "managerDAO")
@LocalBean
public class ManagerDAO {

    @PersistenceContext(unitName = "VotoTesis")
    private EntityManager em;
    @SuppressWarnings("rawtypes")

    public List findAll(Class clase, String orderBy) throws Exception {
        Query q;
        List listado;
        String sentenciaSQL;

        if (orderBy == null || orderBy.length() == 0)
```

```

        sentenciaSQL = "SELECT o FROM " + clase.getSimpleName() +
" o";
    else
        sentenciaSQL = "SELECT o FROM " + clase.getSimpleName()
+ " o ORDER BY " + orderBy;

    q = em.createQuery(sentenciaSQL);
    listado = q.getResultList();
    // mLog.MostrarLog(this.getClass(),"findAll",sentenciaSQL);
    return listado;
}

@SuppressWarnings("rawtypes")

public List findAll(Class clase) throws Exception {
    System.out.println(clase);
    return findAll(clase, null);
}

@SuppressWarnings({ "rawtypes", "unchecked" })

public Object findById(Class clase, Object pID) throws Exception {
    if (pID == null)
        throw new Exception(
            "Debe especificar el código para buscar el
dato.");
    Object o;
    try {
        o = em.find(clase, pID);
    } catch (Exception e) {
        e.printStackTrace();
        throw new Exception("Error al buscar la información
especificada ("
            + pID + ") : " + e.getMessage());
    }
    return o;
}

@SuppressWarnings("rawtypes")

public List findWhere(Class clase, String pClausulaWhere, String
pOrderBy)
    throws Exception {
    // mostrarLog("findAll(where): "+clase.getSimpleName());
    Query q;
    List listado;
    String sentenciaSQL;

    if (pOrderBy == null || pOrderBy.length() == 0)
        sentenciaSQL = "SELECT o FROM " + clase.getSimpleName()
+ " o WHERE " + pClausulaWhere;

    else
        sentenciaSQL = "SELECT o FROM " + clase.getSimpleName()
+ " o WHERE " + pClausulaWhere + " ORDER BY "
+ pOrderBy;

```

```

        q = em.createQuery(sentenciaSQL);
        listado = q.getResultList();
        //
managerLog.MostrarLog(this.getClass(),"findWhere",sentenciaSQL);
        return listado;
    }

    public void insertar(Object pObjeto) throws Exception {
        if (pObjeto == null)
            throw new Exception("No se puede insertar un objeto
null.");
        try {
            em.persist(pObjeto);
            // mostrarLog("Objeto insertado:
"+pObjeto.getClass().getSimpleName());
            System.out.println("ok");
        } catch (Exception e) {
            // mostrarLog("No se pudo insertar el objeto especificado:
"+pObjeto.getClass().getSimpleName());
            throw new Exception("No se pudo insertar el objeto
especificado ("
                                + pObjeto.getClass().getCanonicalName() + ")
"
                                + e.getMessage());
        }
    }

    @SuppressWarnings("rawtypes")

    public void eliminar(Class clase, Object pID) throws Exception {
        if (pID == null) {
            // mostrarLog("Debe especificar un identificador para
eliminar el dato solicitado: "+clase.getSimpleName());
            throw new Exception(
                "Debe especificar un identificador para
eliminar el dato solicitado.");
        }
        Object o = findById(clase, pID);

        try {
            em.remove(o);
            // mostrarLog("Dato eliminado: "+clase.getSimpleName()+" "
            // +pID.toString());
        } catch (Exception e) {
            // mostrarLog("No se pudo eliminar el dato:
"+clase.getSimpleName());
            throw new Exception("No se pudo eliminar el dato (" + pID
+ ")("
                                + clase.getCanonicalName() + ") : " +
e.getMessage());
        }
    }

    public void actualizar(Object pObjeto) throws Exception {
        if (pObjeto == null)

```

```

        throw new Exception("No se puede actualizar un objeto
null");
        /*
        * if(!em.getTransaction().isActive())
em.getTransaction().begin();
        */
        try {
            em.merge(pObjeto);

        } catch (Exception e) {
            // em.getTransaction().rollback();
            throw new Exception("No se pudo actualizar el dato
solicitado ("
                                + pObjeto.getClass().getCanonicalName() + "):
"
                                + e.getMessage());
        }
    }
}

```

### Clase ManagerProceso

```

@Stateless(mappedName = "managerProceso")
@LocalBean
public class ManagerProceso {

    @EJB
    ManagerDAO managerDAO;

    public ManagerProceso() {

    }

    @SuppressWarnings("unchecked")

    public List<ProcesoEleccion> findAllProcesos() throws Exception
    {
        List<ProcesoEleccion> result =
managerDAO.findAll(ProcesoEleccion.class);
        return result;
    }

    public List<SelectItem> findAllProcesos() throws Exception
    {
        ArrayList<SelectItem> itemproc = new ArrayList<SelectItem>();
        itemproc.clear();
        List<ProcesoEleccion> result =
managerDAO.findAll(ProcesoEleccion.class);
        for (ProcesoEleccion procesoEleccion : result) {
            SelectItem(itemproc,procesoEleccion.getCodigoProceso(),procesoEleccion.getNombreProceso());
        }
        return itemproc;
    }

    public void crearProcesos(ProcesoEleccion proceso) throws Exception
    {

```

```

        managerDAO.insertar(proceso);
    }

    public void actualizarProcesos(ProcesoEleccione proceso) throws
Exception
    {
        ProcesoEleccione proceso1 = (ProcesoEleccione)
managerDAO.findById(ProcesoEleccione.class, "2");
        proceso1.setNombreProceso(proceso.getNombreProceso());
        managerDAO.insertar(proceso);
    }
    public void eliminarProcesos(Integer CodigoProceso) throws Exception {
        managerDAO.eliminar(ProcesoEleccione.class, CodigoProceso );
    }
}

```

### Clase BeanProcesos

```

@SessionScoped
@ManagedBean

public class BeanProcesos {
    @EJB
    ManagerProceso managerProceso;
    List<ProcesoEleccione> procesoElecciones;
    List<SelectItem> procesoEleccioness;
    private int codigoProceso;
    private String nombreProceso;
    private String descripcionProceso;
    private String estadoProceso;
    private Date fechaProceso;
    private Time HoraInicio;
    private Time HoraFin;
    @ManagedProperty(value="#{beanVotantes}")
    private beanVotantes beanVotantes;
    public void setBeanVotantes(beanVotantes beanVotantes) {
        this.beanVotantes = beanVotantes;
    }
    public String getDescripcionProceso() {

        return descripcionProceso;
    }
    public void setDescripcionProceso(String descripcionProceso) {
        this.descripcionProceso = descripcionProceso;
    }
    public String getEstadoProceso() {
        return estadoProceso;
    }
    public void setEstadoProceso(String estadoProceso) {
        this.estadoProceso = estadoProceso;
    }
    public Date getFechaProceso() {
        return fechaProceso;
    }
    public void setFechaProceso(Date fechaProceso) {
        this.fechaProceso = fechaProceso;}
}

```

```

public Time getHoraInicio() {
    return HoraInicio;
}

public void setHoraInicio(Time horaInicio) {
    HoraInicio = horaInicio;
}

public Time getHoraFin() {
    return HoraFin;
}

public void setHoraFin(Time horaFin) {
    HoraFin = horaFin;
}

private ProcesoEleccion selected;

public ProcesoEleccion getSelected() {
    return selected;
}

public void setSelected(ProcesoEleccion selected) {
    this.selected = selected;
}

@PostConstruct
public void Iniciar() {
    actionCargarProcesos();
    actionCargarProcesosVotantes();
    beanVotantes.comprobrsession();
}

public BeanProcesos() {
}

public void actionCargarProcesos() {
    try {
        procesoElecciones = managerProceso.findAllProcesos();
    } catch (Exception e) {

        e.printStackTrace();
    }
}

public void actionCargarProcesosVotantes() {
    try {
        procesoEleccioness = managerProceso.findAllProcesoss();
    } catch (Exception e) {

        e.printStackTrace();
    }
}

```



```

    public String actionGuardarProcesos() {
        try {
            ProcesoEleccione proceso = new ProcesoEleccione();
            System.out.println(codigoProceso + " " + nombreProceso+ "
" + descripcionProceso+ " " + estadoProceso+ " " + fechaProceso+ " " +
HoraInicio);

            proceso.setNombreProceso(nombreProceso);
            proceso.setDescripcionProceso(descripcionProceso);
            proceso.setEstadoProceso(estadoProceso);
            proceso.setFechaProceso(fechaProceso);
            proceso.setHoraInicio(HoraInicio);
            managerProceso.crearProcesos(proceso);
            actionCargarProcesos();
            JsفUtil.addSuccessMessage("Proceso Electoral Agregado con
EXITO");

            return "index";

        } catch (Exception e) {

            e.printStackTrace();
            JsفUtil.addSuccessMessage(e.getMessage());
            return "";

        }
    }

    public void editarProceso(){
        try{
            managerProceso.actualizarProcesos(selected);

        }catch(Exception e){

        }

    }

    public void eliminarProceso(){
        try{

            managerProceso.eliminarProcesos(selected.getCodigoProceso());
            actionCargarProcesos();
            JsفUtil.addSuccessMessage("Proceso Electoral Eliminado con
EXITO");

        }catch(Exception e){
            e.printStackTrace();
            JsفUtil.addSuccessMessage(e.getMessage());

        }

    }

    public List<ProcesoEleccione> getProcesoElecciones() {
        return procesoElecciones;
    }

```

```

        public void setProcesoElecciones(List<ProcesoEleccion>
procesoElecciones) {
            this.procesoElecciones = procesoElecciones;
        }

        public int getCodigoProceso() {
            return codigoProceso;
        }

        public void setCodigoProceso(int codigoProceso) {
            this.codigoProceso = codigoProceso;
        }

        public String getNombreProceso() {
            return nombreProceso;
        }
        public void setNombreProceso(String nombreProceso) {
            this.nombreProceso = nombreProceso;
        }
        public List<SelectItem> getProcesoEleccioness() {
            return procesoEleccioness;
        }
        public void setProcesoEleccioness(List<SelectItem> procesoEleccioness)
{
            this.procesoEleccioness = procesoEleccioness;
        }
}

```

### 3.5 Pruebas del Sistema

Las pruebas del sistema se las realizó el jueves 13 de Julio del 2017 en la unidad educativa “Ciudad de Ibarra”, culminando con éxito votaciones de elección popular en los 3eros. De bachillerato “Informática” y “Electrónica”.

En esa oportunidad estuvieron presentes dos delegados de la unidad educativa “Ciudad de Ibarra” además de la autoridad máxima que se encargó de la parte del módulo de consultoría y proclamar resultados oficiales y finales.

En anexos se encuentra las evidencias de la actividad realizada en la institución antes mencionada además de un certificado con firmas de responsabilidad, y detallando actividades realizadas en dicho plantel.

### 3.6 Implementación

Para la implementación se ha creado un dominio denominado:

<https://votoelectrocrypecc.tk>

Este dominio antes mencionado se encuentra certificado por el algoritmo criptográfico ECDSA, en donde la encriptación cliente y servidor se la realiza utilizando los protocolos de curvas elípticas para comprobar se puede mirar el certificado generado en el navegador, que nos garantiza que está activo los protocolos.

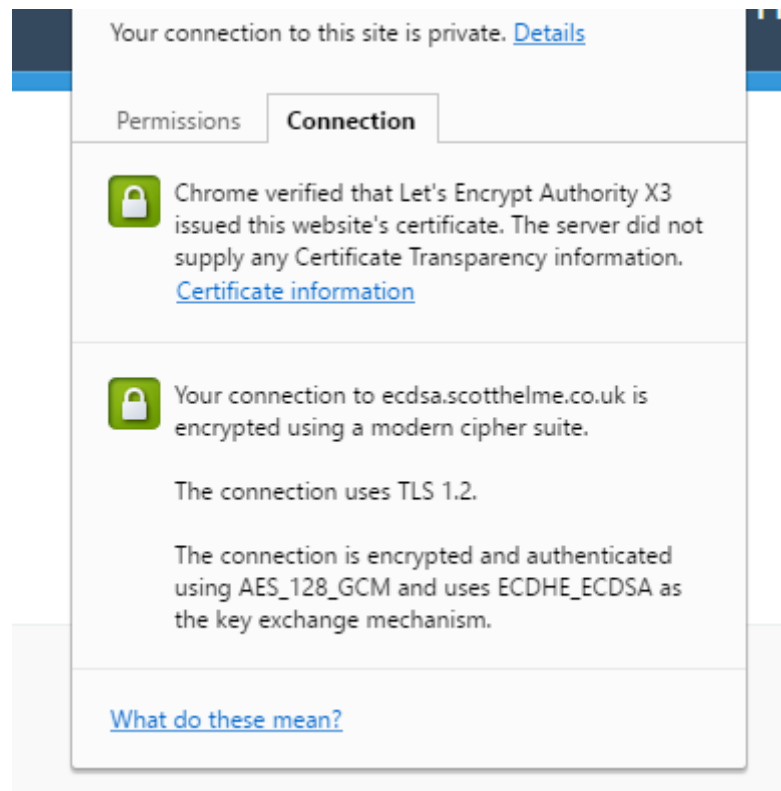


Figura 3. 6.1

Fuente: <http://www.evotoelectrocrypecc.tk> (Propia)

# **CAPÍTULO 4**

## **Conclusiones y Recomendaciones**

### **CONTENIDO DEL CAPÍTULO**

- 4.1 Conclusiones
- 4.2 Recomendaciones
- 4.3 Análisis de Impacto

#### **4.1 Conclusiones**

- La reducción en el tamaño de las claves tanto públicas como privadas, ofrece ventajas para la implementación segura en aplicaciones de tipo web mediante el uso de certificados de navegación, esto es:
- Después de la evaluación realizada al software se comprobó el correcto funcionamiento del mismo, y se dio por concluida la votación UECI- 2016 en la ciudad de Ibarra.
- Se puede usar el sistema de voto electrónico para elecciones populares de manera rápida y fiable.
- Se ha generado comprobantes de voto electrónico para cada votante, así garantizando mayor seguridad para el sistema.
- Los resultados oficiales son fiables para la publicación de los mismos.

#### **4.2 Recomendaciones**

- Se recomienda el uso de protocolo de curvas elípticas en aplicaciones web que necesiten incrementar el nivel de seguridad, y que disponga de información sensible.
- Se recomienda el uso de este trabajo de grado en futuras elecciones populares.

#### **4.3 Impacto Económico**

Una vez implementado el sistema de voto electrónico se ha podido evidenciar una notable mejora en el aspecto económico, y el principal impacto ha sido en la adquisición de hojas de papel, y por consiguiente una mejora en el aspecto ambiental.

Aquí se muestra una comparativa de impactos económicos que hubiese tenido de no ser por el sistema aquí implementado.

IMPACTO ECONÓMICO VOTO ELECTRÓNICO							
INDICADORES	NIVEL DE IMPACTO						
	-3	-2	-1	0	1	2	3
Costo de Licencias de Software					X		
Tiempo de implementación del sistema					X		
Hojas de Papel					X		
Impresiones						x	
TOTAL					3	2	

Nivel de impacto económico =  $\Sigma / \text{número de indicadores} = 5 / 4 = 1.25$

Nivel de impacto económico Voto Electrónico = Impacto medio positivo

IMPACTO ECONÓMICO VOTACIONES TRADICIONALES							
INDICADORES	NIVEL DE IMPACTO						
	-3	-2	-1	0	1	2	3
Tiempo de capacitación		X					
Hojas de Papel		X					
Impresiones		X					
Urnas		X					
Actas		X					
Personal		X					
TOTAL		-12					

Nivel de impacto económico =  $\Sigma / \text{número de indicadores} = -12 / 6 = -2$

Nivel de impacto económico Tradicional = Impacto negativo

Aquí se aprecia la notable diferencia al usar un sistema de voto tradicional y el impacto económico que tiene el usar el sistema de voto electrónico, las principales

diferencias es el tiempo y el personal necesario, al igual que material para la realización de las mismas.

#### 4.4 Impacto Ambiental

Este es uno de los impactos más notables, ya que en la actualidad se utiliza hojas recicladas para la única impresión necesaria, que es el comprobante de voto, el resto del proceso es amigable al medio ambiente.

IMPACTO AMBIENTAL VOTO ELECTRÓNICO							
INDICADORES	NIVEL DE IMPACTO						
	-3	-2	-1	0	1	2	3
Uso de Papel						X	
Energía Eléctrica						X	
Tratamiento Reciclaje						X	
<b>TOTAL</b>						6	

$$\text{Nivel de impacto económico} = \Sigma / \text{número de indicadores} = 6 / 3 = 2$$

$$\text{Nivel de impacto económico} = \text{Impacto medio positivo}$$

#### 4.5 Impacto Social

Una vez que el proceso ha llegado a culminarse con éxito, el impacto social que se ha detectado es determinante el positivismo y aprobatorio al uso del mismo.

IMPACTO SOCIAL VOTO ELECTRÓNICO							
INDICADORES	NIVEL DE IMPACTO						
	-3	-2	-1	0	1	2	3
Proyección de imagen social							X
Institución socialmente responsable						X	

<b>Información Oportuna Acerca de resultados</b>						X	
<b>Fiabilidad de los resultados</b>							X
<b>Reducción de tiempo en capacitaciones</b>							X
<b>Reducción de tiempo en el proceso de votación</b>							X
<b>Aceptación del Proyecto</b>							X
<b>TOTAL</b>						4	15

Nivel de impacto económico =  $\Sigma / \text{número de indicadores} = 19 / 7 = 2.71$   
Nivel de impacto económico = Impacto medio positivo

## Bibliografía

- Anonimo. (19 de Octubre de 2015). *www.loquemeinteresadelared.wordpress.com*. Obtenido de <https://loquemeinteresadelared.wordpress.com/2015/10/19/creacion-de-una-clave-publica-y-privada-con-openssl/>
- Byous, J. (21 de Abril de 2005). *www.wikipedia.com*. Recuperado el 2016, de [https://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))
- Data, S. (08 de 02 de 2016). <https://www.securitydata.net.ec>. Obtenido de [https://www.securitydata.net.ec/wp-content/downloads/listas/lista\\_precios.pdf](https://www.securitydata.net.ec/wp-content/downloads/listas/lista_precios.pdf)
- Duarte, E. R. (2009). *Curva Elípticas, hiperelítica y libcurve*. México: Criptored.
- Encrypt, L. (8 de 07 de 2016). <https://community.letsencrypt.org>. Obtenido de <https://community.letsencrypt.org/t/certificate-for-windows-server-2012-r2/17647>
- Gina Gallegos-García, M. A.-G. (1 de 04 de 2014). Obtenido de <http://www.revista.unam.mx>: <http://www.revista.unam.mx/vol.15/num4/art24/#>
- GlobalSign. (05 de 06 de 2015). <https://support.globalsign.com>. Obtenido de <https://support.globalsign.com/customer/portal/articles/2003943-generate-ecc-csr-windows-server-2008->
- Interactiva, A. (08 de 02 de 2011). <https://econianza.wordpress.com>. Obtenido de <https://econianza.wordpress.com/tag/curvas-elipticas/>
- Josep M. Miret Javier Valera & Magda Valls. (12 de Mayo de 2015). <http://www.criptored.upm.es/>. Recuperado el 4 de Octubre de 2016, de <http://www.criptored.upm.es/crypt4you/temas/ECC/leccion1/leccion1.html#12>
- Lang, J.-P. (09 de 09 de 2015). <https://wiki.strongswan.org>. Obtenido de <https://wiki.strongswan.org/projects/strongswan/wiki/WindowsSuiteB#1-Preparations>
- Lerma, E. V. (30 de Junio de 2010). *www.adictosaltrabajo.com*. Recuperado el 1 de Septiembre de 2016, de <https://www.adictosaltrabajo.com/tutoriales/introduccion-primefaces/>



- Madrid, U. P. (01 de 09 de 2015). <http://www.criptored.upm.e>. Obtenido de <http://www.criptored.upm.es/paginas/docencia.htm>
- Madrid, U. P. (03 de 09 de 2015). <http://www.criptored.upm.es>. Obtenido de <http://www.criptored.upm.es/crypt4you/temas/ECC/leccion2/leccion2.html#apartado24>
- Mateo, M. (12 de Septiembre de 2015). <http://html.rincondelvago.com/>. Obtenido de [http://html.rincondelvago.com/criptografia\\_seguridad-informatica.html](http://html.rincondelvago.com/criptografia_seguridad-informatica.html)
- Murcia, U. d. (21 de 10 de 2007). <http://www.um.es>. Obtenido de <http://www.um.es/docencia/barzana/IAGP/lagp10.html>
- Namecheap. (1 de 11 de 2016). <https://www.namecheap.com>. Obtenido de <https://www.namecheap.com/support/knowledgebase/article.aspx/9503/38/what-is-an-ecc-elliptic-curve-cryptography-certificate>
- NetoMeter. (12 de 04 de 2016). <https://www.netometer.com>. Obtenido de <https://www.netometer.com/blog/?p=1758>
- Pierini, G. (16 de Julio de 2014). [www.blogspot.com](http://www.blogspot.com). Recuperado el 1 de Septiembre de 2016, de <http://experienciasenjava.blogspot.com/2014/07/aplicaciones-web-en-java-primefaces.html>
- Rios, J. (22 de Septiembre de 2010). [www.monografias.com](http://www.monografias.com). Recuperado el 12 de Octubre de 2016, de <http://www.monografias.com/trabajos82/la-seguridad-informatica/la-seguridad-informatica2.shtml>
- Sanchez, L. G. (5 de Enero de 2016). <https://www.softeng.es/>. Recuperado el 25 de Octubre de 2016, de <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>
- SCHNEIDER, J. (30 de 08 de 2016). <http://blog.jonschneider.com>. Obtenido de <http://blog.jonschneider.com/2016/08/fix-ssl-handshaking-error-in-windows.html>
- Singh, C. (30 de 12 de 2014). <https://examples.javacodegeeks.com>. Obtenido de <https://examples.javacodegeeks.com/core-java/sql/import-csv-file-to-mysql-table-java-example/>
- Willis, R. (12 de 2 de 2016). <http://robwillis.info>. Obtenido de <http://robwillis.info/2016/02/iis-8-with-ecc-certificates-increasing-your-ssl-security-on-windows-server-2012/>

# **ANEXOS**



*Fotografía. Anexo 1*

*Fuente: Propia*

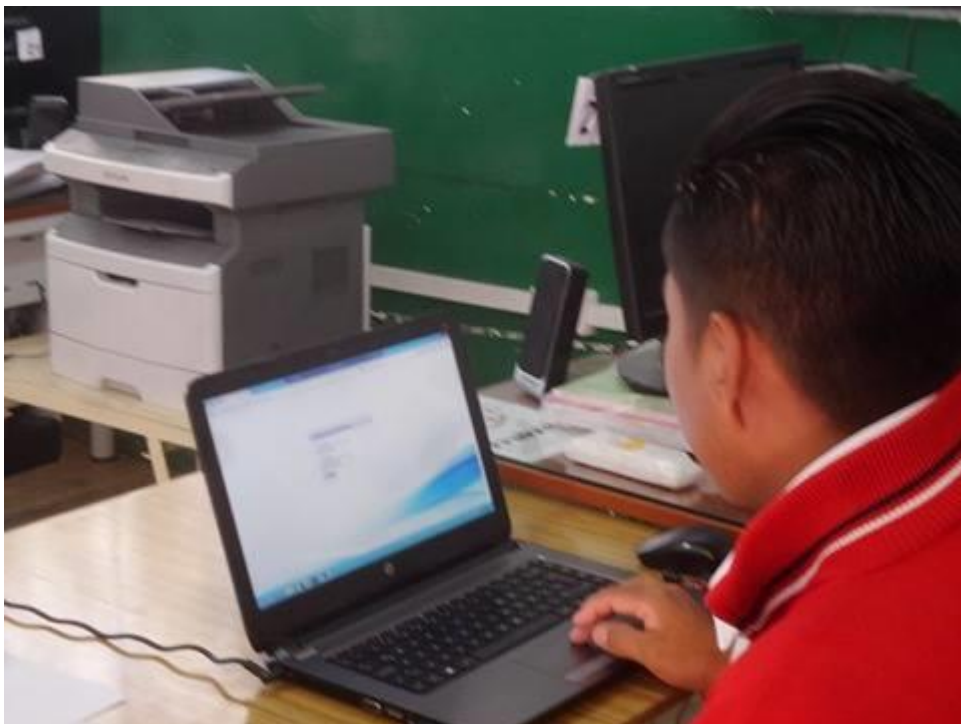
*Descripción: Estudiantes U.E.C.I. realizando una fila ordenada para poder ejercer su derecho al voto.*



*Fotografía. Anexo 2*

*Fuente: Propia*

*Descripción: Estudiante de la U.E.C.I. ingresando al Sistema de Voto Electrónico con su cédula en la mano.*



*Fotografía. Anexo 3*

*Fuente: Propia*

*Descripción: Estudiante de la U.E.C.I. ingresando al Sistema de Voto Electrónico.*



Fotografía. Anexo 4

Fuente: Propia

Descripción: Laboratorio de la U.E.C.I. con los estudiantes y tutores encargados de aplicar la votación electrónica.

## Codificación del Sistema de Voto Electrónico

### ManagerCandidatos

```
@Stateless(mappedName = "managerCandidatos")
@LocalBean

public class ManagerCandidatos {

    @EJB
    ManagerDAO managerDAO;

    public ManagerCandidatos() {
        // TODO Auto-generated constructor stub
    }

    @SuppressWarnings("unchecked")
    public List<Candidato> findAllProcesos() throws Exception
    {
        List<Candidato> result = managerDAO.findAll(Candidato.class);
        return result;
    }

    public void crearCandidato(Candidato proceso) throws Exception
```

```

    {
        managerDAO.insertar(proceso);
    }

    public void eliminarCandidato(Integer codigoCandidato) throws
Exception {
        managerDAO.eliminar(Candidato.class, codigoCandidato );
    }
}

```

## BeanCandidatos

```

@SessionScoped
@ManagedBean

```

```

public class BeanCandidatos {
    @EJB

    ManagerCandidatos managerCandidatos;
    ManagerLista managerlista;
    beanVotantes benvotantes;

    List<Candidato> candidatos;
    List<SelectItem> listadoListas;

    private int codigoCandidato;
    private int codigoLista;
    private int identificacion;
    private String imagen;
    private String dignidad;

    private Candidato selected;

    public void actionCargarListas() {
        try {
            listadoListas = managerlista.findNombreListas();
        } catch (Exception e) {

            e.printStackTrace();
        }
    }

    public List<Candidato> getCandidatos() {
        return candidatos;
    }

    public void setCandidatos(List<Candidato> candidatos) {
        this.candidatos = candidatos;
    }
}

```

```
public Candidato getSelected() {
    return selected;
}

public void setSelected(Candidato selected) {
    this.selected = selected;
}

public int getCodigoCandidato() {
    return codigoCandidato;
}

public void setCodigoCandidato(int codigoCandidato) {
    this.codigoCandidato = codigoCandidato;
}

public int getCodigoLista() {
    return codigoLista;
}

public void setCodigoLista(int codigoLista) {
    this.codigoLista = codigoLista;
}

public int getIdentificacion() {
    return identificacion;
}

public void setIdentificacion(int identificacion) {
    this.identificacion = identificacion;
}

public String getImagen() {
    return imagen;
}

public void setImagen(String imagen) {
    this.imagen = imagen;
}
}
```

```

public String getDignidad() {
    return dignidad;
}

public void setDignidad(String dignidad) {
    this.dignidad = dignidad;
}

@PostConstruct
public void Iniciar() {
    actionCargarCandidatos();
    // actionCargarListas();
}

public BeanCandidatos() {
    // TODO Auto-generated constructor stub
}

public void actionCargarListas() {
    try {
        listadоListas = managerlista.findNombreListas();
    } catch (Exception e) {

        e.printStackTrace();
    }
}

public void actionCargarCandidatos() {
    try {
        candidatos = managerCandidatos.findAllProcesos();
    } catch (Exception e) {

        e.printStackTrace();
    }
}

public void actionGuardarVotantes(int id) {
    try {
        Candidato candi = new Candidato();

        imagen="hola";

        System.out.println(id + " " + codigoLista+ " " + imagen
+" "+ dignidad );

        Lista l = new Lista();

        l.setCodigoLista(codigoLista);

        System.out.println(l.getCodigoLista());

```



```

        Persona p= new Persona();
        p.setIdentificacion(id);

        candi.setLista(l);
        candi.setImagen(imagen);
        candi.setDescripcionDignidad(dignidad);
        candi.setPersona(p);

        managerCandidatos.crearCandidato(candi);
        JsfUtil.addSuccessMessage("Agregado con EXITO");

    } catch (Exception e) {

        e.printStackTrace();
        JsfUtil.addSuccessMessage(e.getMessage());
    }
}

public void eliminarCandidato(){
    try{

        managerCandidatos.eliminarCandidato(selected.getCodigoCandidato());

        actionCargarCandidatos();;
        JsfUtil.addSuccessMessage("Lista Eliminada con EXITO");

    }catch(Exception e){
        e.printStackTrace();
        JsfUtil.addSuccessMessage(e.getMessage());
    }
}

public void handleFileUpload(FileUploadEvent event) {

    FacesMessage message = new FacesMessage("Successful",
event.getFile().getFileName() + " is uploaded.");
    FacesContext.getCurrentInstance().addMessage(null, message);

    imagen = event.getFile().getFileName();

    try {
        copyFile(event.getFile().getFileName(),
event.getFile().getInputStream());
    } catch (IOException e) {
        e.printStackTrace();
    }
}

private String destination =
"D:\\proyectos\\workspace\\VotoTesis\\WebContent\\imagenes\\fotos\\";

public void copyFile(String fileName, InputStream in) {

```

```

        try {
            // write the inputStream to a FileOutputStream
            OutputStream out = new FileOutputStream(new
File(destination + fileName));

            int read = 0;
            byte[] bytes = new byte[1024];

            while ((read = in.read(bytes)) != -1) {
                out.write(bytes, 0, read);
            }

            in.close();
            out.flush();
            out.close();

            System.out.println("New file created!");
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }
}

```

## BeanLista

```

public class BeanLista {
    @EJB

    ManagerLista managerLista;
    ManagerProceso managerProceso;
    ManagerDAO managerDAO;

    BeanProcesos b;
    List<Lista> ListadListas;
    List<SelectItem> procesoElecciones;
    List<SelectItem> nombreListas;

    private String nombreLista;
    private int nroLista;
    private String estadoLista;
    private Lista selected;
    private int codigo_lista;
    private int codigoproceso;
    private String logo;

    public void actionCargarProcesos() {

```

```

        try {
            procesoEleccioness = managerProceso.findAllProcesoss();
        } catch (Exception e) {

            e.printStackTrace();
        }
    }

    public int getCodigoproceso() {
        return codigoproceso;
    }

    public void setCodigoproceso(int codigoproceso) {
        this.codigoproceso = codigoproceso;
    }

    public int getCodigo_lista() {
        return codigo_lista;
    }

    public void setCodigo_lista(int codigo_lista) {
        this.codigo_lista = codigo_lista;
    }

    public Lista getSelected() {
        return selected;
    }

    public void setSelected(Lista selected) {
        this.selected = selected;
    }

    public List<Lista> getListadListas() {
        return ListadListas;
    }

    public void setListadListas(List<Lista> listadListas) {
        ListadListas = listadListas;
    }

    public String getNombreLista() {
        return nombreLista;
    }

    public void setNombreLista(String nombreLista) {
        this.nombreLista = nombreLista;
    }

    public int getNroLista() {
        return nroLista;
    }

    public void setNroLista(int nroLista) {
        this.nroLista = nroLista;
    }

    public String getEstadoLista() {

```

```

        return estadoLista;
    }

    public void setEstadoLista(String estadoLista) {
        this.estadoLista = estadoLista;
    }

    public List<SelectItem> getProcesoElecciones() {
        return procesoElecciones;
    }

    public void setProcesoElecciones(List<SelectItem> procesoElecciones)
    {
        this.procesoElecciones = procesoElecciones;
    }

    public List<SelectItem> getNombreListas() {
        return nombreListas;
    }

    public void setNombreListas(List<SelectItem> nombreListas) {
        this.nombreListas = nombreListas;
    }

    @PostConstruct
    public void Iniciar() {
        actionCargarListas();
        actionCargarNombredeListas();
    }

    public String getLogo() {
        return logo;
    }

    public void setLogo(String logo) {
        this.logo = logo;
    }

    public void actionCargarProcesosVotantes() {
        try {
            procesoElecciones = managerProceso.findAllProcesoss();
        } catch (Exception e) {

            e.printStackTrace();
        }
    }

    public BeanLista() {
    }

    public void actionCargarListas() {
        try {

            ListadListas = managerLista.findAllListas();
        } catch (Exception e) {

```

```

        e.printStackTrace();
    }
}

public void actionCargarNombredelistas(){
    try {
        nombreListas = managerLista.findNombreListas();
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

public String actionGuardarListas() {
    try {
        actionCargarListas();
        Lista procesoLi = new Lista();

        System.out.println(nombreLista + " " + nroLista + " " +
estadoLista + " " + logo+ " " + codigoproceso);

        procesoLi.setNombreLista(nombreLista);
        procesoLi.setNumeroLista(nroLista);
        procesoLi.setEstadoLista(estadoLista);
        procesoLi.setLogo(logo);

        ProcesoEleccion p = new ProcesoEleccion();
        p.setCodigoProceso(codigoproceso);

        System.out.println(p.getCodigoProceso());

        procesoLi.setProcesoEleccion(p);

        managerLista.crearListas(procesoLi);

        actionCargarListas();
        JsFUtil.addSuccessMessage("Lista Agregada con EXITO");
        return "index";
    } catch (Exception e) {

        e.printStackTrace();
        JsFUtil.addSuccessMessage(e.getMessage());
        return "";
    }
}

public void handleFileUpload(FileUploadEvent event) {

    FacesMessage message = new FacesMessage("Successful",
event.getFile().getFileName() + " is uploaded.");
    FacesContext.getCurrentInstance().addMessage(null, message);

    logo = event.getFile().getFileName();
}

```

```

        try {
            copyFile(event.getFile().getFileName(),
event.getFile().getInputStream());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private String destination =
"D:\\proyectos\\workspace\\VotoTesis\\WebContent\\imagenes\\";

    public void copyFile(String fileName, InputStream in) {
        try {

            // write the inputStream to a FileOutputStream
            OutputStream out = new FileOutputStream(new
File(destination + fileName));

            int read = 0;
            byte[] bytes = new byte[1024];

            while ((read = in.read(bytes)) != -1) {
                out.write(bytes, 0, read);
            }

            in.close();
            out.flush();
            out.close();

            System.out.println("New file created!");
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }

    public void eliminarLista() {
        try {

            managerLista.eliminarLista(selected.getCodigoLista());
            actionCargarListas();
            ;
            JsfUtil.addSuccessMessage("Lista Eliminada con EXITO");

        } catch (Exception e) {
            e.printStackTrace();
            JsfUtil.addSuccessMessage(e.getMessage());
        }
    }
}

```

## ManagerLista

```

public class ManagerLista {

    @EJB
    ManagerDAO managerDAO;
}

```

```

public ManagerLista() {

}
@SuppressWarnings("unchecked")

public List<Lista> findAllListas() throws Exception
{
    List<Lista> result = managerDAO.findAll(Lista.class);
    return result;
}

public void crearListas(Lista procesoLi) throws Exception
{
    managerDAO.insertar(procesoLi);
}

public void eliminarLista(Integer codigoLista) throws Exception {
    managerDAO.eliminar(Lista.class, codigoLista );
}

public List<SelectItem> findNombreListas() throws Exception
{
    ArrayList<SelectItem> itemproc = new ArrayList<SelectItem>();
    itemproc.clear();
    List<Lista> result = managerDAO.findAll(Lista.class);
    for (Lista lista : result) {
        itemproc.add(new
SelectItem(lista.getCodigoLista(), lista.getNombreLista()));
    }
    return itemproc;
}

public List<Lista> BusquedaProcesoLista(int proceso) throws Exception{
    String filtroSQL;
    filtroSQL= "o.procesoEleccione="+proceso;

    return managerDAO.findWhere(Lista.class, filtroSQL,
"o.procesoEleccione");
}
}

```

## BeanProcesos

```

@SessionScoped
@ManagedBean

```

```

public class BeanProcesos {

    @EJB

    ManagerProceso managerProceso;
    List<ProcesoEleccione> procesoElecciones;
    List<SelectItem> procesoEleccioness;
    private int codigoProceso;
    private String nombreProceso;
    private String descripcionProceso;
}

```

```

private String estadoProceso;
private Date fechaProceso;
private Time HoraInicio;
private Time HoraFin;

@ManagedProperty(value="#{beanVotantes}")
private beanVotantes beanVotantes;

public void setBeanVotantes(beanVotantes beanVotantes) {
    this.beanVotantes = beanVotantes;
}

public String getDescripcionProceso() {
    return descripcionProceso;
}

public void setDescripcionProceso(String descripcionProceso) {
    this.descripcionProceso = descripcionProceso;
}

public String getEstadoProceso() {
    return estadoProceso;
}

public void setEstadoProceso(String estadoProceso) {
    this.estadoProceso = estadoProceso;
}

public Date getFechaProceso() {
    return fechaProceso;
}

public void setFechaProceso(Date fechaProceso) {
    this.fechaProceso = fechaProceso;
}

public Time getHoraInicio() {
    return HoraInicio;
}

public void setHoraInicio(Time horaInicio) {
    HoraInicio = horaInicio;
}

public Time getHoraFin() {
    return HoraFin;
}

public void setHoraFin(Time horaFin) {
    HoraFin = horaFin;
}

private ProcesoEleccione selected;

public ProcesoEleccione getSelected() {
    return selected;
}

```



```

}

public void setSelected(ProcesoEleccion selected) {
    this.selected = selected;
}

@PostConstruct
public void Iniciar() {
    actionCargarProcesos();
    actionCargarProcesosVotantes();
    beanVotantes.comprobrsession();
}

public BeanProcesos() {
}

public void actionCargarProcesos() {
    try {
        procesoElecciones = managerProceso.findAllProcesos();
    } catch (Exception e) {

        e.printStackTrace();
    }
}

public void actionCargarProcesosVotantes() {
    try {
        procesoElecciones = managerProceso.findAllProcesos();
    } catch (Exception e) {

        e.printStackTrace();
    }
}

public String actionGuardarProcesos() {
    try {
        ProcesoEleccion proceso = new ProcesoEleccion();
        System.out.println(codigoProceso + " " + nombreProceso+ "
" + descripcionProceso+ " " + estadoProceso+ " " + fechaProceso+ " " +
HoraInicio);

        proceso.setNombreProceso(nombreProceso);
        proceso.setDescripcionProceso(descripcionProceso);
        proceso.setEstadoProceso(estadoProceso);
        proceso.setFechaProceso(fechaProceso);
        proceso.setHoraInicio(HoraInicio);
        managerProceso.crearProcesos(proceso);
        actionCargarProcesos();
        JsFUtil.addSuccessMessage("Proceso Electoral Agregado con
EXITO");

        return "index";
    } catch (Exception e) {

        e.printStackTrace();
        JsFUtil.addSuccessMessage(e.getMessage());
    }
}

```

```

        return "";
    }
}

public void editarProceso(){
    try{
        managerProceso.actualizarProcesos(selected);
    }catch(Exception e){

    }
}

public void eliminarProceso(){
    try{

managerProceso.eliminarProcesos(selected.getCodigoProceso());
        actionCargarProcesos();
        JsfUtil.addSuccessMessage("Proceso Electoral Eliminado con
EXITO");

    }catch(Exception e){
        e.printStackTrace();
        JsfUtil.addSuccessMessage(e.getMessage());
    }
}

public List<ProcesoEleccion> getProcesoElecciones() {
    return procesoElecciones;
}

public void setProcesoElecciones(List<ProcesoEleccion>
procesoElecciones) {
    this.procesoElecciones = procesoElecciones;
}

public int getCodigoProceso() {
    return codigoProceso;
}

public void setCodigoProceso(int codigoProceso) {
    this.codigoProceso = codigoProceso;
}

public String getNombreProceso() {
    return nombreProceso;
}

public void setNombreProceso(String nombreProceso) {
    this.nombreProceso = nombreProceso;
}

public List<SelectItem> getProcesoEleccioness() {

```

```

        return procesoElecciones;
    }

    public void setProcesoElecciones(List<SelectItem> procesoElecciones)
    {
        this.procesoElecciones = procesoElecciones;
    }
}

```

## ManagerProceso

```

@Stateless(mappedName = "managerProceso")
@LocalBean
public class ManagerProceso {

    @EJB
    ManagerDAO managerDAO;

    public ManagerProceso() {

    }

    @SuppressWarnings("unchecked")

    public List<ProcesoEleccion> findAllProcesos() throws Exception
    {
        List<ProcesoEleccion> result =
managerDAO.findAll(ProcesoEleccion.class);
        return result;
    }

    public List<SelectItem> findAllProcesoss() throws Exception
    {
        ArrayList<SelectItem> itemproc = new ArrayList<SelectItem>();
        itemproc.clear();
        List<ProcesoEleccion> result =
managerDAO.findAll(ProcesoEleccion.class);
        for (ProcesoEleccion procesoEleccion : result) {
            itemproc.add(new
SelectItem(procesoEleccion.getCodigoProceso(),procesoEleccion.getNombreProc
eso()));
        }
        return itemproc;
    }

    public void crearProcesos(ProcesoEleccion proceso) throws Exception
    {
        managerDAO.insertar(proceso);
    }

    public void actualizarProcesos(ProcesoEleccion proceso) throws
Exception
    {
        ProcesoEleccion proceso1 = (ProcesoEleccion)
managerDAO.findById(ProcesoEleccion.class, "2");
        proceso1.setNombreProceso(proceso.getNombreProceso());
        managerDAO.insertar(proceso);
    }
}

```

```

    }
    public void eliminarProcesos(Integer CodigoProceso) throws Exception {
        managerDAO.eliminar(ProcesoEleccione.class, CodigoProceso );
    }
}

```

## beanVotantes

```

@SessionScoped
@ManagedBean

```

```

public class beanVotantes {
    @EJB

    ManagerVotantes managerVotantes;
    ManagerProceso managerProceso;
    ManagerPerfil managerPerfil;
    ManagerControl managerControl;
    ManagerLista managerLista;
    ManagerDAO managerDAO;
    ProcesoEleccione procesose;

    BeanProcesos b;
    List<Persona> persona;
    List<SelectItem> procesoEleccioness;
    List<Lista> liston;
    List<Lista> listasss;
    List<Control> control;
    List<Votacione>listas1;

    private int identificacion;
    private String nombre;
    private String email;
    private String clave;
    private int telefono;
    private int codigoProceso;
    private String fotografia;
    private Persona selected;
    private String Perfil;
    private boolean log = false;
    private String descripcion;
    private String fechaingreso;

    public boolean isLog() {
        return log;
    }

    public void setLog(boolean log) {
        this.log = log;
    }

    public void actionCargarProcesos() {
        try {
            procesoEleccioness = managerProceso.findAllProcesoss();
        } catch (Exception e) {

            e.printStackTrace();
        }
    }
}

```

```

    }
}

public Persona getSelected() {
    return selected;
}

public void setSelected(Persona selected) {
    this.selected = selected;
}

public List<Persona> getPersona() {
    return persona;
}

public void setPersona(List<Persona> persona) {
    this.persona = persona;
}

public int getIdentificacion() {
    return identificacion;
}

public void setIdentificacion(int identificacion) {
    this.identificacion = identificacion;
}

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getClave() {
    return clave;
}

public void setClave(String clave) {
    this.clave = clave;
}

public int getTelefono() {
    return telefono;
}

public void setTelefono(int telefono) {
    this.telefono = telefono;
}

```

```

public int getCodigoProceso() {
    return codigoProceso;
}

public void setCodigoProceso(int codigoProceso) {
    this.codigoProceso = codigoProceso;
}

public String getFotografia() {
    return fotografia;
}

public void setFotografia(String fotografia) {
    this.fotografia = fotografia;
}

public String getPerfil() {
    return Perfil;
}

public void setPerfil(String perfil) {
    Perfil = perfil;
}

public List<SelectItem> getProcesoEleccioness() {
    return procesoEleccioness;
}

public void setProcesoEleccioness(List<SelectItem> procesoEleccioness)
{
    this.procesoEleccioness = procesoEleccioness;
}

public String getDescripcion() {
    return descripcion;
}

public void setDescripcion(String descripcion) {
    this.descripcion = descripcion;
}

public String getFechaingreso() {
    return fechaingreso;
}

public void setFechaingreso(String fechaingreso) {
    this.fechaingreso = fechaingreso;
}

public List<Lista> getListon() {
    return liston;
}

public void setListon(List<Lista> liston) {
    this.liston = liston;
}

public List<Control> getControl() {
    return control;
}

```

```

}

public void setControl(List<Control> control) {
    this.control = control;
}

public List<Lista> getListasss() {
    return listasss;
}

public void setListasss(List<Lista> listasss) {
    this.listasss = listasss;
}

public List<Votacione> getListas1() {
    return listas1;
}

public void setListas1(List<Votacione> listas1) {
    this.listas1 = listas1;
}

@PostConstruct
public void Iniciar() throws MessagingException {
    comprbrsession();
    actionCargarVotantes();
}

public void actionCargarProcesosVotantes() {
    try {
        procesoEleccioness = managerProceso.findAllProcesoss();
    } catch (Exception e) {

        e.printStackTrace();
    }
}

public beanVotantes() {
}

public void actionCargarVotantes() {
    try {
        persona = managerVotantes.findAllProcesos();
    } catch (Exception e) {

        e.printStackTrace();
    }
}

public String actionGuardarVotantes() {
    try {
        actionCargarVotantes();
        Persona persona = new Persona();

        System.out.println(identificacion + " " + nombre + " " +
email + " " + telefono + " " + codigoProceso);
        persona.setIdentificacion(identificacion);

```

```

        persona.setNombre(nombre);
        persona.setEmail(email);
        persona.setTelefono(telefono);

        ProcesoEleccion p = new ProcesoEleccion();

        p.setCodigoProceso(codigoProceso);
        System.out.println(p.getCodigoProceso());

        persona.setProcesoEleccion(p);

        managerVotantes.crearVotante(persona);

        actionCargarVotantes();

        JsfUtil.addSuccessMessage("Votante Agregado con EXITO");
        return "index";
    } catch (Exception e) {
        System.out.println(e.getMessage());

        e.printStackTrace();
        JsfUtil.addSuccessMessage(e.getMessage());
        return "";
    }
}

public void eliminarVotante() {
    try {

        managerVotantes.eliminarVotante(selected.getIdentificacion());

        actionCargarVotantes();
        JsfUtil.addSuccessMessage("Votante Eliminado con EXITO");

    } catch (Exception e) {
        e.printStackTrace();
        JsfUtil.addSuccessMessage(e.getMessage());

    }
}

public List<String> completeText(String query) {
    List<String> results = new ArrayList<String>();
    for (int i = 0; i < 100; i++) {
        results.add(query + i);
    }

    return results;
}

public String cambioClave() {
    Persona p;
    try {

        p = managerVotantes.BusquedaID(identificacion);

        if (p == null) {
            JsfUtil.addSuccessMessage("No existe ese usuario");

```



```

        return "";
    } else {
        p.setClave(clave);

        System.out.println(p.getClave());
        managerVotantes.ActualizaaVotante(p);

        return "";
    }
} catch (Exception e) {
    e.printStackTrace();
}
return "";
}

public void logout() {
    //FacesContext context = FacesContext.getCurrentInstance();
    HttpSession session = (HttpSession)
FacesContext.getCurrentInstance().getExternalContext().getSession(false);
    session.invalidate();
    log = false;

    //
    context.getExternalContext().redirect("/VotoTesis/VotacionesUsuarios/E
xito.xhtml");

    // boolean log = false;

}

public String comprobrsession() {
    if (log == false) {
        return "LoginFin?faces-redirect=true";
    }
    return "";
}

public void BusquedaporCedula() {

    String ci = "";
    // ci = String.valueOf(identificacion);
    try {
        managerVotantes.BuscarporID(identificacion);
        JsfUtil.addSuccessMessage("Votante VerificadoID:" + " " +
identificacion + " " + ci);

        return;

    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

public String Logout() {
    FacesContext context = FacesContext.getCurrentInstance();

```

```

FacesContext.getCurrentInstance().getExternalContext().invalidateSession();

        try {
context.getExternalContext().redirect("/VotoTesis/LoginFin.xhtml");
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return "LoginFin?faces-redirect=true";
    }

    // Enviar e-mail

    public void SendEmail() throws AddressException, MessagingException {
        String smtpServer = "smtp.gmail.com";
        int port = 587;
        final String userid = "votoelectronico740@gmail.com";// change

            // accordingly
        final String password = "1001073178";// change accordingly
        String contentType = "text/html";
        String subject = "CONFIRMACIÓN PREVIO VOTACIÓN ELECTRÓNICA ";
        String from = "votoelectronico740@gmail.com"; //
de:votoelectronico740@gmail.com

        // pass:1001073178
        String to = email;// some invalid address

        String body = "Esta es una prueba de correo electronico para
mail de voto eletronico" + "Su id es:"
            + "identificacion"
            + "Ingresar en la siguiente direccion:
http://localhost:8282/VotoTesis/CambioClave/Id.xhtml";

        Properties props = new Properties();

        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.starttls.enable", "true");
        props.put("mail.smtp.host", smtpServer);
        props.put("mail.smtp.port", "587");
        props.put("mail.transport.protocol", "smtp");

        Session mailSession = Session.getInstance(props, new
javax.mail.Authenticator() {
            protected PasswordAuthentication
getPasswordAuthentication() {
                return new PasswordAuthentication(userid,
password);
            }
        });

        MimeMessage message = new MimeMessage(mailSession);
        message.addFrom(InternetAddress.parse(from));
        message.setRecipients(Message.RecipientType.TO, to);
        message.setSubject(subject);

```

```

message.setContent(body, contentType);

Transport transport = mailSession.getTransport();
try {
    System.out.println("Enviando ....");
    transport.connect(smtpServer, port, userid, password);
    transport.sendMessage(message,
message.getRecipients(Message.RecipientType.TO));
    System.out.println("Envio realizado ...");
} catch (Exception e) {
    System.err.println("Error de envio: ");
    e.printStackTrace();
}
transport.close();
} // end function main()

public void actionCargarControl() {
    try {
        control = managerControl.findAllControl();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public String logindos() {
    Persona p;
    fechaingreso = new java.util.Date().toString(); // Fecha actual
del
    // sistema
    descripcion = "Logeo e ingreso";

    try {
clave);
        p = managerVotantes.verificarUsuario(identificacion,

        FacesContext context = FacesContext.getCurrentInstance();

        if (p == null) {
            context.addMessage(null, new FacesMessage("No
existe tal usuario "));

            setIdentificacion(0);
            setClave("");
            return "";
        } else {

            Control cont = new Control();
            cont.setIdentificacion(identificacion);
            System.out.println(cont.getIdentificacion());

            cont.setDescripcion(descripcion);

```

```

        System.out.println(cont.getDescripcion());

        cont.setFecha_ingreso(fechaingreso);
        System.out.println(cont.getfecha_ingreso());

        System.out.println(fechaingreso + " " +
descripcion + " " + identificacion);

        managerVotantes.crearControl(cont);

        p =
managerVotantes.verificarUsuario(identificacion, clave);

        context.getExternalContext().getSessionMap().put("user",
p.getNombre());

        FacesMessage("Siga")); // context.addMessage(null, new

        // l= managerPerfil.findbyIDdos(identificacion);

        if (p.getPerfil().equals("Admin")) {
            //
if(l.getDescripcion().equals("ADMINISTRADOR")){

                context.getExternalContext().redirect("/VotoTesis/admin.xhtml");
            }
            if (p.getPerfil().equals("Consultor")) {
                //
if(l.getDescripcion().equals("CONSULTOR")){

                    context.getExternalContext().redirect("/VotoTesis/consultor.xhtml");
                }
                if (p.getPerfil().equals("Votante")) {
                    // if(l.getDescripcion().equals("VOTANTE")){

                    context.getExternalContext().redirect("/VotoTesis/Personas.xhtml");
                }

                context.addMessage(null, new FacesMessage("No
existe UN ROL "));

                log = true;
                return "";

            }

        } catch (Exception e) {
            e.printStackTrace();
        }
        return "";
    }

    public void buscarProceso() {
        // retorna el codigo de proceso
        Lista l;
        Persona p;

```

```

    try {

        p = managerVotantes.BusquedaID(identificacion);

        System.out.println(p.getProcesoEleccion().getCodigoProceso());
        int ValorProceso =
p.getProcesoEleccion().getCodigoProceso();
        liston = managerVotantes.BusquedaProceso(ValorProceso);

        System.out.println("exito");

    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}

public String generarVoto(int codigoLista) throws Exception {
    // genera certificado y complemento en la tabla votaciones

    // Votacione vo;

    // vo = managerVotantes.verificarVotacion(identificacion);
    FacesContext context = FacesContext.getCurrentInstance();
    FacesMessage msg = null;
    // if (vo != null) {

    // return "";

    // } else {
        try {

            System.out.println(identificacion + "--" + codigoLista);

            Votacione v = new Votacione();

            Persona p = new Persona();
            p.setIdentificacion(identificacion);
            v.setPersona(p);

            Lista l = new Lista();

            l.setCodigoLista(codigoLista);
            System.out.println(l.getCodigoLista());
            v.setLista(l);

            managerVotantes.crearCertificado(v);

            context.getExternalContext().redirect("/VotoTesis/VotacionesUsuarios/E
xito.xhtml");
            return "";

        }
        catch (Exception e) {

```

```

        context.getExternalContext().redirect("/VotoTesis/VotacionesUsuarios/Y
aVoto.xhtml");
        return "";
    }
}

public void Resultados(){
    System.out.println(codigoProceso);
    //Lista l;
    try {
        listasss=
managerLista.BusquedaProcesoLista(codigoProceso);

        System.out.println("exito");

    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

public void buscarResultado() {
    // retorna el codigo de proceso

    System.out.println(codigoProceso);

    try {
        listasss = managerVotantes.BusquedaProceso(codigoProceso);
        // System.out.println(listasss);
        for(int x=0;x<listasss.size();x++) {
            //
            System.out.println(listasss.get(x).getCodigoLista());
            int nro = (listasss.get(x).getCodigoLista());

            listas1= managerVotantes.BuscarNroVotos(nro);

            int tamaño = listas1.size();

            System.out.println("El número de votos para la lista: "+
listasss.get(x).getCodigoLista()+ " es de "+tamaño);

            FacesContext context = FacesContext.getCurrentInstance();

            context.addMessage(null, new FacesMessage("BIEN", "El número
de votos para la lista: "+ listasss.get(x).getCodigoLista()+ " es de
"+tamaño) );

        }

    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
}
}

```

```
}
```

## MangerVotantes

```
@Stateless(mappedName = "managerVotantes")  
@LocalBean
```

```
public class ManagerVotantes {  
  
    @EJB  
    ManagerDAO managerDAO;  
  
    public ManagerVotantes() {  
    }  
  
    @SuppressWarnings("unchecked")  
  
    public List<Persona> findAllProcesos() throws Exception  
    {  
        List<Persona> result = managerDAO.findAll(Persona.class);  
        return result;  
    }  
  
    public void crearVotante(Persona persona) throws Exception  
    {  
        managerDAO.insertar(persona);  
    }  
  
  
  
    public void crearCertificado(Votacion votacion) throws Exception  
    {  
        managerDAO.insertar(votacion);  
    }  
  
  
  
    public void crearControl(Control controles) throws Exception {  
        managerDAO.insertar(controles);  
    }  
  
  
  
  
  
  
  
  
  
    public void ActualizaVotante(Persona persona) throws Exception  
    {  
        managerDAO.actualizar(persona);  
    }  
  
  
  
    public void eliminarVotante(Integer codigoVotante) throws Exception {  
        managerDAO.eliminar(Persona.class, codigoVotante );  
    }  
  
  
  
  
  
  
  
  
  
    public void BuscarporID(Integer id) throws Exception{
```

```

        managerDAO.findById(Persona.class, id);
    }

    public List<Votacione> BuscarNroVotos(int codigoLista) throws
Exception{
    String filtroSQL;
    filtroSQL= "lista="+codigoLista;

    return managerDAO.findWhere(Votacione.class, filtroSQL,
"o.lista");
}

    public Persona verificarUsuario1(int username, String password) throws
Exception{
    Persona p=(Persona) managerDAO.findWhere(Persona.class, "
o.Persona.identificacion = "+username
        + " and o.Persona.clave = "+"'+password+'",
    "").get(0);
    return p;
}

    public Persona verificarUsuario(int cedula,String password) throws
Exception{
    Persona persona=(Persona) managerDAO.findById(Persona.class,
cedula);

    if((persona.getIdentificacion().equals(cedula))&&persona.getClave().eq
uals(password)){
        return persona;
    }else{
        return null;
    }
}

    public Votacione verificarVotacion(int cedula)throws Exception{
cedula);
    Votacione vota=(Votacione) managerDAO.findById(Votacione.class,
Integer citemp=vota.getPersona().getIdentificacion());

    if(citemp.equals(cedula)){
        return vota;
    }else{
        return null;
    }
}

    public Persona BusquedaID(int cedula) throws Exception{

```



```

        Persona persona=(Persona) managerDAO.findById(Persona.class,
cedula);
        if(persona.getIdentificacion().equals(cedula)){
            return persona;
        }
        return null;
    }

    public List<Lista> BusquedaProceso(int proceso) throws Exception{
        String filtroSQL;
        filtroSQL= "o.procesoEleccione="+proceso;

        return managerDAO.findWhere(Lista.class, filtroSQL,
"o.procesoEleccione");
    }
}

```

## JavaMail

```

public class javamail {
    private final Properties properties = new Properties();
    private Session session;

    private void init() {
        properties.put("mail.smtp.host", "smtp.gmail.com");
        properties.put("mail.smtp.starttls.enable", "true");
        properties.put("mail.smtp.port", 587);
        properties.put("mail.smtp.mail.sender",
"diegorea593@gmail.com");
        properties.put("mail.smtp.password", "1001073178dr");
        properties.put("mail.smtp.user", "diegorea593@gmail.com");
        properties.put("mail.smtp.auth", "true");
        session = Session.getDefaultInstance(properties);
    }

    public void send(String destino, String asunto, String mensaje) {
        init();
        try {
            MimeMessage message = new MimeMessage(session);
            message.setFrom(new InternetAddress((String) properties
                .get("mail.smtp.mail.sender")));
            message.addRecipient(Message.RecipientType.TO, new
InternetAddress(
                destino));
            message.setSubject(asunto);

            message.setText(mensaje, "<h1>Plan anual de
compras</h1>");

            Transport t = session.getTransport("smtp");
            t.connect((String) properties.get("mail.smtp.user"),
                (String)
properties.get("mail.smtp.password"));
            t.sendMessage(message, message.getAllRecipients());
            t.close();
        } catch (MessagingException e) {
            return;
        }
    }
}

```

}  
}  
}