

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA SISTEMAS

DISERTACIÓN PREVIA A LA OBTENCIÓN EL TÍTULO
DE
INGENIERO(A) EN SISTEMAS Y COMPUTACIÓN

“Diseño e implementación del voto electrónico mediante el uso de una aplicación web, para las elecciones de la federación de estudiantes de la Pontificia Universidad Católica del Ecuador”

NOMBRES:

Alex Calvopiña

Stephanie García

DIRECTOR/A: Fabián De La Cruz

Ciudad Quito, Año 2016

DEDICATORIA

Este esfuerzo lo dedico a mis padres quienes han sido el pilar más importante para lograr este objetivo , ellos que mediante su apoyo incondicional me han sabido guiar en esta etapa de mi vida universitaria que está por finalizar.

Dedico este trabajo a mi familia y en especial a mi madre quien con sus consejos, su comprensión y amor me apoyo en los momentos más difíciles y por ayudarme con todos los recursos necesarios para que pueda culminar con mi carrera universitaria.

AGRADECIMIENTO

Agradezco a la Pontificia Universidad Católica del Ecuador y a sus prestigiosos profesores de la Facultad de Ingeniería, Escuela de Sistemas que han impartido en mí el conocimiento necesario para culminar con éxito mis estudios en la carrera.

A mi director de tesis , Fabián de la Cruz que ha sido la luz que guió cada paso de esta disertación.

A mis compañeros que me han brindado su tiempo a lo largo de la carrera universitaria y principalmente a Dios por bendecirme día a día, darme la sabiduría, la fortaleza y la salud.

Agradezco a mis profesores quienes compartieron todos sus conocimientos conmigo permitiéndome finalizar mi carrera universitaria con éxito.

A mi director de tesis, quien nos apoyo y guió durante todo el desarrollo de esta disertación.

A mis amigos que me apoyaron y recorrieron este camino junto a mí.

Agradezco a Dios quien supo guiarme por el buen camino, darme fuerzas cuando se presentaron problemas y enseñarme a encarar todas las adversidades que se presentaron.

ÍNDICE

1. INTRODUCCIÓN	6
1.1. Datos de la Organización	7
1.2. Justificación	8
1.3. Antecedentes	8
1.4. Alcance	8
1.5. Objetivos	9
1.5.1. Objetivo General	9
1.5.2. Objetivos Específicos	9
2. FUNDAMENTO TEÓRICO	10
2.1. Metodología para el Desarrollo de Aplicaciones.....	10
2.1.1. Metodologías Ágiles.....	11
2.1.2. Extreme Programming (XP).....	12
2.2. Elementos para Desarrollo de Aplicaciones	16
2.2.1. Importancia del Modelado.....	16
2.2.2. Lenguajes de Programación a Utilizar	25
2.2.3. Base de Datos a Utilizar	29
2.3. Patrón de diseño a utilizar	34
2.3. Herramientas Seleccionadas	38
3. CASO DE ESTUDIO	40
3.1. Exploración.....	40
3.1.1. Requerimientos Funcionales.....	40
3.1.2. Diagrama General	41
3.1.3. Modelo Conceptual.....	42
3.1.4. Prototipo	43
3.1.5. Requerimientos No Funcionales.....	45
3.2. Planificación	47
3.3. Primera Iteración.....	47
3.3.1. Requerimientos Funcionales.....	48
3.3.2. Diseño.....	54
3.3.3. Desarrollo	67
3.3.4. Pruebas	67
3.4. Segunda Iteración	69

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

3.4.1. Requerimientos Funcionales	70
3.4.2. Diseño	77
3.4.3. Desarrollo	88
3.4.4. Pruebas	89
3.5. Tercera Iteración	90
3.5.1. Requerimientos Funcionales	90
3.5.2. Diseño	94
3.5.3. Desarrollo	101
3.5.4. Pruebas	101
3.6. Producción o Despliegue	102
3.7. Mantenimiento	103
3.8. Fin de Proyecto	110
4. CONCLUSIONES Y RECOMENDACIONES	113
4.1. Conclusiones	113
4.2. Recomendaciones	114
5. BIBLIOGRAFÍA	115

1. INTRODUCCIÓN

La presente disertación está basada en el desarrollo de una aplicación web para la Federación de Estudiantes de la Pontificia Universidad Católica del Ecuador, la cual es una organización estudiantil que representa la comunidad universitaria y busca el bienestar de todos los miembros que conforman dicha comunidad.

Como parte de la comunidad se observó que el proceso de sufragio, que se lleva a cabo para escoger a los representantes de la FEUCE, es el proceso habitual que se ha manejado por años en el Ecuador, en el cual los votantes se acercan a una mesa electoral para emitir su voto y aquellos representantes de cada mesa son los encargados del conteo y la emisión de resultados obtenidos en su mesa.

Mediante el desarrollo de una aplicación web de votación electrónica, se desea automatizar el proceso de sufragio, permitiendo la gestión de las listas y candidatos postulantes, ejercer el derecho al voto por parte de los estudiantes y la emisión de los resultados obtenidos en el proceso de sufragio, con la finalidad de mejorar el conteo de votos y evitar el impacto ambiental que ocasiona el empleo de papeletas de votación.

Para el desarrollo de la aplicación se utilizó la metodología Extreme Programming (XP), porque es una metodología ágil que permite un desarrollo flexible y adaptable a los cambios que se presenten durante la implementación. Además, esta metodología permite una alta interacción con el cliente a lo largo del proceso de desarrollo.

Para el cumplimiento de las funcionalidades planteadas y el desarrollo de la aplicación se lo realizó en tres iteraciones, en cada iteración se realizó de desarrollo de tres funcionalidades, en la segunda y tercera iteración se agregaron mejores que permiten incrementar el desempeño de la aplicación.

Empleando planes de pruebas al culminar cada iteración, podemos mantener contacto con el cliente, de tal manera que puede observar el avance de la aplicación y puede retroalimentarnos con sugerencias acerca del diseño o la implementación de la aplicación. Aquellas sugerencias aceptadas en el informe de pruebas serán incluidas en el desarrollo de la siguiente iteración.

Se empleo Modelo Vista Controlador (MVC) como patrón de diseño para el desarrollo de la aplicación, es decir, se encuentra dividida en tres capas: el modelo donde se encuentran todas las conexiones a la base de datos, el controlador que es el encargado de los eventos y la vista que es lo visible para el usuario.

Finalmente, la presente disertación se ha dividido en los siguientes capítulos: Capítulo 1 que es el capítulo que se presenta actualmente; Capítulo 2, se muestra toda la Fundamentación Teórica que sirve de sustento para el desarrollo de la presente disertación; Capítulo 3, se presenta el desarrollo de la disertación en base a la metodología escogida y basándose en los fundamentos teóricos del capítulo 2; y Capítulo 4 muestra una recopilación de las Conclusiones y Recomendaciones que se presentan al finalizar la presente disertación.

1.1. Datos de la Organización

Nombre: Federación de Estudiantes de la Pontificia Universidad Católica del Ecuador.

Actividad: Estructurar una Federación de Estudiantes que permita a cada uno de sus miembros obtener bienestar y representación en su vivencia dentro de la comunidad universitaria a través de políticas institucionales incluyentes, democráticas, participativas, equitativas, eficientes, y solidarias. ¹

Ubicación: Av. 12 de Octubre 1076 y Roca (PUCE) / (593 02) 299 1668 (Quito – Ecuador)

Características: Somos una organización de representación estudiantil sin fines de lucro, sin distinción de credo, clase o ideologías, creada en la PUCE, SEDE QUITO. ²

Generadora de proyectos y propuestas innovadoras, pionera en el ámbito académico y de desarrollo, para los estudiantes y la comunidad universitaria, con proyección hacia la sociedad civil.

¹ Federación de Estudiantes de la Pontificia Universidad Católica del Ecuador “Quienes Somos” <http://www.feuce.ec/category/asociacion/quienes-somos/> Acceso: 15-09-2015

² Federación de Estudiantes de la Pontificia Universidad Católica del Ecuador “Quienes Somos” <http://www.feuce.ec/category/asociacion/quienes-somos/> Acceso: 15-09-2015

Realizamos nuestras actividades gracias a la autogestión y aporte de los estudiantes; contamos con personal comprometido, calificado y una estructura organizacional sustentable.³

1.2. Justificación

La FEUCE es una organización estudiantil que ha sido desarrollada con el fin de ayudar a los estudiantes de la PUCE en aspectos académicos, sociales y culturales. La FEUCE está consolidada por un grupo de líderes que llevan a cabo funciones específicas, entre ellas está el representar fielmente a todos los estudiantes de la universidad. Dichos líderes, han sido escogidos mediante el proceso habitual de votación electoral que se maneja dentro del país

1.3. Antecedentes

Durante nuestros años como estudiantes de la PUCE hemos ejercido nuestro derecho al voto siguiendo el proceso habitual de votación electoral que se maneja en el Ecuador, el cual tiene como aspectos negativos, la ineficiencia en el conteo de los votos, el daño ambiental ocasionado por el uso de una gran cantidad de papel, entre otros.

Por esta razón nos hemos motivado a realizar una automatización de este proceso brindando una forma accesible, eficiente, segura y amigable con el medio ambiente.

1.4. Alcance

La presente disertación culminará con el diseño y la implementación de un sistema de votación electrónica que permita la gestión de las listas y los candidatos postulantes, emisión del voto por parte de los votantes mediante el ingreso sistema por un usuario y contraseña y la emisión de los resultados obtenidos después de haberse realizado el

³ Federación de Estudiantes de la Pontificia Universidad Católica del Ecuador “Quienes Somos”
<http://www.feuce.ec/category/asociacion/quienes-somos/> Acceso: 15-09-2015

proceso de sufragio. Se aplican pruebas de tipo alfa que permitirán obtener sugerencias por parte del usuario, permitiendo mejorar la calidad y el desempeño del sistema.

1.5. Objetivos

1.5.1. Objetivo General

Diseñar e implementar el voto electrónico mediante el uso de una aplicación web, para las elecciones de la Federación de Estudiantes de la Pontificia Universidad Católica del Ecuador.

1.5.2. Objetivos Específicos

- Realizar el diseño adecuado de la aplicación web.
- Seguir fielmente la metodología establecida para el desarrollo.
- Definir y cumplir estándares durante todo el proceso de desarrollo.
- Emplear herramientas que efectivicen el desarrollo de la aplicación.
- Realizar pruebas tipo alfa que evidencien el correcto desempeño de la aplicación.

2. FUNDAMENTO TEÓRICO

El fundamento teórico, que se desarrolla a continuación, proporciona una perspectiva más clara acerca de los temas y conceptos básicos que se establecen como cimiento para el desarrollo de esta disertación.

Primero partiremos con la descripción de las metodologías para el desarrollo de aplicaciones y la definición detallada de las metodologías ágiles y extreme programming, que es la metodología que se aplicará durante todo el proceso de desarrollo de la aplicación.

Posteriormente se habla sobre los elementos o herramientas que se emplean durante el desarrollo de aplicaciones como son la importancia del modelado y UML, el lenguaje de programación que se va a usar para el desarrollo de la aplicación con una descripción puntual de PHP, la base de datos en la cual se almacenara la información manejada por la aplicación con una introducción a PostgreSQL.

Por último, se define que es un patrón de diseño y las características principales del patrón Modelo Vista Controlador (MVC) el cuál se empleará en el proceso de desarrollo.

2.1. Metodología para el Desarrollo de Aplicaciones

Una metodología de desarrollo de software se refiere a un marco de trabajo o a la estructura que se utiliza para organizar, planificar y controlar el proceso de desarrollo de un software. Existe una amplia variedad de marcos de trabajo que han ido evolucionando a lo largo de los años, cada uno de estos consta de sus propias fortalezas y debilidades que han sido reconocidas. Una metodología de desarrollo de software no es necesariamente adecuada para ser empleada por todos los proyectos. Cada una de las metodologías disponibles se definen de manera determinada para que tipo de proyectos son aptas, basándose en diversas técnicas, consideraciones de organización, proyecto y equipo. (CMS, 2008)

Una metodología de desarrollo de software consta de:

- Una filosofía de desarrollo de programas que están enfocados en la estructura del proceso de desarrollo de software.
- Herramientas, modelos y métodos que ayuden a la asistencia de un proceso de desarrollo de software.

Algunos marcos de trabajo se encuentran vinculados o asociados con algún tipo de organización que desarrolle, apoye el uso, y promueva dicha metodología. Dicha metodología es posteriormente documentada.

Las Metodologías de Desarrollo de Software primordialmente se encuentran enfocadas en técnicas tradicionales y modernas de modelado de sistemas que permitan desarrollar un producto de calidad, para ser entregado al cliente.

2.1.1. Metodologías Ágiles

Las metodologías ágiles son adaptativas más que predictivas(tradicionales). Se las conoce así ya que a diferencia de las tradicionales, que potencian la planificación detallada de todo el Desarrollo de Software a largo plazo, las metodologías ágiles están listas para adaptarse a los cambios que existan durante este proceso llegando incluso hasta el punto de cambiar ellos mismos, para conseguir adaptarse.

Las metodologías ágiles están más orientadas al personal de desarrollo que al proceso del desarrollo. Intentan trabajar con la naturaleza del personal de desarrollo, de tal manera que permiten que el Desarrollo de Software se convierta en una actividad grata e interesante.

El enfoque fue planteado por primera vez por Martin en 1991, y se dio a conocer en la comunidad de Ingeniería de Software con el mismo nombre que su libro, RAD (Rapid Application Development). RAD consistía en un entorno de desarrollo altamente productivo, en el que participaban grupos pequeños de programadores utilizando herramientas que generaban código en forma automática tomando como entradas sintaxis de alto nivel.

Como resultado de esta nueva teoría se crea un Manifiesto Ágil cuyas principales ideas son:

- Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados.
- Es más importante crear un producto software que funcione que escribir documentación exhaustiva.
- La colaboración con el cliente debe prevalecer sobre la negociación de contratos.
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

Entre los principales métodos ágiles tenemos el XP (eXtreme Programming), Scrum, Iconix, Cristal Methods, AUP entre otras.

2.1.2. Extreme Programming (XP)

Extreme Programming (XP) es una metodología de desarrollo ágil creada en el año 1996. Dichas metodologías están formadas y se basan en el manifiesto ágil, mencionado anteriormente.

XP está centrada en potenciar las relaciones entre los miembros del equipo, el trabajo en equipo como punto primordial para el aprendizaje de todos y promocionando un buen ambiente de trabajo. La relación entre el cliente y los miembros del equipo es continua durante todo el proceso, de esta forma se busca satisfacer las necesidades y crear un producto de calidad.

XP permite al cliente, mediante las llamadas, Historias de Usuario, especificar sus requisitos mediante tarjetas de papel.

2.1.2.1. Características de XP.

Una vez conocido el significado de Extreme Programming y explicado a breves rasgos sus funciones esenciales, se procede a mostrar cuales son las características esenciales a modo de comparación, y que hacen de XP una metodología de desarrollo Ágil. Estas características son:

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

- La interacción con el cliente durante todo el proceso es más importante que un contrato.
- La adaptabilidad ante los cambios es más importante que el seguimiento de un plan
- Los individuos y sus interacciones son más importantes que los procesos y las herramientas.
- Un Software en funcionamiento es más importante que una documentación exhaustiva.

(Bustamante & Rodriguez, 2014)

2.1.2.2. Roles en XP

Esta Metodología de Desarrollo divide a los miembros del equipo de trabajo mediante roles, con el fin de que cada miembro esté enfocado en uno de estos. Para XP, dichos roles son los expuestos a continuación:

- **Programador:** Son los encargados de programar el sistema en general.
- **Cliente:** Es quien escribe las historias de usuario y valida la implementación.
- **Tester:** Escribe las pruebas funcionales con el cliente, realiza un estudio y entrega los resultados al responsable.
- **Tracker:** Encargado de hacer el seguimiento a todo el proceso de XP, analizar si el tiempo y los recursos predestinados están siendo bien usados.
- **Entrenador:** Debe conocer todo el proceso para dar guías a los distintos miembros del equipo, revisar si el proceso fluya correctamente.
- **Consultor:** Debe conocer acerca de un tema específico y ayudar al resto del equipo con problemas que tengan respecto a dicho tema.
- **Gestor:** Se encarga de la relación programador-cliente, ayuda en la coordinación a lo largo de todo el proceso.

(Bustamante & Rodriguez, 2014)

2.1.2.3. Fases en XP

se obtenga un software con la menor cantidad de defectos al final de cada iteración. Para la programación se hace uso de la programación en parejas.

Entregables por cada iteración: Casos de Uso, Diagramas de clases, Diagramas de Secuencia, Diagramas de Actividades, Prototipos, Módulos funcionales y planes de pruebas.

- **Fase IV:** Producción o Despliegue: Se analiza posibles características nuevas que se las pueda añadir al software, se prueba su rendimiento en ambientes ideales.
- **Fase V:** Mantenimiento: Esta fase puede requerir incorporar nuevo personal y cambiar la estructura del equipo para un próximo proyecto. Una vez puesto en producción el sistema, se realiza el proceso de mantenimiento del software por si en algún momento llegase a fallar.
- **Fase VI:** Fin del proyecto: Se finaliza cuando el cliente no tiene más historias que incluir, por lo tanto los miembros del equipo han satisfecho todas las necesidades del cliente en el software presentado. Los miembros del equipo de trabajo pueden realizar una retroalimentación y analizar sus puntos fuertes y sus débiles para en un futuro los mejoren. (Penadés & Letelier, 2006)

Algunos de los entregables serán explicados en el siguiente apartado.

2.1.2.4. *Ventajas y desventajas en XP*

Al aplicar una Metodología de Desarrollo, las ventajas y desventajas muchas veces dependen del proyecto al cuál estará dirigida, por lo tanto XP, cuenta con una gran variedad de ventajas que permiten facilitar el desarrollo del proyecto, algunas de ellas son las siguientes:

- Permite un despliegue flexible y adaptable al cambio dentro del proceso de desarrollo.

- No existe sobrecarga para el equipo de desarrollo gracias a la distribución de Roles.
- Interacción con el cliente durante todo el proceso de desarrollo.

No es pertinente utilizar la metodología XP, cuando:

- Son proyectos a largo plazo.
- Se tiene un alto grado de dificultad y el equipo no este en la capacidad de desarrollarlo en su totalidad, ya que se pueden generar altas comisiones en caso de fallar el proceso.

(Bustamante & Rodriguez, 2014)

2.2.Elementos para Desarrollo de Aplicaciones

Dentro del desarrollo de aplicaciones se debe de tener en consideración los elementos que se requieren durante el proceso de desarrollo, estos elementos varían de acuerdo a la funcionalidad o el propósito con el que se los emplee, los elementos o herramientas son objetos elaborados con el fin de facilitar la realización de una tarea o actividad dentro del desarrollo de la aplicación.

Las herramientas son diseñadas y fabricadas para cumplir uno o más propósitos específicos; que apoyen, faciliten o presten asistencia al personal de desarrollo durante la elaboración y puesta en marcha de la aplicación.

2.2.1. Importancia del Modelado

2.2.1.1. ¿Qué es UML?

UML es un acrónimo para Lenguaje Unificado de Modelado. UML es un lenguaje empleado para describir un software desarrollado, es decir, es una notación que puede

ser utilizada en conjunto con una metodología, por lo tanto no se lo considera como una metodología. (Mas Sobre UML, 2005)

El Lenguaje Unificado de Modelado o UML unifica sobre todo los métodos de Booch, Rumbaugh (OMT) y Jacobson, pero con un alcance mayor al de éstos. La mayoría de los métodos consisten en un lenguaje y un proceso de modelar. El lenguaje de modelado es la notación, principalmente gráfica, de que se valen los métodos para expresar los diseños. El proceso es la orientación o directrices que nos dan para desarrollar el diseño de un software. (Que es UML?, 1999)

Los principales objetivos en el diseño de UML fueron éstos: obtener un lenguaje simple pero suficientemente expresivo, que permitiese modelar aplicaciones en cualquier dominio; obtener un lenguaje legible, puesto que sería un lenguaje utilizado por las personas; y permitir la generación automática de código. (García, Moreira, & Rossi, 2004)

La aparición de UML ha supuesto el reconocimiento de la actividad del diseño en base de modelos, como una actividad clave para producir software de calidad. Sin embargo, resulta paradójico que al mismo tiempo haya surgido el movimiento del desarrollo ágil de software (Agile Software), que considera que el valor está en el código y en las personas más que en los modelos y los procesos, así como la llamada programación extrema (eXtreme Programming, XP). (García, Moreira, & Rossi, 2004)

UML es considerado un estándar por OMG (Grupo de Administración de Objetos) el cual ha sido diseñado para visualizar, especificar, construir y documentar software. Un modelo es una simplificación de la realidad. Como podemos observar en la Ilustración 2, un lenguaje de modelado consiste de vistas, diagramas, símbolos utilizados en los modelos y un conjunto de mecanismos generales o reglas que muestran cómo emplear los elementos.

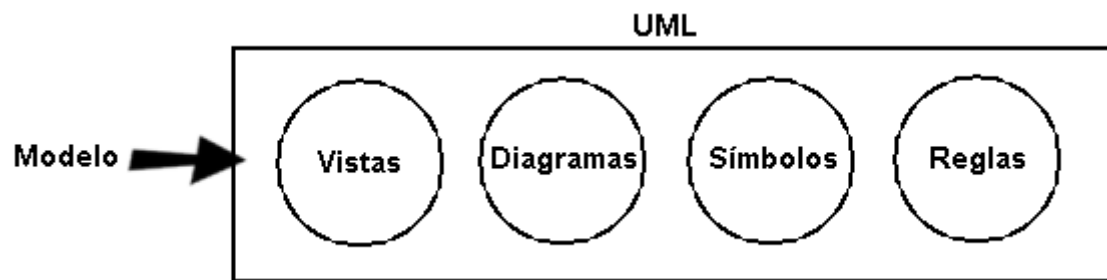


Ilustración 2 - UML (Roman, 2015)

El modelado es esencial en la construcción de software para:

- Comunicar la estructura de un sistema complejo.
- Especificar el comportamiento deseado del sistema.
- Comprender mejor lo estamos construyendo.
- Descubrir oportunidades de simplificación y reutilización.
- Visualizar cómo es o queremos que sea un sistema.
- Especificar la estructura o comportamiento de un sistema.
- Proporcionar plantillas que nos sirvan de guía para la construcción de un sistema.
- Documentar las decisiones que se han adoptado.

(Booch, Rumbaugh, & Jacobson, 2000)

Existe un paradigma de desarrollo, no centrado en el código sino en los modelos, que se denomina desarrollo guiado por modelos. Con esta nueva visión, se puede obtener código a partir de modelos centrados en el dominio del problema e independientes de cualquier aspecto relacionado con la plataforma, mediante transformaciones de modelos. (García, Moreira, & Rossi, 2004)

La Ingeniería de Modelos es un área que cada vez cobra más importancia y que no se reduce al ámbito académico sino que está recibiendo atención por parte de las principales empresas de desarrollo de software, y ya existen IDEs (Integrated Development Environments) que incorporan herramientas de transformación de modelos basadas en MDA (Model-Driven Architecture).

El objetivo sería que las herramientas generasen tanto código como fuese posible a partir de modelos, en la línea que se ha seguido a lo largo de la historia de la programación de liberar al programador de escribir todo aquel código que pueda ser generado por herramientas. Con la importancia que cobran los modelos y la ingeniería de modelos, como es lógico, hay áreas de investigación que han adquirido especial importancia como son la calidad y evaluación de los modelos, la definición de perfiles, la formalización de las transformaciones de modelos o la creación de herramientas de transformación.

Desde la primera versión adoptada por OMG, UML ha evolucionado a través de diferentes versiones. El crecimiento ha significado mejorar su expresividad y precisión, pero no ha sufrido cambios en sus aspectos esenciales. (García, Moreira, & Rossi, 2004)

2.2.1.2. Principios del Modelado

Con toda la experiencia adquirida en el paso de los años por el uso del modelado se sugieren cuatro principios para su desarrollo:

Primero:

"La elección de qué modelos crear tiene una profunda influencia sobre cómo se acomete un problema y cómo se da forma a una solución." (Booch, Rumbaugh, & Jacobson, 2000)

La elección de los modelos correctos pueden permitir mejor comprensión de problemas complejos a los cuales no se les habría encontrado solución o entendimiento desde cualquier otro punto de vista. Y en el caso de emplear modelos erróneos se desviarían del camino al entendimiento o incluso la solución del mismo.

Segundo:

"Todo modelo puede ser expresado a diferentes niveles de precisión." (Booch, Rumbaugh, & Jacobson, 2000)

En ocasiones, se puede necesitar una representación sencilla y superficial de un modelo de software, pero también habrá ocasiones que se necesite de un modelado más detallado de todo lo que se desea obtener. Por lo cual es mejor escoger un modelado que se pueda observar desde distintos niveles de precisión, debido a que el usuario final y el desarrollador tienen diferente perspectiva para observar o apreciar un sistema, y cada uno se enfoca a lo que quiere o necesita.

Tercero:

"Los mejores modelados están ligados a la realidad." (Booch, Rumbaugh, & Jacobson, 2000)

El modelado está definido como una simplificación de la realidad, entonces el modelado de un software es la representación o abstracción del mismo sin alejarse de las condiciones reales o en este va a ser implementado.

Cuarto:

"Un único modelo no es suficiente. Cualquier sistema no trivial se aborda mejor a través de un pequeño conjunto de modelos casi independientes." (Booch, Rumbaugh, & Jacobson, 2000)

Para comprender la arquitectura de un sistema se necesitan tener diferentes puntos de vista de la arquitectura, para que pueda ser comprendido en su totalidad para lo cual se tiene una variedad de diagramas en los que se representa el sistema.

2.2.1.3. Diagramas UML

Un Diagrama es la representación gráfica de un conjunto de elementos, visualizado la mayoría de veces como un grafo conexo de nodos (elementos) y arcos (relaciones). Los diagramas se grafican para visualizar un sistema desde diferentes perspectivas, de forma que un diagrama es una proyección de un sistema. Un diagrama puede contener cualquier combinación de elementos y relaciones.

UML estandariza 9 tipos de diagramas para representar gráficamente un sistema de distintos puntos de vista:

- Diagrama de Clases
- Diagrama de Objetos
- Diagrama de Casos de Uso
- Diagrama de Secuencia
- Diagrama de Colaboración
- Diagrama de Estados
- Diagrama de Actividades
- Diagrama de Componentes
- Diagrama de Despliegue.

2.2.1.3.1. *Diagramas de Clases*

Un Diagrama de Clases es un diagrama que muestra un conjunto de interfaces, colaboraciones y sus relaciones. Gráficamente un diagrama de clases es una colección de nodos y arcos. Los diagramas de clases cubren la vista de diseño estática de un sistema.

Los diagramas de clases contienen normalmente los siguientes elementos:

- Clases
- Interfaces
- Colaboraciones
- Relaciones de Dependencia, Generalización y Asociación.

En la Ilustración 3 podemos observar un ejemplo de diagrama de clase, donde tenemos como clases Persona, Despachador, Cliente, Detalle y Pedido, también podemos observar una relación de generalización entre despachador y persona, se tiene una relación de asociación entre detalle y pedido.

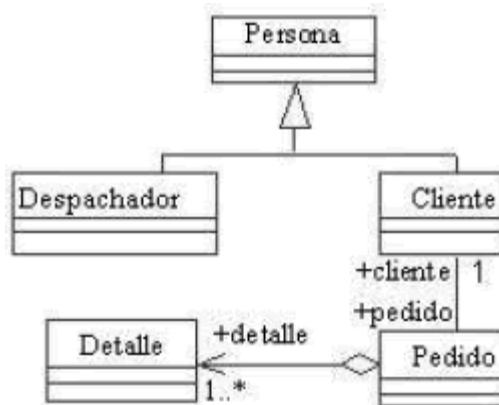


Ilustración 3 - Diagrama de Clases (DYNA, 2016)

2.2.1.3.2. Diagramas de Casos de Uso

Un Diagrama de casos de uso es un diagrama que muestra un conjunto de casos de uso, actores y relaciones. Un caso de uso es una descripción de un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable de valor para un actor. (Booch, Rumbaugh, & Jacobson, 2000)

Los diagramas de caso de uso se suelen utilizar en el modelado del sistema desde el punto de vista de sus usuarios para representar las acciones que realiza cada tipo de usuario.

Los diagramas de caso de uso contiene:

- Casos de uso
- Actores
- Relaciones de dependencia, generalización y asociación.

En la Ilustración 4 tenemos un ejemplo de diagrama de caso de uso con algunos de sus componentes.

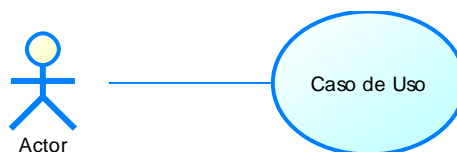


Ilustración 4 - Diagrama de Caso de Uso

Elaborado por: Stephanie García, Alex Calvopiña; 2016

2.2.1.3.3. Diagramas de Secuencia

Muestran una interacción concreta: un conjunto de objetos y sus relaciones, junto con los mensajes que se envían entre ellos.

Resaltan la ordenación temporal de los mensajes que se intercambian.

En la Ilustración 5, podemos observar un diagrama de secuencia para realizar el Ingreso de un Reclamo y podemos observar la interacción entre capas y los mensajes que se envían entre ellas.

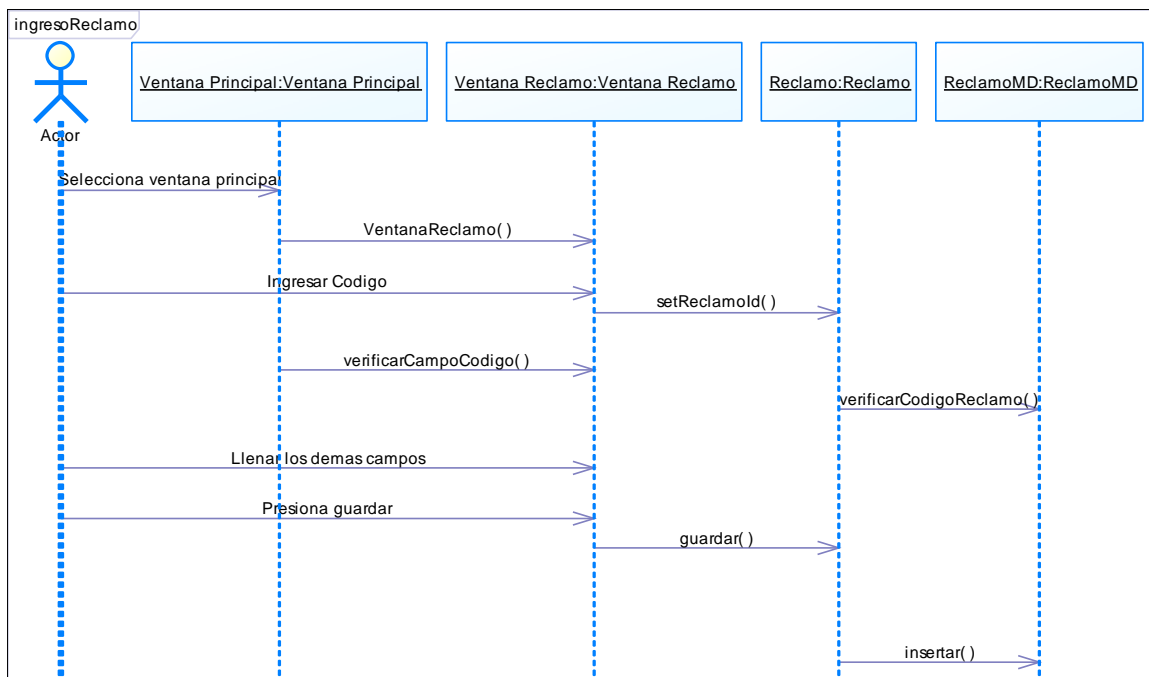


Ilustración 5 - Diagrama de Secuencia

Elaborado por: Stephanie García, Alex Calvopiña; 2016

2.2.1.3.4. Diagramas de Actividades

Es un tipo especial diagrama de estados que muestra el flujo de actividades dentro de un sistema. Cubren la vista dinámica de un sistema, son importantes para modelar el funcionamiento de un sistema y resaltan el flujo de control entre objetos.

Un diagrama de actividades muestra el flujo de actividades. Una actividad es una ejecución no atómica en curso, dentro de una máquina de estados. Las actividades

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

producen finalmente alguna acción, que está compuesta de computaciones atómicas ejecutables que producen un cambio en el estado del sistema o la devolución de un valor. Gráficamente es una colección de nodos y arcos. (Booch, Rumbaugh, & Jacobson, 2000)

Muestran el orden en que se van realizando las tareas dentro de un sistema.

Los diagramas de actividades contienen:

- Estados de actividad y estados de acción.
- Transiciones
- Objetos.

Los diagramas de actividades permiten mostrar el flujo de un proceso como podemos observar en la Ilustración 6, donde se realiza una cancelación de reserva.

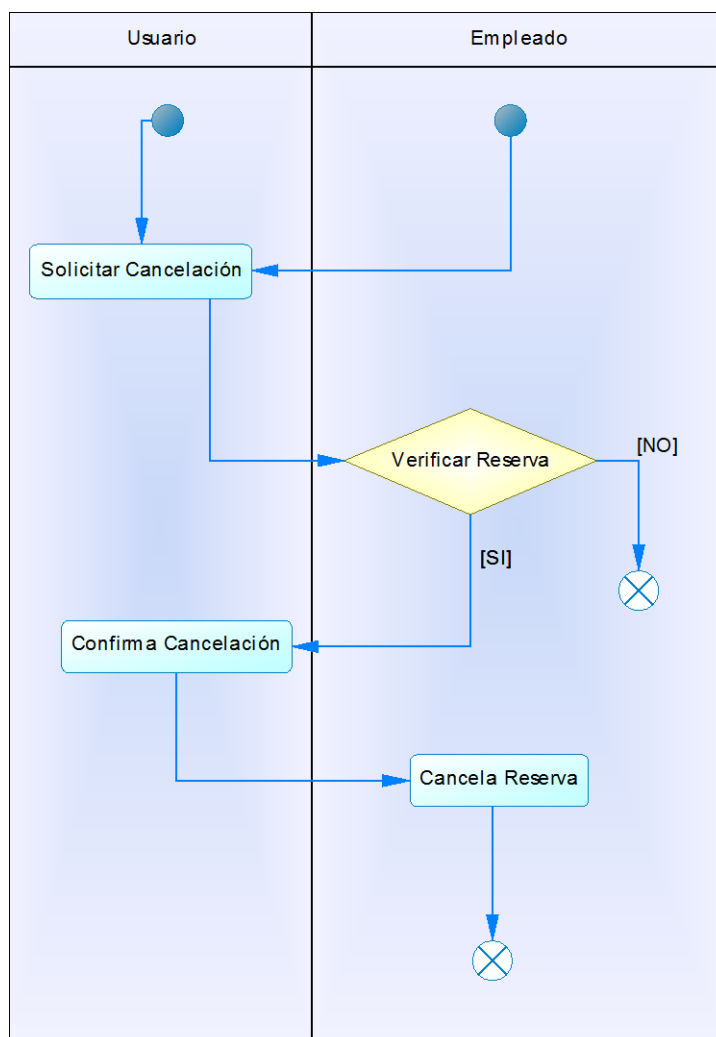


Ilustración 6 - Diagrama de Actividades

Elaborado por: Stephanie García, Alex Calvopiña; 2016

2.2.2. Lenguajes de Programación a Utilizar

Es un lenguaje artificial que ha sido diseñado para expresar acciones que pueden ser ejecutadas por los computadores. Pueden ser empleados en la creación de programas que permitan controlar el comportamiento físico y lógico de una máquina, o como una forma de expresar la comunicación humana. (EcuRed, 2016)

Por lo tanto, un lenguaje de programación es una forma práctica para que los seres humanos puedan dar instrucciones a una máquina. El uso de un lenguaje de programación tiene sus ventajas:

- Es un tipo de lenguaje más fácil de comprender que el lenguaje de máquina (1s ó 0s)
- Permite tener una mayor portabilidad del programa o aplicación desarrollada, es decir que se puede adaptar o ejecutar fácilmente en diferentes tipos de máquinas. (CCM, 2016)

Al igual que cualquier otro tipo de lenguaje está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. El proceso en el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa se denomina como programación.

A la programación también se la define como el proceso de creación de un programa de computadora, empleando los siguientes pasos:

- El desarrollo lógico del programa para resolver un problema en particular.
- Escritura de la lógica del programa empleando un lenguaje de programación específico (codificación del programa)
- Ensamblaje o compilación del programa hasta convertirlo en lenguaje de máquina.
- Prueba y depuración del programa.

- Desarrollo de la documentación. (Lenguaje de Programación, 2016)

2.2.2.1. *Historia de PHP*

El lenguaje PHP fue diseñado en 1994 por Rasmus Lerdford para sus necesidades personales posteriormente se realizó el lanzamiento a principios del año 1995.

En 1995 se publicó una versión reescrita bajo el nombre de PHP/FI versión 2. Es capaz de manejar formularios y acceder a la base de datos mSQL.

En 1997, se realiza el lanzamiento de la versión 3 el cual fue desarrollado por un equipo liderado por Rasmus Lerdford.

En 2000, el analizador PHP se migra al motor de análisis de Zend el cual le permite proporcionar un mejor rendimiento y admitir un mayor número de extensiones, todo esto se obtiene en la versión 4.

En 2004 nace la versión 5 la cual es basada en la versión del motor Zend2, aporta varias características nuevas, la mayoría de ellas relacionadas con el desarrollo orientado a objetos. (Heurtel, 2015)

En 2015, se libera la versión 7 la cual está basada en Zend Engine 3, el cual presenta mejoras en el rendimiento, presenta el reemplazo de muchos "errores" restantes de la versión anterior con la excepciones renovadas, y sintaxis abreviada para la importación de varios artículos. (Popov, 2014)

2.2.2.2. *PHP*

PHP es un acrónimo de Hypertext Preprocessor. PHP es un lenguaje de programación interpretado, de alto nivel y de código abierto que es ampliamente utilizado. Scripts PHP se ejecutan del lado del servidor, su código es incluido en una página HTML. A diferencia de un lenguaje como Java Script, donde el código se ejecuta del lado del cliente (en el explorador), el código de PHP es ejecutado del lado del servidor. El resultado de esta ejecución se incrusta en la página HTML, que se envía al navegador. El cual no tiene conocimiento de la existencia del proceso que se está llevando a cabo del lado del servidor. (Heurtel, 2015)

Los archivos PHP pueden contener texto HTML, CSS, JavaScript y código PHP los cuales son ejecutados en el servidor y su resultado es devuelto al navegador con un HTML plano.

PHP permite realizar páginas web de contenidos dinámicos; puede crear, abrir, leer, escribir, borrar y cerrar archivos en el servidor; permite el envío y recepción de cookies; además nos permite cifrar los datos, lo cual le da un valor de seguridad al lenguaje; y todo esto se puede generar total o parcialmente en el momento de la llamada de la página gracias a la información que se recopila en un formulario o se extrae de una base de datos, en la cual se puede añadir, borrar o modificar los datos. (Heurtel, 2015)

PHP tiene varias características que lo definen como son:

- La alta velocidad de proceso para el desarrollo óptimo del lenguaje ya que únicamente consume pocos recursos.
- Posee estabilidad, aunque esta no se encuentre totalmente libre de errores, pero es lo más cerca que se puede estar de la perfección, debido al apoyo y colaboración de una gran comunidad de programadores que van en busca de una solución de cualquier fallo que se pueda presentar.
- La seguridad, PHP provee distintos niveles de seguridad, los cuales pueden ser configurados desde el archivo “.ini” de nuestra carpeta raíz.
- La simplicidad, la facilidad de aprendizaje permite la generación de una gran cantidad de código empleando cantidades reducidas de tiempo.
- La compatibilidad, puede ejecutado en varias plataformas y tiene compatibilidad con casi todos los servidores empleados en la actualidad, además de soportar una amplia gama de bases de datos. (Ken & Pool, 2015)

2.2.2.3. *Historia de Javascript*

Como sabemos, el Internet siempre esta en constante cambio, y claro, lo estuvo desde los mediados de la década de los 90's, tiempo en el cuñ, Brendan Eich entendió que existía la necesidad de realizar cambios a la interacción que existe entre el Internet y el

usuario, para lo cuál desarrollo un lenguaje de Script llamado Mocha, que posteriormente paso a ser LiveScript.

Brendan trabajaba en Netscape, empresa que recibió una licencia por parte de Sun, de esta forma, Javascript fue concedido como el nombre final para este lenguaje.

(Lyle, 2013)

2.2.2.4. *Javascript*

Es un lenguaje de Script actualmente muy utilizado , ya que tiene la facilidad de ser incrustado dentro de HTML fácilmente. Javascript puede ser utilizado de parte del lado del servidor o del cliente, dependiendo de la aplicación.

Javascript permite crear animaciones, generar eventos, dentro de la pagina HTML, agradables y necesarias para los programadores, tiene librerías existentes para distintos propósitos, entre ellas JQuery.

2.2.2.4.1. *jQuery*

jQuery es una librería con una extensa cantidad de funciones que tienen como propósito facilitar el uso de JavaScript dentro de una página web. Dentro de las funcionalidades de jQuery, este nos permite la manipulación, manejo de eventos, animación y más; es fácil de usar debido a que funciona mediante una gran variedad de navegadores. Con una combinación de versatilidad y capacidad de ampliación.

A diferencia de JavaScript para llevar a cabo tareas comunes que requieren de gran cantidad de líneas de código, jQuery las envuelve en métodos que pueden ser llamados mediante una línea de código.

jQuery contiene las siguientes características:

- HTML manipulación
- Manipulación de CSS
- Eventos en HTML
- Efectos y animaciones

- AJAX
- Utilidades

jQuery es considerado el más popular, y también la más extensible librería. (jQuery Intro, 2016)

2.2.3. Base de Datos a Utilizar

Las Bases de Datos son un repositorio, de datos computarizados referentes a un mismo entorno, que mediante el uso de Sistemas de Bases de Datos se puede:

- Insertar nuevos archivos.
- Consultar datos de archivos.
- Modificar datos en archivos.
- Eliminar archivos existentes.

(Date, 2001)

2.2.3.1. *Sistemas gestores de Bases de Datos (SGBD)*

Los SGBD, son los encargados de entablar la comunicación entre el usuario final y el sistema operativo.

Los Sistemas gestores de Bases de Datos, permiten a los usuarios realizar las acciones explicadas en el punto anterior, mediante dos tipos de lenguajes que son:

Lenguaje de definición de datos(DDL): Se utilizan para diseñar los esquemas, las vistas y las estructuras de almacenamiento.

Lenguaje de manipulación de datos(DML): Se utilizan para realizar operaciones de consulta y modificación de datos.

2.2.3.1.1. *Arquitectura de los Sistemas gestoras de Bases de Datos*

Los SGBD tienen tres niveles que facilitan la realización de las funciones:

Nivel físico: Íntimamente relacionado con el almacenamiento de los datos. El SGBD realiza las lecturas del sitio donde se encuentren los datos directamente.

Nivel lógico: Se presentan los datos mediante un esquema conceptual.

Nivel externo: Íntimamente relacionado con el usuario. (Ramírez, 2008)

2.2.3.2. Esquemas de las Bases de Datos

Cuando se tiene la necesidad de crear una base de datos, el proceso es abstraer la información del mundo real para almacenarlo en una base de datos, como podemos observar en la Ilustración 7, estos son los puntos iniciales y finales.

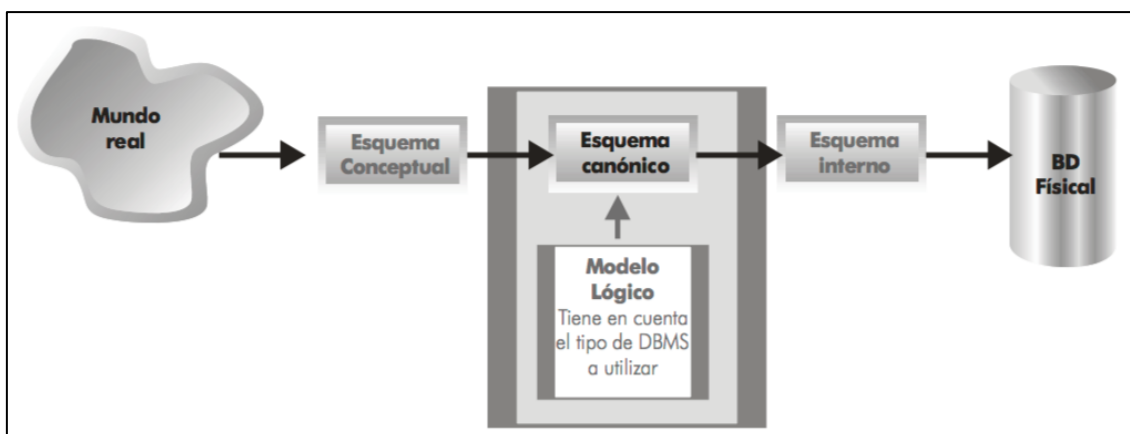


Ilustración 7 - Principios sobre Bases de Datos Relacionales (Sánchez, 2004)

2.2.3.2.1. Esquema Conceptual- Modelo Entidad Relación

Se traduce los requerimientos del sistema a un lenguaje común, que entiendan los miembros del equipo tanto como los clientes. Como se observa en la Ilustración 8, uno de los tipos de esquema conceptual es el Modelo de Entidad-Relación, en el cuál como su nombre lo dice, se expresa, mediante diagramas, en un alto nivel los datos obtenidos del mundo real con sus relaciones pertinentes.

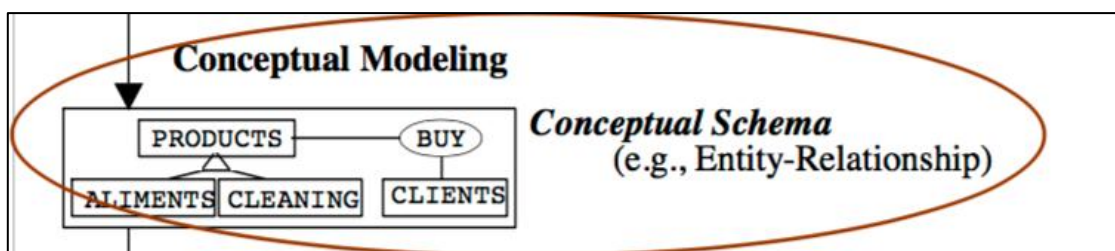


Ilustración 8 - Fundamentos de Bases de Datos

(Fundamentos de Bases de Datos, s.f.)

2.2.3.2.2. Elementos del modelo Entidad – Relación

- **Entidad:** Son abstracciones de objetos físicos o cosas abstractas existentes en el mundo real, que se diferencian entre sí. Se los representa mediante rectángulos.
- **Atributos:** Son las características que identifican plenamente a la entidad en cuestión. Se los representa mediante óvalos.
- **Relación:** Permite generar una dependencia entre dos o mas entidades.
- **Cardinalidad:**
 - **1-1:** Las entidades en cuestión se asocian una a una.

La ilustración 9, presenta que un hombre, por la ley, debe estar casado con una mujer.

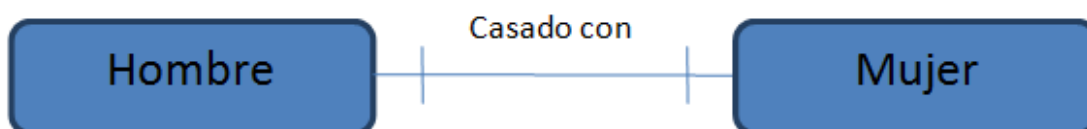


Ilustración 9 - Cardinalidad 1-1

Elaborado Por: Alex Calvopiña, Stephanie García; 2016

- **1-n:** Un aspecto de una Entidad está asociada a (n) de otra Entidad.

La Ilustración 10, define que un padre, puede tener uno o más hijos.

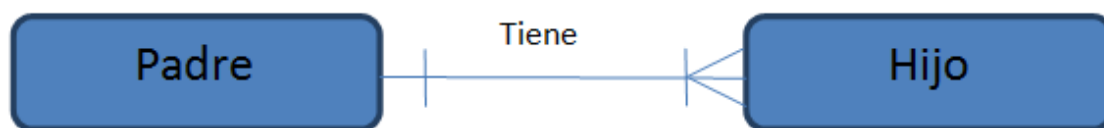


Ilustración 10 - Cardinalidad 1-n

Elaborado por: Alex Calvopiña, Stephanie García; 2016

- **n-n:** Cualquier aspecto de las Entidades en cuestión puede estar asociada con (n) de la otra, o viceversa. (Gutierrez, 2010)

La ilustración 11, describe que un doctor puede atender uno o más pacientes, a la vez que un paciente puede asistir a revisarse por un doctor o más de uno.

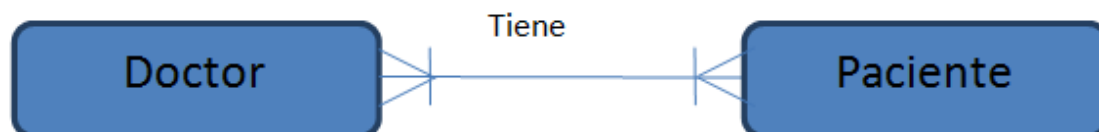


Ilustración 11 - Cardinalidad n-n

Elaborado por: Alex Calvopiña, Stephanie García; 2016

2.2.3.3. *PostgreSql*

2.2.3.3.1. *Historia de PostgreSQL*

Las bases de datos se utilizan principalmente para el almacenamiento y recuperación de datos. Puede utilizar un procesador de textos o una hoja de cálculo para almacenar pequeñas cantidades de datos, pero a la hora de hacerlo con grandes cantidades de datos, la mejor opción son las bases de datos y sus gestores.

PostgreSql es un sistema para la administración de Bases de Datos relacionales orientadas a objetos. Se desarrollo en el departamento de Ciencias Computacionales de la Universidad de Berkeley en California.

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

PostgreSql es licenciado de manera libre, de manera que permite que cualquiera pueda usar, modificar y distribuir de manera libre y gratuita.

Este sistema fue desarrollado en C, lo que facilita la compatibilidad con la mayoría de sistemas operativos.

(Bruce, 2001)

2.2.3.3.2. Tipos de datos habituales:

Una de las características mas importantes de un campo existente en una base de datos, es su tipo de dato. A continuación se presentan los tipos de datos mas habituales a la hora de ingresar datos en PostgreSql, los mismos que determinan la clase de datos que se podrá almacenar, por ejemplo: Integer, Numeric, Permiten tipos de datos numéricos; Char: Permite datos numéricos, alfanuméricos, letras, etc.; mientras que Date, lo hace para datos de tipo Fecha.

Nombre	Tamaño de almacenamiento	Dato
Integer	4 bytes	Rango: -2147483648 to +2147483647
numeric	variable	Rango: Hasta 131072 Dígitos antes del punto decimal; Hasta 16383 Dígitos después del punto decimal.
Char	1 Byte	
Date	4 Bytes	Bajo: 4713 BC Alto: 5874897 AD
Timestamp	8 bytes	Bajo:4713 BC Alto: 294276 AD
Boolean	1 Byte	Estado: True, False

Tabla 1 - Tipos de datos habituales

Elaborado por: Alex Calvopiña, Stephanie García; 2016

Otros: Puntos, líneas , polígonos, arreglos, objetos binarios: Imágenes, sonidos, videos.

2.2.3.3.3. *Números de Postgresql*

Es importante conocer mas a fondo las características que posee esta herramienta, por lo que, las capacidades que tiene son una de estas. Como se puede apreciar en la Tabla 2, las capacidades son prácticamente ilimitadas en su totalidad, por lo que puede utilizarse para la mayoría de sistemas a realizar.

Límite	Valor
Tamaño máximo de base de datos	Ilimitado
Tamaño máximo de tabla	32 TB
Tamaño máximo de fila	1,6 TB
Tamaño máximo de campo	1 GB
Máximo de filas por tabla	Ilimitado
Máximo de columnas por tabla	250-1600(dependiendo del tipo)
Máximo de índices por tabla	Ilimitado

Tabla 2 - PostgreSQL, La base de datos más potente (Arana, 2014)

2.2.3.3.4. *Arquitectura*

Servidor: El que administra los archivos de las bases de datos, el que gestiona las conexiones y ejecuta las acciones sobre la base de datos.

Cliente: Envía las órdenes de ejecución de operaciones a la Base de Datos. (Arana, 2014)

2.3. **Patrón de diseño a utilizar**

Los patrones de diseño son soluciones que se han presentado a problemas típicos y recurrentes se han encontrado al momento de desarrollar una aplicación. Estos patrones pueden ser empleados como el esqueleto o la base para las soluciones de problemas comunes.

Aunque cada aplicación puede ser única, también puede contener partes o módulos que han sido desarrollados en otras aplicaciones. Y en lugar de reinventarlo, podemos emplear algún patrón, ya que estos son soluciones probadas y documentadas por una multitud de programadores. Al emplear un patrón de diseño debemos tomar en cuenta que el problema debe estar sujeto a contextos similares con el patrón.

Los patrones de diseño nos ayudan a cumplir con muchas de las reglas de diseño, como el control de cohesión y acoplamiento además la reutilización de código son algunos de los beneficios al utilizar patrones. (Tedeschi, 2016)

Los patrones se categorizan de la siguiente manera:

- Patrones Creacionales: Inicialización y configuración de objetos.
- Patrones Estructurales: Separan la interfaz de la implementación. Se ocupan de cómo las clases y objetos se agrupan, para formar estructuras más complejas.
- Patrones de Comportamiento: Describen la interaccionan entre como las clases y objetos. (Tedeschi, 2016)

2.3.1. MODELO VISTA-CONTROLADOR

2.3.1.1. Historia MVC

En 1979, mediante el lenguaje SmallTalk, se introdujo por primera vez el concepto del Modelo Vista-Controlador, a partir de ahora MVC; junto con algunos otros lenguajes como Simula I, fueron construyendo gradualmente este patrón de diseño que considera dividir la lógica del negocio de la interfaz de usuario dentro de una aplicación. Se divide en: El modelo, la vista y el controlador.

(Mestras, 2008) (Bascón, 2004)

2.3.1.2. *MVC*

“El patrón MVC es lo que le trae orden al caos, estructura al espagueti, arquitectura a la anarquía en los aspectos relacionados a la implementación de interfaz de usuario.” (Gutierrez, 2010)

2.3.1.2.1. *El modelo*

Es un conjunto de clases que contienen la funcionalidad de la aplicación, la cuál se define del mundo real que se está analizando. Es Independiente del Controlador y la Vista. Sin embargo al aplicar esto resulta un tanto complicado el hecho de que debe existir una comunicación entre los distintos módulos, por lo que SmallTalk siguiere que este modelo este compuesto por dos submódulos.

Modelo del dominio: Son un conjunto de clases que se modelan del mundo real.

Modelo de la aplicación: Conjunto de clases que se relacionan con las clases del modelo del dominio, cuyo funcionamiento es conocer sobre la capa de las vistas y notificar los cambios que existan en el modelo del dominio.

(Bascón, 2004)

2.3.1.2.2. *La vista:*

Como se aprecia en la Ilustración 12, la vista está conectada a un modelo, Puede existir más de una vista. La vista puede acceder al Modelo pero nunca cambiar su estado, pero si recibir notificaciones de esta acción.

La ilustración 12 presenta una conexión entre la Vista y el usuario, lo que significa que ésta es la encargada de mostrar la información al cliente durante todo el proceso.

2.3.1.2.3. *El controlador:*

Es quien se encarga de dirigir el flujo del control de la aplicación mediante mensajes externos, ingresados por el cliente(tanto teclado como el ratón, etc.) como se aprecia en

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

la Ilustración 12. Posteriormente el controlador se encarga de modificar el modelo, abrir o cerrar las vistas. El modelo y las vistas desconocen la existencia del controlador. El controlador se lo conoce como un manejador de los eventos en la aplicación, analizando la Ilustración 12, se identifica que las conexiones que salen y entran a un controlador poseen el nombre de Eventos efectivamente.

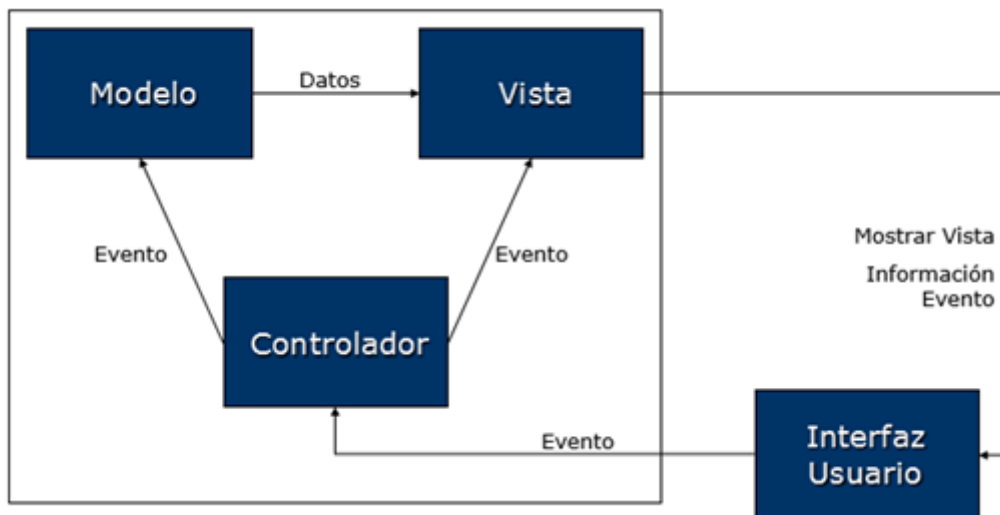


Ilustración 12 - La arquitectura MVC(Patron Modelo Vista-Controlador, 2013)

2.3.1.3. *Ejemplo de un flujo que sigue una aplicación:*

Ejemplo: Ingreso de un nuevo producto en un sistema para una tienda de repuestos de computadoras.

Área involucrada del MVC	Acción desplegada
Vista	La vista carga los datos obtenidos del modelo en una interfaz mostrándolos a través de ComboBox, Tables, etc.
Vista	El usuario llena los campos necesarios y procede a interactuar accionando un evento(encapsulado en un botón) mediante los periféricos de entrada (Teclado, ratón, pantallas táctiles, etc) para ejecutar un nuevo ingreso.

Controlador	Mediante un gestor de eventos(Handler), el controlador ejecuta los eventos accionados(Ingresar nuevo producto).
Controlador, modelo	El controlador accede al modelo con el fin de aplicar la acción enviada por el usuario.(Ingresar en el modelo el nuevo producto, puede ser a una base de datos de productos.)
Vista	A la espera de nuevas interacciones que pongan en marcha un nuevo ciclo.

Tabla 3 - Ejemplo MVC

Elaborado: Alex Calvopiña, Stephanie García; 2016

2.3. Herramientas Seleccionadas

A lo largo de este apartado se ha puesto en manifiesto los conceptos básicos y características centrales de los temas que serán utilizados como base para el desarrollo de esta disertación y después analizar de manera exhaustiva los posibles requerimientos para el desarrollo de la aplicación hemos determinado cuales herramientas son las más óptimas para cumplir con nuestro propósito.

Se ha encontrado pertinente la aplicación de una metodología ágil como lo es Extreme Programmmig (XP), la cual tiene como puntos a favor la programación en parejas, sus iteraciones al momento del proceso de desarrollo y las historias de usuario.

Al ser una aplicación orientada a un entorno web consideramos pertinente el uso de un lenguaje de alto nivel como lo es PHP, debido a las bondades y a la familiaridad que tenemos con el este lenguaje.

Tomando en cuenta que los datos manejados por la aplicación son de gran valor para la institución, se ha puesto énfasis en contar con un nivel de seguridad acorde a dichas características, por esta razón emplearemos el motor de base de datos PostgreSQL.

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

Todas estas herramientas nos permitirán el desarrollo de un producto de calidad que cumpla con los requerimientos previstos.

METODOLOGÍA DE DESARROLLO	Extreme Programming (XP)
LENGUAJES DE DESARROLLO	PHP, Javascript
BASE DE DATOS	Postgresql
PATRÓN DE DISEÑO	Modelo Vista-Controlador (MVC)

Tabla 4 - Herramientas Seleccionadas

Elaborado por: Alex Calvopiña y Stephanie García

3. CASO DE ESTUDIO

El caso de estudio, que se desarrolla a continuación, se realiza aplicando la metodología y las herramientas descritas en el capítulo 2: Fundamento Teórico.

Para el caso de estudio se aplica Extreme Programming como metodología para el desarrollo de la aplicación. La construcción del producto se realiza en base a las fases descritas en esta metodología, para lo cual se despliegan tres iteraciones, en la primera iteración se realiza algunas funcionalidades, en la segunda y tercera iteración se desarrolla más funcionalidades y se realiza mejoras o se complementa el desarrollo de funcionalidades elaboradas en la iteración anterior.

3.1. Exploración

Dentro de esta fase hablaremos de los requerimientos funcionales y no funcionales del sistema.

Los requerimientos son las funcionalidades que se espera obtener de un sistema informático. Estos deben ser claros, precisos y concisos ya que son la base para la construcción y desarrollo del producto. Los requerimientos permiten visualizar más profundamente el sistema deseado y lo que el cliente desea obtener.

Los requerimientos también son un apoyo contractual que sirven como respaldo para lo que se tiene que construir y conocer desde otro punto de vista, la cantidad de trabajo que se ha elaborado del producto.

3.1.1. Requerimientos Funcionales

F0: El sistema permitirá a los usuarios el ingreso al sistema mediante la verificación de su usuario y contraseña.

F1: El sistema gestionará las listas postulantes con sus respectivos miembros.

F2: El sistema mostrará la información de cada lista y sus propuestas.

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

F3: El sistema gestionará los candidatos.

F4: El sistema permitirá sufragar.

F5: El sistema almacenará los detalles del votante en el tiempo, como usuario y hora en que se realizó la votación.

F6: El sistema notificará mediante un correo electrónico la finalización del proceso de votación y emitirá una papeleta de votación.

F7: El sistema realizará el conteo de los votos por cada lista, los votos blancos y nulos.

F8: El sistema emitirá los resultados de las votaciones por candidato.

3.1.2. Diagrama General

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

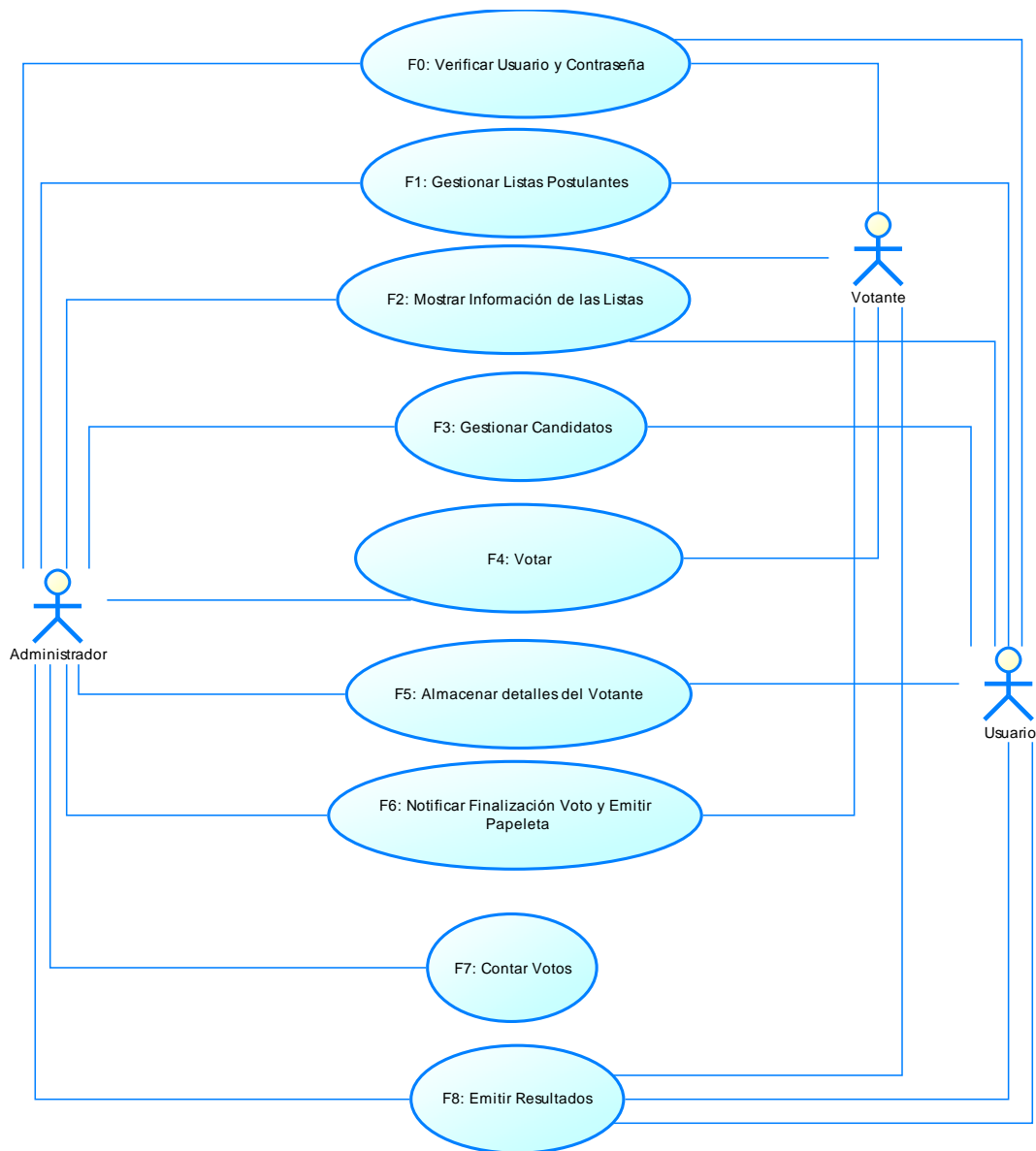


Diagrama 1 - Diagrama de Casos de Uso General

Elaborado por: Alex Calvopiña, Stephanie García; 2016

3.1.3. Modelo Conceptual

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

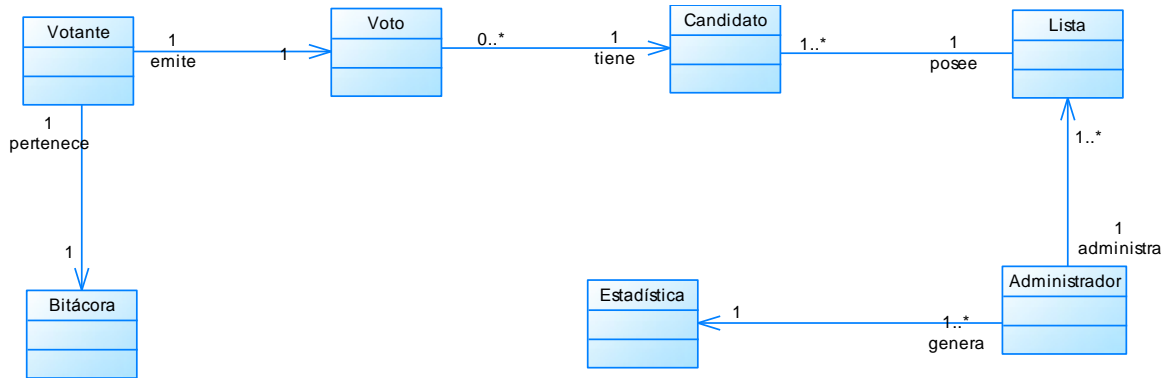


Diagrama 2 - Diagrama Conceptual

Elaborado por: Alex Calvopiña, Stephanie García; 2016

3.1.4. Prototipo



Ilustración 13 - Página Principal

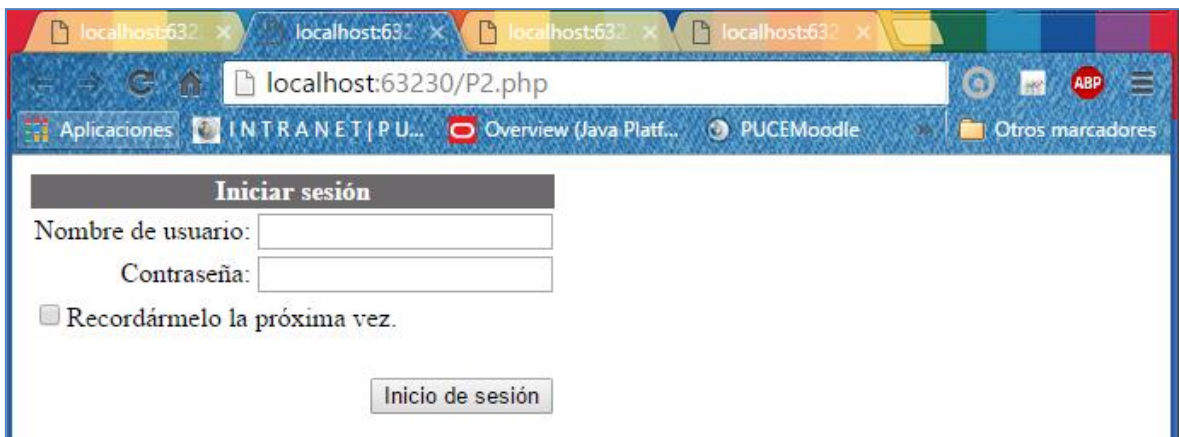


Ilustración 14 - Página Inicio de Sesión

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.



Ilustración 15 - Página Administrador: Gestionar Lista



Ilustración 16 - Página Administrador: Información Votaciones



Ilustración 17 - Página Administrador: Estadísticas

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.



Ilustración 18 - Página Votante

3.1.5. Requerimientos No Funcionales

Eficiencia: Se define la eficiencia del sistema de acuerdo a la rapidez en que se ejecutó el sistema y cuánta memoria requiere el sistema.

- Rapidez en la conectividad a internet.
- Rapidez de respuesta por parte del Servidor.
- Emplear dispositivos que puedan conectarse a internet.

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

Portabilidad: Es la capacidad que tiene un programa para ser transferido de una configuración hardware o entorno software a otro, es decir una plataforma diferente a la que fue diseñada.

- La aplicación puede ser ejecutado en diversos browsers.

Usabilidad: Son un conjunto de características de un programa informático que lo permiten ser sencillo y facilitan la lectura de contenidos como la rapidez de carga de la información.

- La visualización del sistema será la misma en cualquier browser. Además, que se sigue principios de usabilidad y se realizara test con futuros usuarios de la aplicación.
- Se empleará Responsive Design que le permite a la aplicación visualizar en cualquier dispositivo con conectividad a internet.

3.2. Planificación

Una vez analizados los requerimientos de este sistema, las características de las herramientas y las destrezas de los miembros del equipo se ha optado por desarrollar el sistema en 3 iteraciones. Cada una de las iteraciones estará compuesta de tres fases: Diseño, Desarrollo y Pruebas. Para el control de versiones emplearemos GIT.

En la primera iteración se desarrollará la página principal que dará la bienvenida al sistema y una perspectiva general acerca del proceso electoral dentro de la PUCE, a su vez se desarrollará la página correspondiente al inicio de sesión dependiendo del tipo de usuario, ya sea administrador o votante. Como punto final de la primera iteración se implementará una parte de la página de administrador, que es la Gestión de Listas.

En la segunda iteración se tomará en cuenta las observaciones de la primera iteración y se desarrollará la Gestión de Candidatos, la página de votante, en la cual se presentan los candidatos y las opciones de votación conjuntamente con la opción de votar.

En la tercera iteración se considerará las observaciones de las dos iteraciones anteriores y continuaremos con la implementación de la página de administrador, agregando la información de las votaciones junto con las estadísticas y resultados obtenidos del proceso de sufragio.

3.3. Primera Iteración

En base a lo planificado en el punto 3.2 se inicia el desarrollo del sistema con la primera iteración, en la cual se desarrolla la página principal que da la bienvenida al sistema y una perspectiva general acerca del proceso electoral dentro de la PUCE, dentro de esta página también se muestra las listas postulantes y los candidatos de cada lista, a su vez se desarrolla la ventana correspondiente al inicio de sesión dependiendo del tipo de usuario con el que se desee acceder ya sea Administrador o Votante.

Como punto final de la primera iteración se implementa parte de la página de administrador, que es la Gestión de Listas, donde el usuario puede realizar el Ingreso, Modificación, Eliminación y Consultas sobre las Listas Postulantes.

3.3.1. Requerimientos Funcionales

3.3.1.1. Casos de Uso

En base al Diagrama General que se presentó en los Requerimientos Funcionales a continuación se presentan los Diagramas de Caso de Uso Detallados de las funcionalidades que se desarrollarán en la primera iteración. Estos diagramas nos permiten representar las secuencias de acciones que realiza cada tipo de actor durante su contacto con el sistema y en este caso con cada una de las funcionalidades.

F0. Verificar Usuario y Contraseña

Descripción: Verifica el usuario y contraseña para permitir el ingreso al sistema.

Actores: Administrador, Votante, Usuario

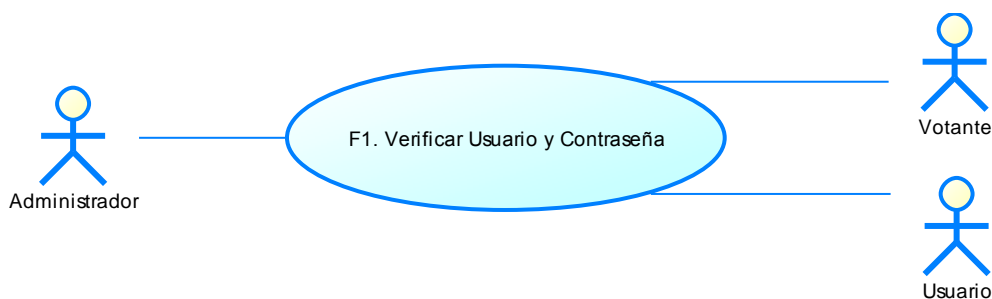


Diagrama 3 - Caso de Uso Verificar Usuario y Contraseña

Elaborado por : Alex Calvopiña, Stephanie García; 2016

Flujo Principal:

1. Actor selecciona la opción "Inicio Sesión" de la página principal.
2. El sistema muestra la página de inicio de sesión.
3. Actor ingresa su usuario y contraseña.
4. Sistema verifica la existencia del usuario y confirma la contraseña. (E1)
5. Sistema muestra la página respectiva al usuario.

Excepciones:

E1: Problemas con la base de datos. La solución contactar al administrador de la base de datos (DBA).

F1: Gestionar Listas Postulantes

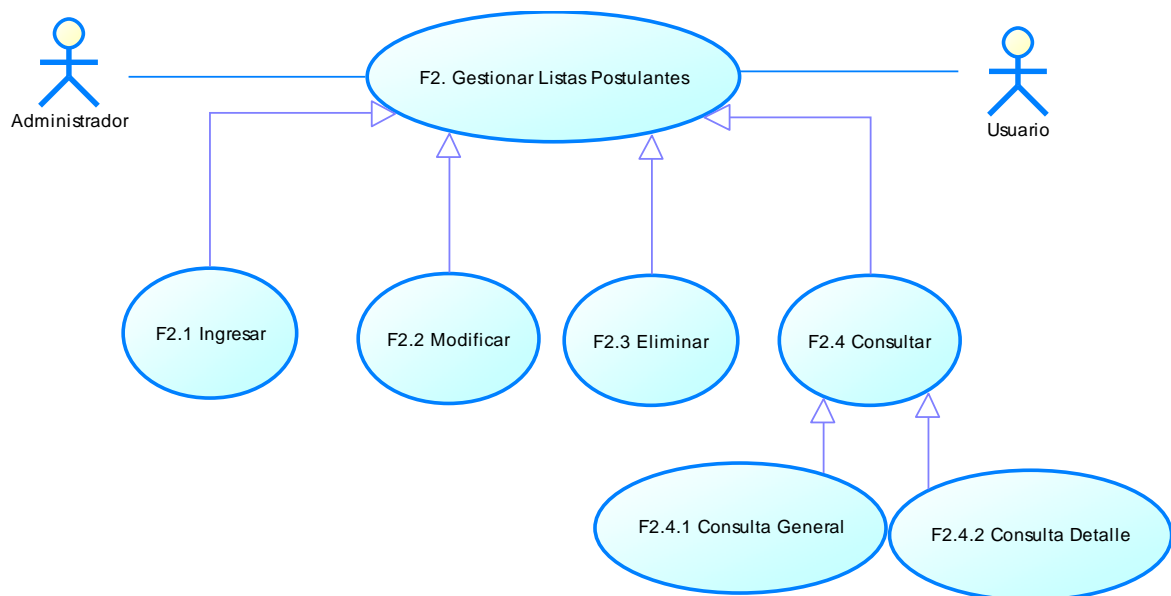


Diagrama 4 - Caso de Uso Gestionar Listas Postulantes

Elaborado por : Alex Calvopiña, Stephanie García; 2016

F1.1 Ingresar Listas Postulantes

Descripción: Permite realizar el ingreso de nuevas listas postulantes.

Actores: Administrador, Usuario

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.



Diagrama 5 - Caso de Uso Ingresar Listas Postulantes

Elaborado por : Alex Calvopiña, Stephanie García; 2016

Flujo Principal

1. Actor selecciona la opción Gestionar Listas
2. Actor selecciona la opción "Ingresar una Lista".
3. El sistema muestra la página de ingreso de Listas.
4. Actor ingresa datos de la Lista.
5. Actor selecciona botón guardar.
6. Sistema verifica el tipo de datos ingresados. (E1)
7. Sistema almacena datos y muestra mensaje.

Excepciones:

E1: Verificación de datos, campos no válidos. La solución volver a ingresar.

F1.2 Modificar Listas Postulantes

Descripción: Permite modificar los datos de las listas postulantes almacenadas en la Base de Datos, mediante la especificación de su nombre.

Actores: Administrador, Usuario



Diagrama 6 - Caso de Uso Modificar Listas Postulantes

Elaborado por : Alex Calvopiña, Stephanie García; 2016

Flujo Principal

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

1. Actor selecciona la opción "Modificar una Lista"
2. El sistema muestra la página de modificación de Postulantes.
3. Actor selecciona el Código de la lista.
4. Sistema consulta los datos de la lista.
5. Sistema muestra los datos de la lista a modificar.
6. Actor modifica el o los datos de la lista postulante.
7. Actor selecciona botón "Guardar".
8. Sistema verifica el tipo de datos ingresados. (E1)
9. Sistema almacena datos y muestra mensaje.

Excepciones:

E1: Verificación de datos, campos no válidos. La solución volver a ingresar.

F1.3 Eliminar Listas Postulantes

Descripción: Permite eliminar los datos de las listas postulantes almacenadas en la Base de Datos, mediante la especificación de su nombre.

Actores: Administrador, Usuario

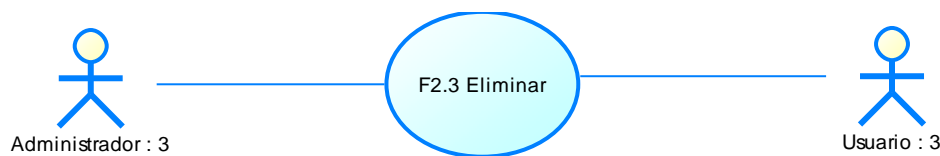


Diagrama 7 - Caso de Uso Eliminar Listas Postulantes

Elaborado por : Alex Calvopiña, Stephanie García; 2016

Flujo Principal

1. Actor selecciona la opción "Eliminar Lista"
2. El sistema muestra la página de Eliminación de Postulantes.

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

3. Actor ingresa el nombre de la lista.
4. Sistema comprueba el nombre. (F1)
5. Actor presiona opción "Eliminar".
6. Sistema elimina lista y candidatos asociados.

Flujo Alternativo:

F1: Si existe la lista postulante ir al Caso de Uso F1.1 o F1.2

F1.4.1 Consulta General Listas Postulantes

Descripción: Permite consultar todas las listas postulantes almacenadas en la Base de Datos.

Actores: Administrador, Usuario

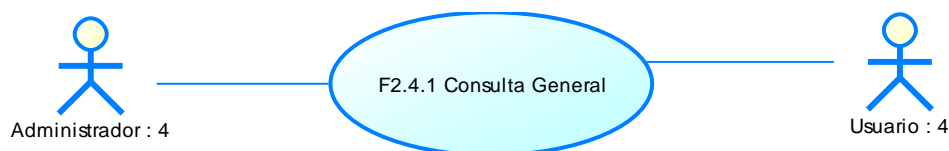


Diagrama 8 - Caso de Uso Consulta General Listas Postulantes

Elaborado por : Alex Calvopiña, Stephanie García; 2016

Flujo Principal

1. El actor selecciona la opción "Consultar Listas"
2. El sistema muestra la página de Consulta de Listas Postulantes.
3. El actor presiona el botón consultar.
4. El sistema muestra toda la información de las Listas Postulantes. (E1)

Excepciones

E1: Problemas con la base de datos. La solución contactar al administrador de la base de datos (DBA).

F1.4.2 Consulta a detalle Listas Postulantes

Descripción: Permite consultar la información de una lista postulante mediante una búsqueda por el nombre.

Actores: Administrador, Usuario

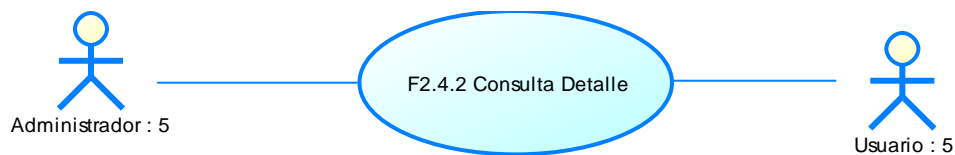


Diagrama 9 - Caso de Uso Consulta Detalle Listas Postulantes

Elaborado por : Alex Calvopiña, Stephanie García; 2016

Flujo Principal

1. El actor selecciona la opción " Consultar Listas "
2. El sistema muestra la página de Consulta de Listas Postulantes.
3. El actor ingresa el nombre de la lista que desea buscar.
4. El actor presiona el botón consultar. (F1)
5. El sistema muestra toda la información de la Lista Postulante especificada. (E1)

Flujo Alternativo

F1: Si el nombre no existe ir al caso de uso: F1.1

Excepciones

E1: Problemas con la base de datos. La solución contactar al administrador de la base de datos (DBA).

F2. Mostrar Información de las Listas

Descripción: El actor puede observar toda la información de las listas postulantes y sus candidatos.

Actores: Administrador, Usuario y Votante

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

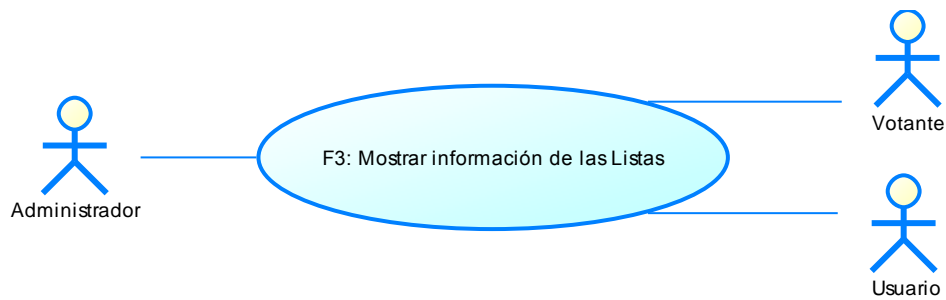


Diagrama 10 - Caso de Uso Mostrar Información de las Listas

Elaborado por : Alex Calvopiña, Stephanie García; 2016

Flujo Principal

1. El actor ingresa al sistema.
2. El sistema muestra toda la información de cada Lista Postulante y Candidatos.

3.3.2. Diseño

3.3.2.1. Diagrama de Clases

Un Diagrama de Clases es un diagrama que nos permite representar un conjunto de interfaces, colaboraciones y la manera como estas interactúan o se relacionan.

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.



Diagrama 11 - Diagrama de Clases

Elaborado por : Alex Calvopiña, Stephanie García; 2016

3.3.2.2. Diagramas de Secuencia

Una vez expresados los diagramas de Caso de Uso a Detalle de las Funcionalidades: F0, F1 y F2 se desplegará los diagramas de secuencia de cada funcionalidad efectuada en la iteración. Los diagramas de secuencia nos permiten resaltar la secuencia de las acciones, las interacciones del sistema entre sus capas y los mensajes que son enviados entre ellas.

F1. Verificar Usuario y Contraseña

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

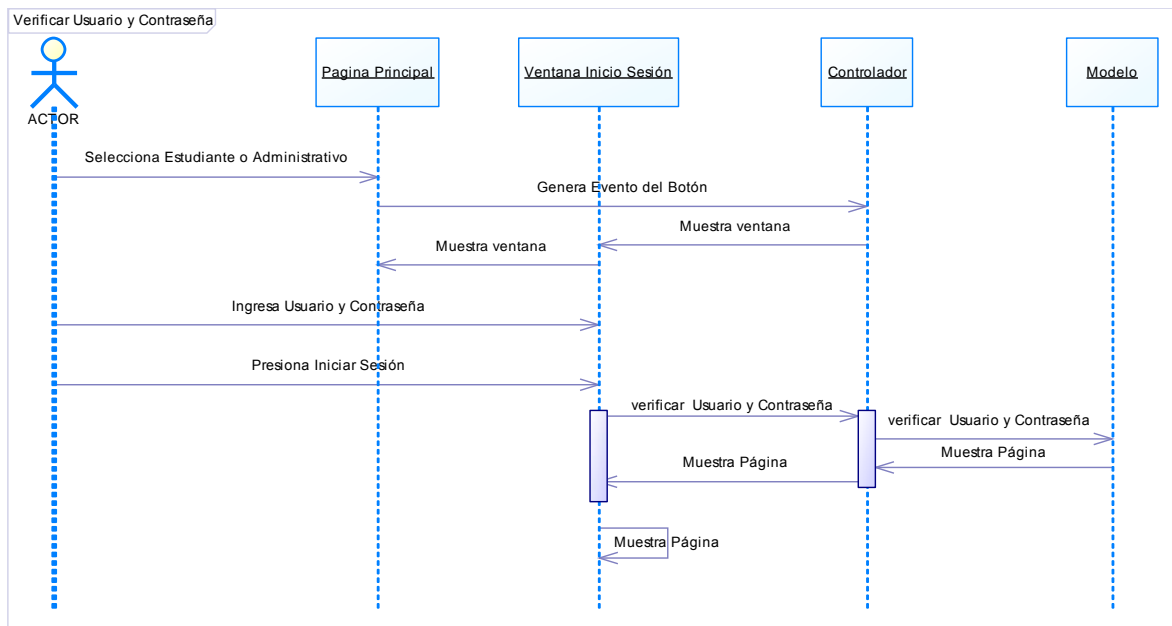


Diagrama 12 - Secuencia Verificar Usuario y Contraseña

Elaborado por : Alex Calvopiña, Stephanie García; 2016

F1: Gestionar Listas Postulantes

F1.1 Ingresar Listas Postulantes

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

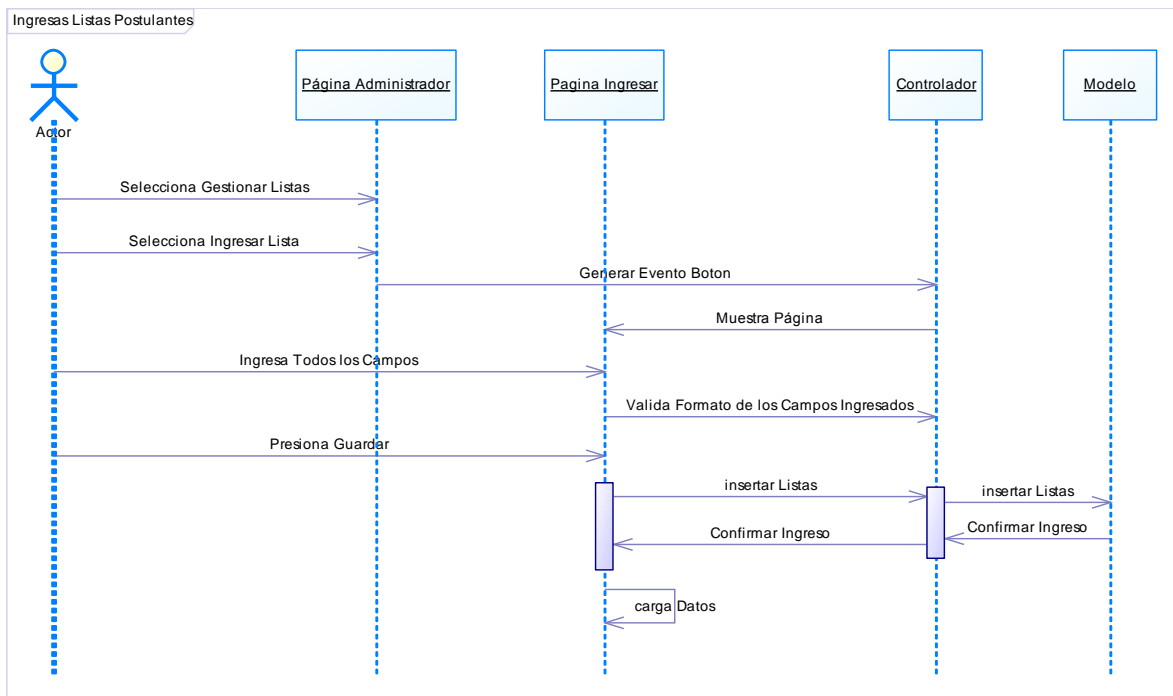


Diagrama 13 - Secuencia Ingresar Listas Postulantes

Elaborado por : Alex Calvopiña, Stephanie García; 2016

F1.2 Modificar Listas Postulantes

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

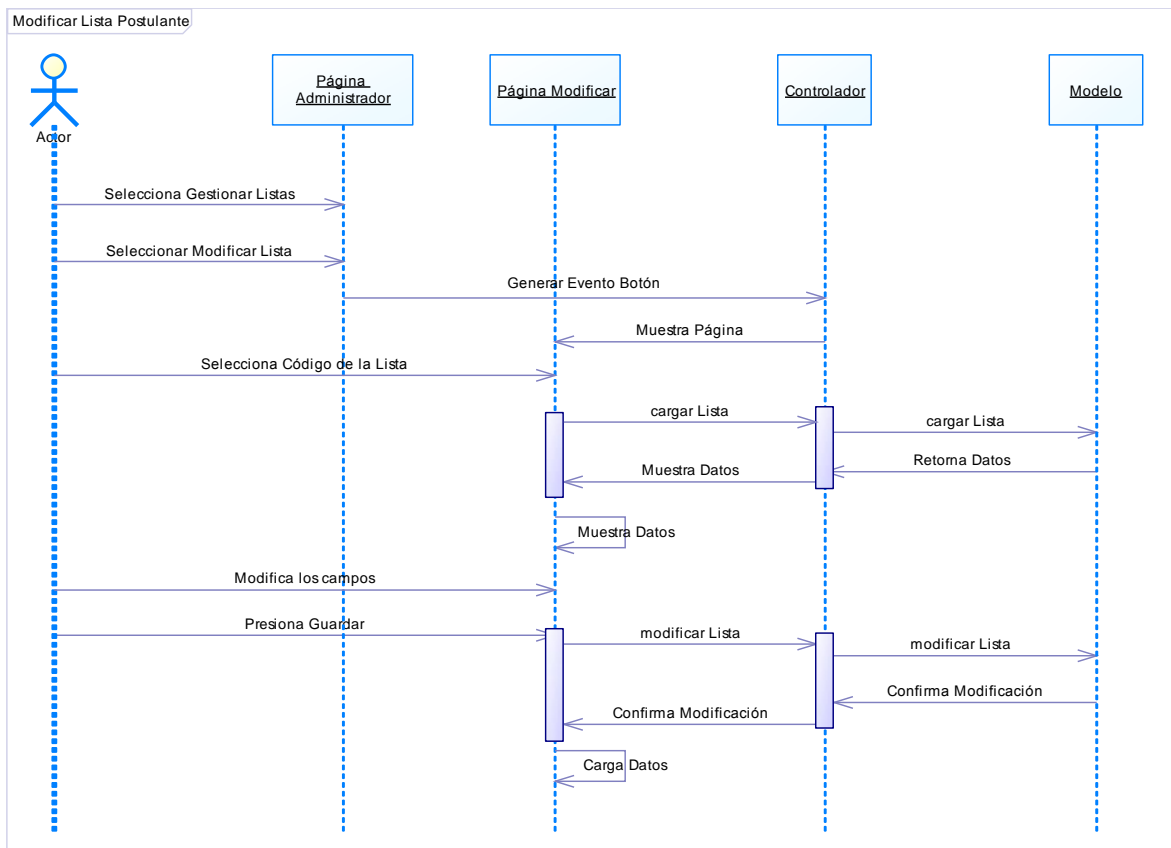


Diagrama 14 - Secuencia Modificar Listas Postulantes

Elaborado por : Alex Calvopiña, Stephanie García; 2016

F1.3 Eliminar Listas Postulantes

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

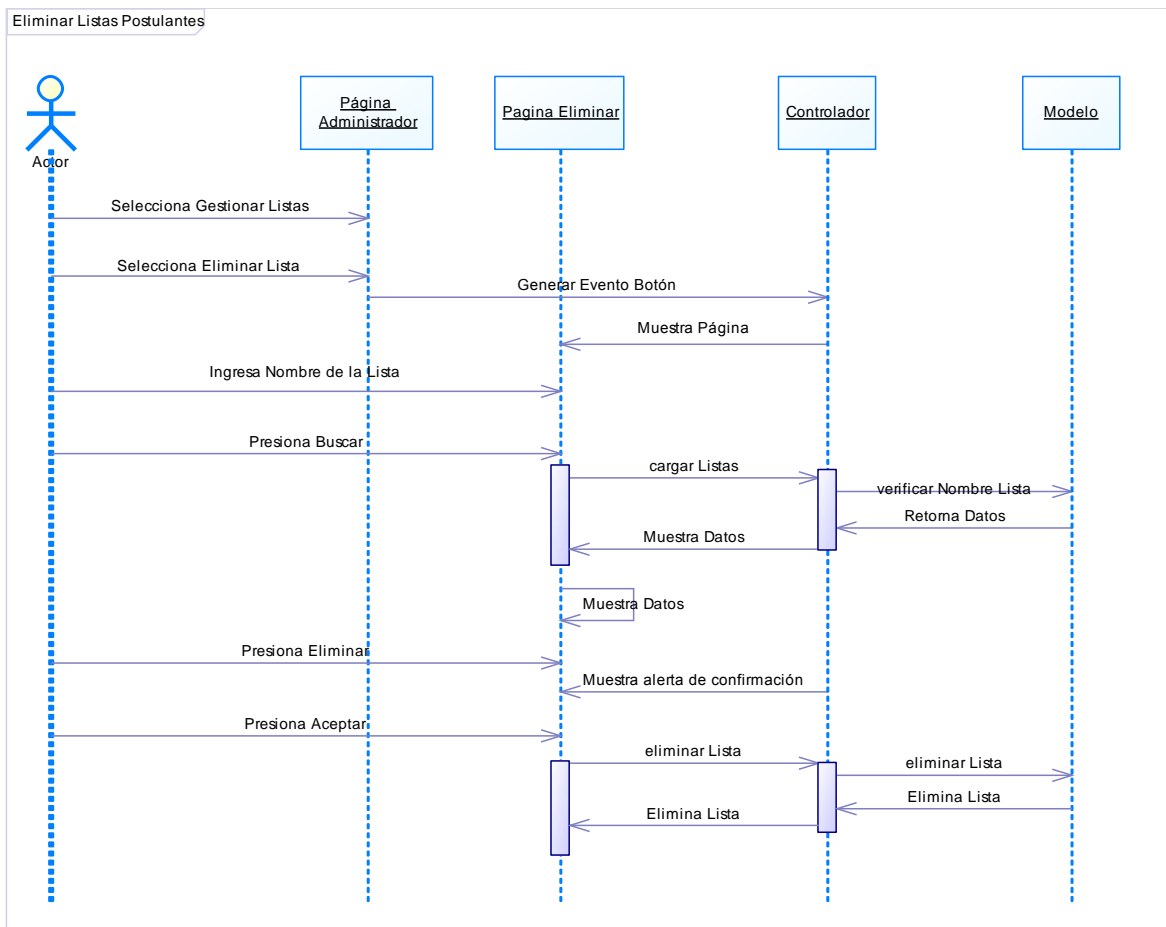


Diagrama 15 - Secuencia Eliminar Listas Postulantes

Elaborado por : Alex Calvopiña, Stephanie García; 2016

F1.4. Consultar

F1.4.1 Consulta General Listas Postulantes

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

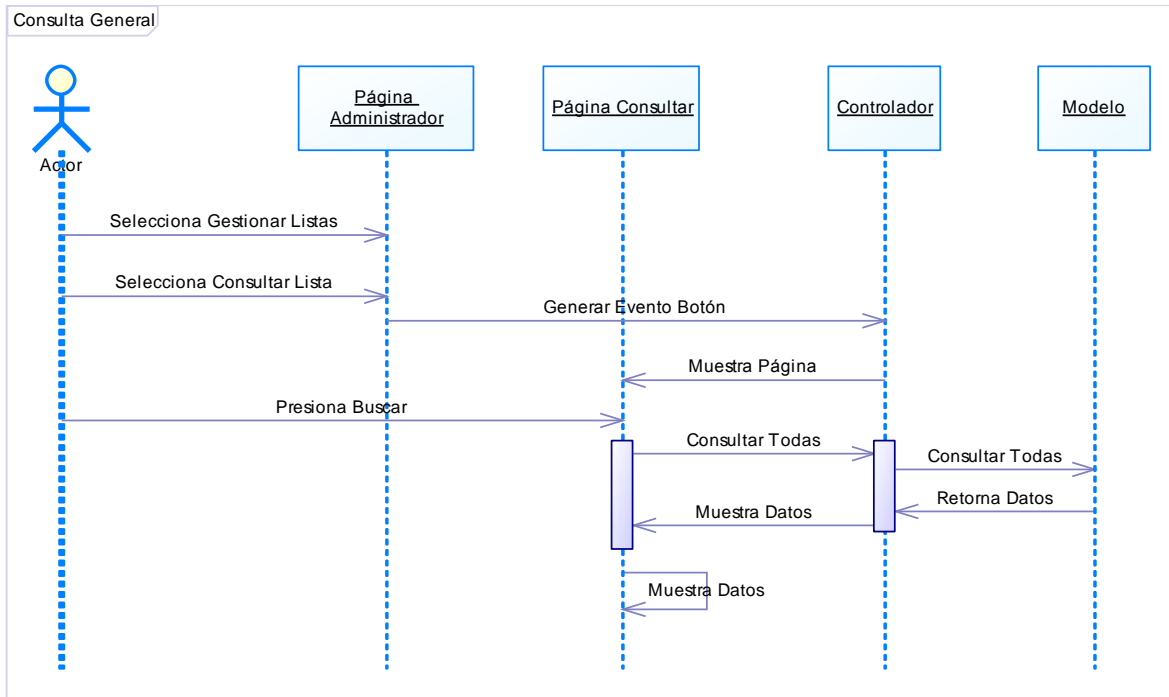


Diagrama 16 - Secuencia Consulta General Listas Postulantes

Elaborado por : Alex Calvopiña, Stephanie García; 2016

F1.4.2 Consulta Detalle Listas Postulantes

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

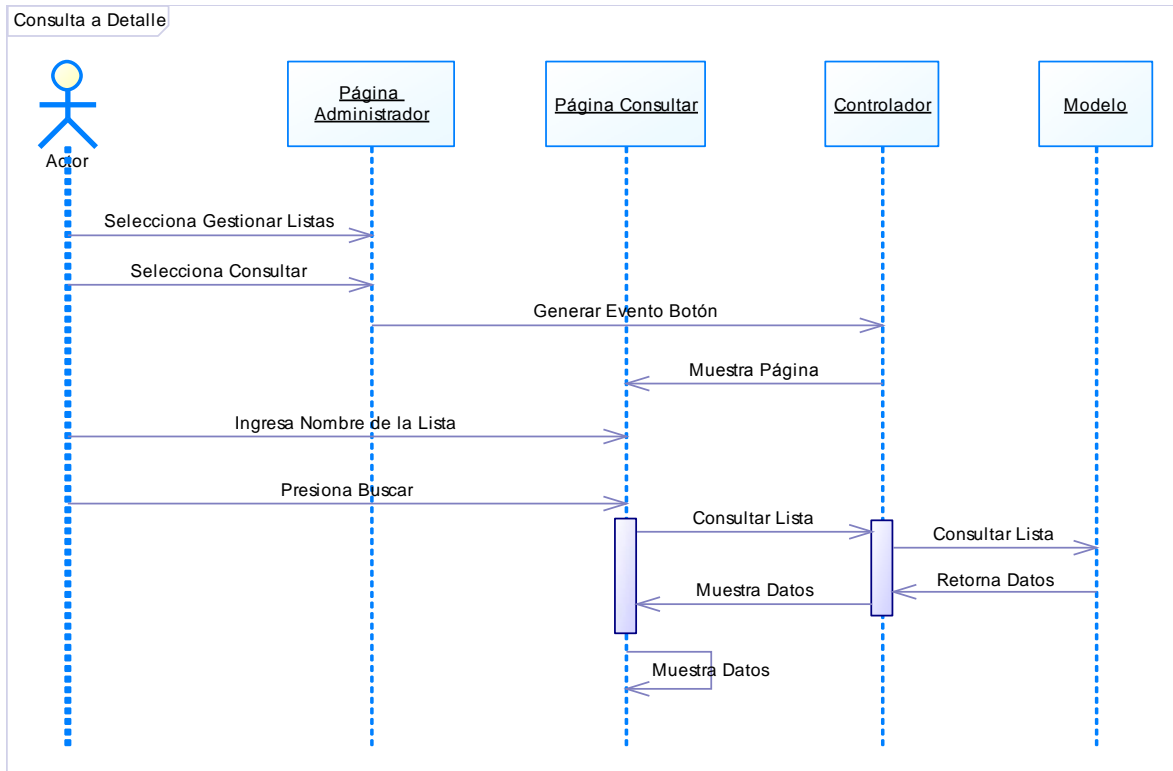


Diagrama 17 - Secuencia Consulta Detalle Listas Postulantes

Elaborado por : Alex Calvopiña, Stephanie García; 2016

F2: Mostrar Información de las Listas

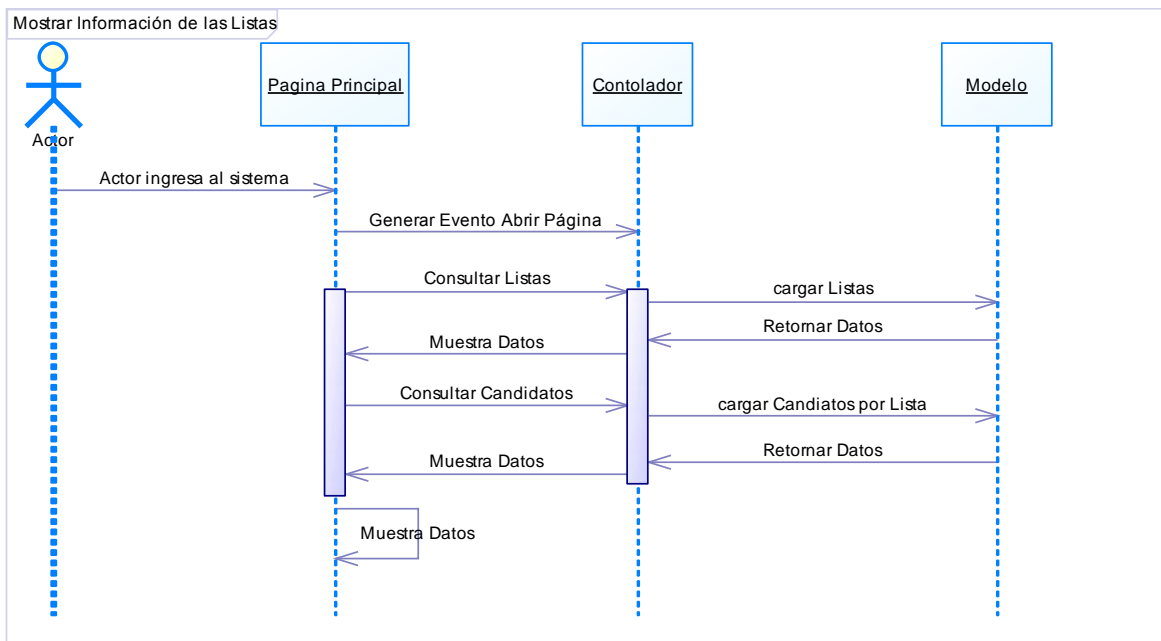


Diagrama 18 - Secuencia Mostrar Información

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

Elaborado por : Alex Calvopiña, Stephanie García; 2016

3.3.2.3. *Prototipos*

F2: Mostrar Información de las Listas

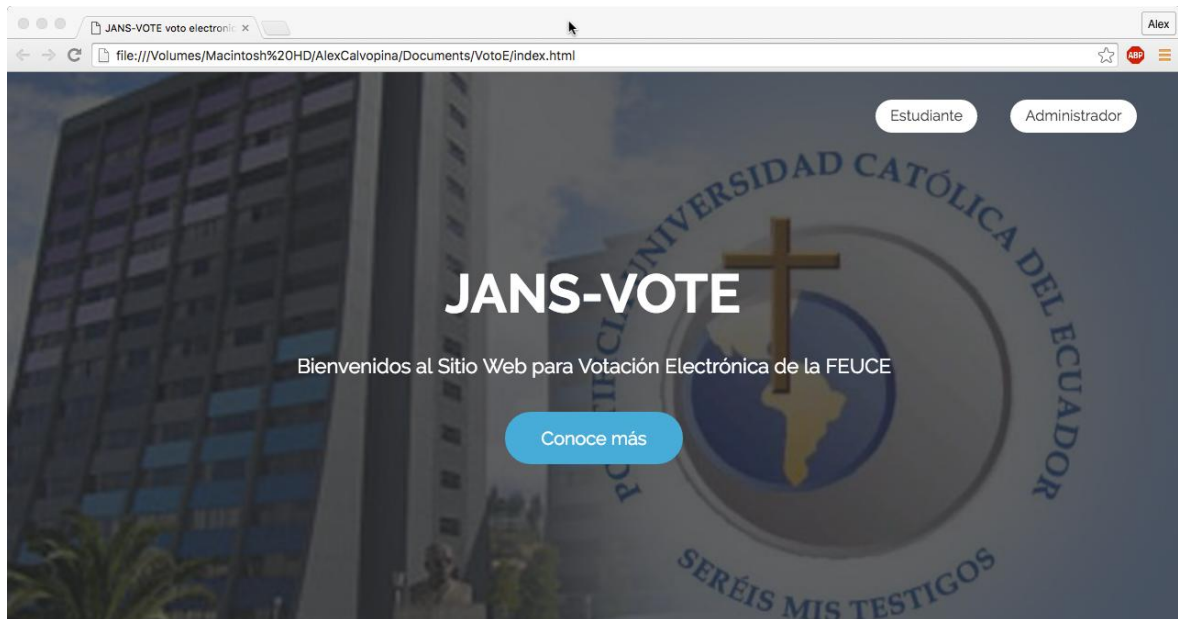


Ilustración 19 - Mostrar Información de las Listas



Ilustración 20 - Información Listas Postulantes

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.



Ilustración 21 - Mostrar Candidatos



Ilustración 22 - Mostrar Información Candidatos

F0: Verificar Usuario y Contraseña

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

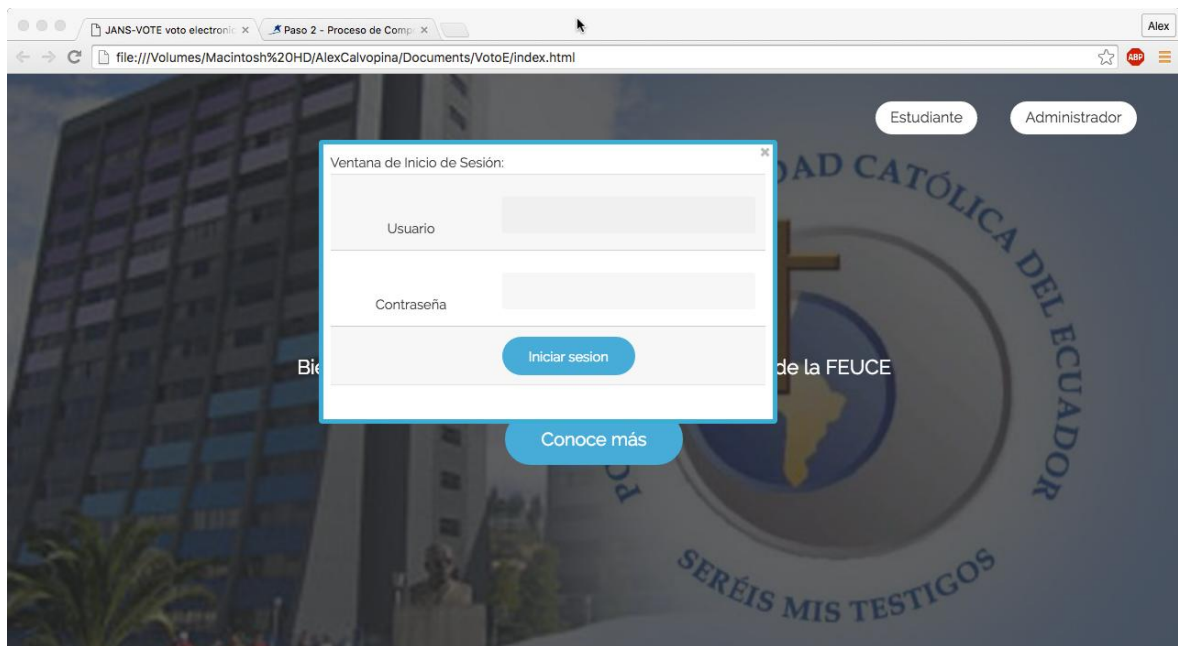


Ilustración 23 - Verificar Usuario y Contraseña

F2: Gestionar Listas Postulantes

F1.1 Ingresar Listas Postulantes

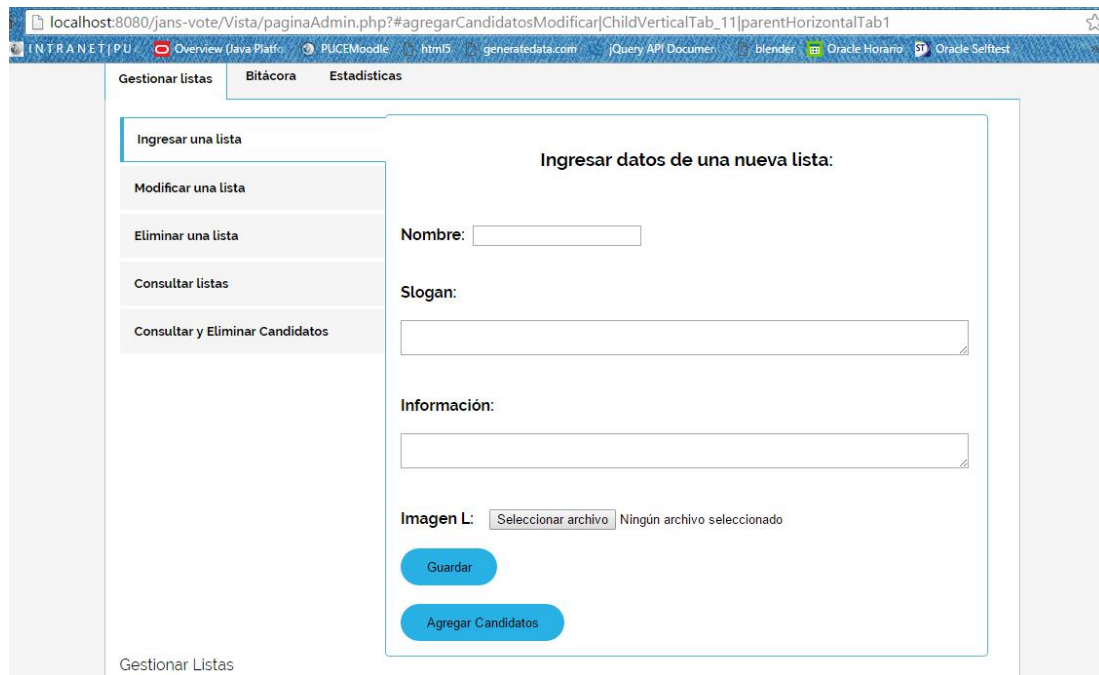
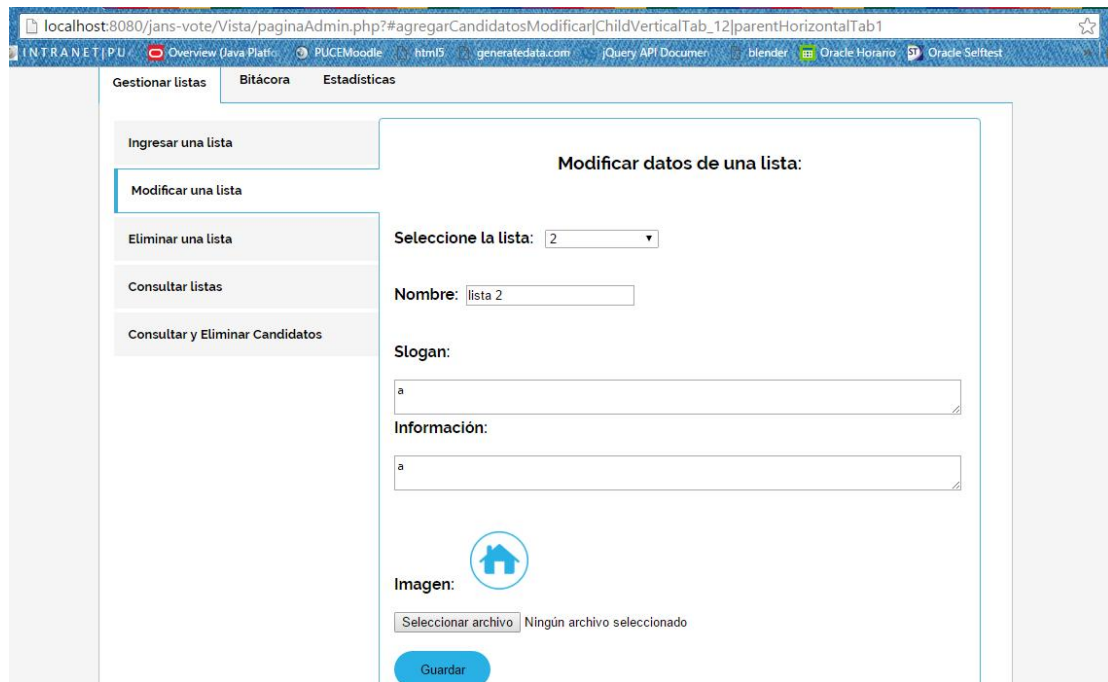


Ilustración 24 - Ingresar Listas Postulantes

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

F1.2 Modificar Listas Postulantes



The screenshot shows a web browser window with the URL `localhost:8080/jans-vote/Vista/paginaAdmin.php?#agregarCandidatosModificar[ChildVerticalTab_12]parentHorizontalTab1`. The browser's address bar and tabs are visible. The page content includes a navigation menu with 'Gestionar listas', 'Bitácora', and 'Estadísticas'. A sidebar on the left contains buttons for 'Ingresar una lista', 'Modificar una lista', 'Eliminar una lista', 'Consultar listas', and 'Consultar y Eliminar Candidatos'. The main content area is titled 'Modificar datos de una lista:' and contains the following form elements: a dropdown menu for 'Seleccione la lista:' with the value '2'; a text input for 'Nombre:' containing 'lista 2'; a text input for 'Slogan:' containing 'a'; a text input for 'Información:' containing 'a'; an 'Imagen:' section with a home icon, a 'Seleccionar archivo' button, and the text 'Ningún archivo seleccionado'; and a blue 'Guardar' button at the bottom.

Ilustración 25 - Modificar Listas Postulantes

F1.3 Eliminar Listas Postulantes

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

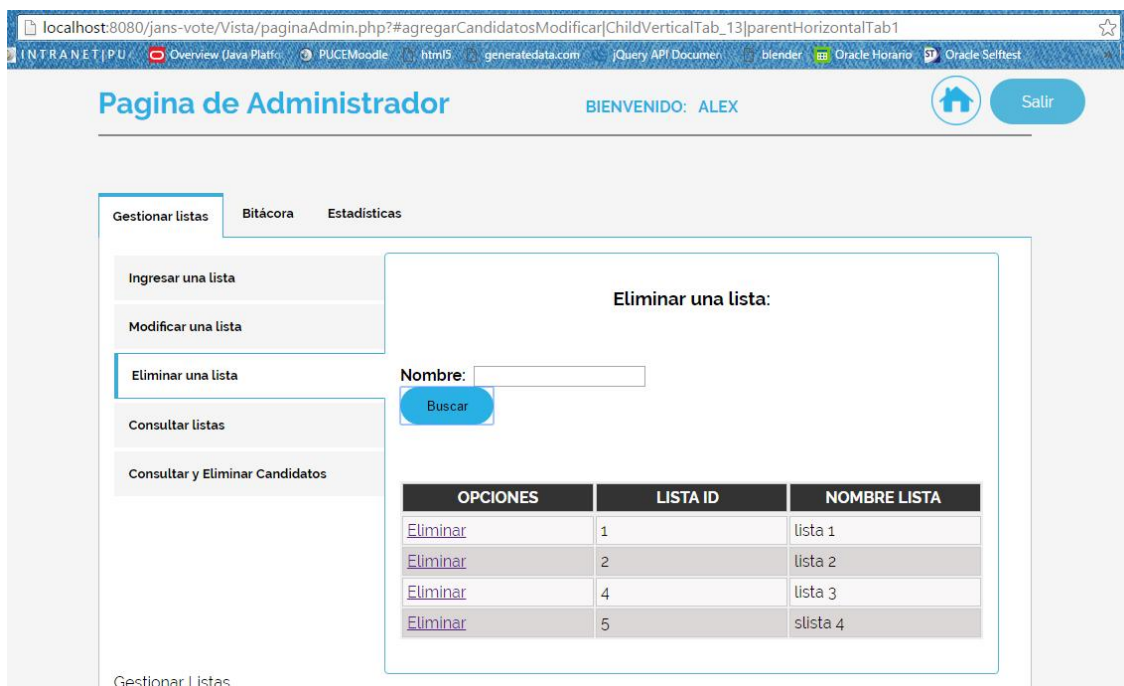


Ilustración 26 - Eliminar Listas Postulantes

F1.4.1 Consulta General Listas Postulantes

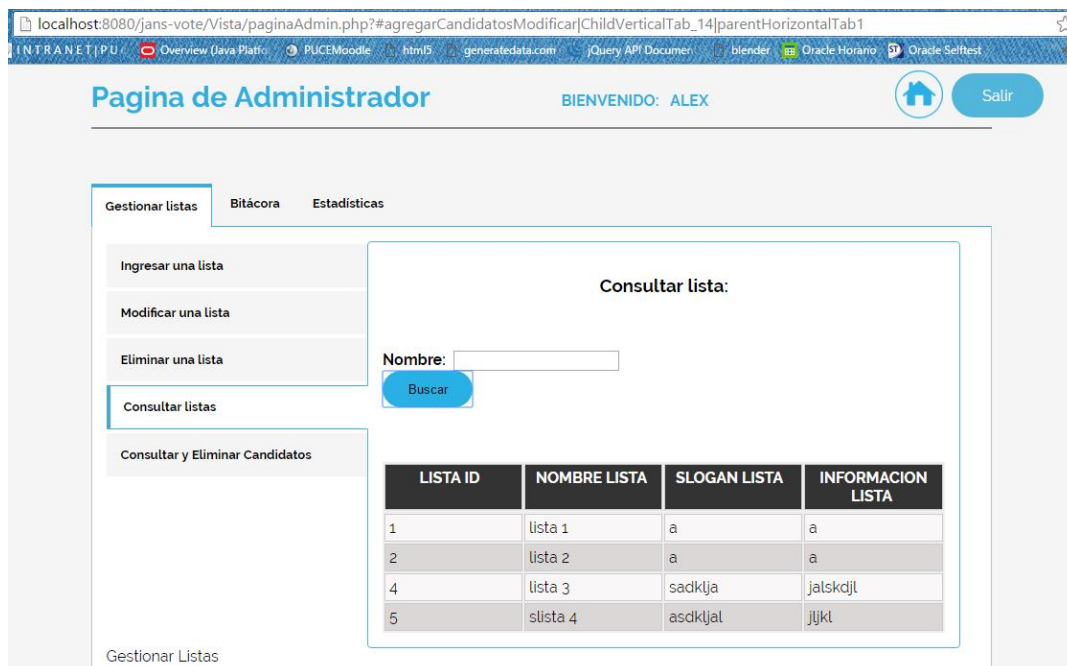


Ilustración 27 - Consulta General Lista Postulantes

F1.4.2 Consulta Detalle Listas Postulantes

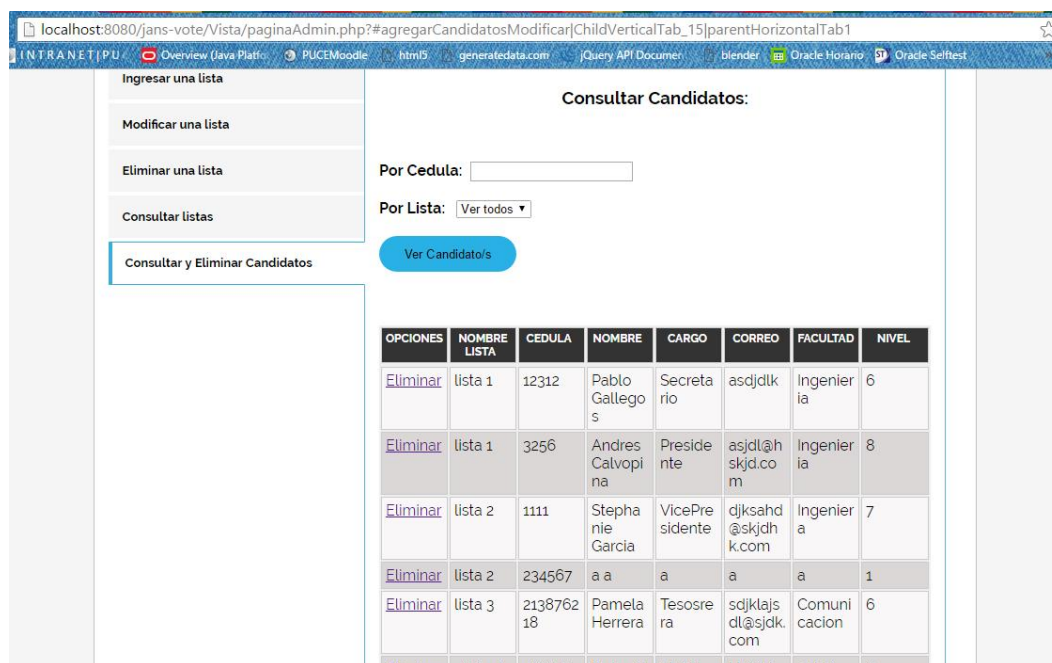


Ilustración 28 - Consulta Detalle Listas Postulantes

3.3.3. Desarrollo

Para el desarrollo de la aplicación, se usan como base los diagramas descritos en el diseño, para la primera iteración se realizó el desarrollo de las siguientes funcionalidades:

F0: Verificar Usuario y Contraseña.

F1: Gestión de Listas: Ingresar, Modificar, Eliminar y Consultar Listas.

F2: Mostrar información de las Listas.

El código fuente completo se encontrará en el CD.

3.3.4. Pruebas

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

Las pruebas nos permiten obtener una visión objetiva sobre el correcto funcionamiento del software, en este caso se realizarán pruebas a las funcionalidades desarrolladas en la primera iteración.

3.3.4.1. Informe de Pruebas

Las pruebas de la primera iteración nos permiten conocer los errores de la aplicación o las mejoras que se pueden implementar para incrementar el desempeño y facilitar el uso del sistema. Aquellos errores que se encuentren durante el proceso de pruebas se les realizará una depuración en la siguiente iteración.

Los sujetos que se encargaran de usar el sistema y probar sus funcionalidades son aquellos clientes, considerados usuario final del sistema, en el caso de este producto son los integrantes de la FEUCE.

Las funcionalidades que se probaron en la primera iteración son:

- F0: Inicio de sesión mediante usuario y contraseña tanto de un Votante como de un Administrador.
- F1: Gestión de listas postulantes: Ingreso, modificación, eliminación, consulta de una o varias listas.
- F2: Mostrar información de las listas: Muestra de forma interactiva en la pantalla de inicio de la pagina web, la información de cada lista.

Resultados:

Funcionalidad	Sugerencias	Aceptación
F0: Verificar Usuario y Contraseña		
F1: Gestión de Listas Postulantes	<ul style="list-style-type: none">• Mostrar por default las tablas de las listas.	<ul style="list-style-type: none">• Se realizará una mejora al mostrar las tablas consultar listas.
F2: Mostrar Información de Listas.	<ul style="list-style-type: none">• Añadir íconos de la PUCE y	<ul style="list-style-type: none">• Se añadirá los íconos y los links en el pie de

	FEUCE en el pie de la página. <ul style="list-style-type: none">• Añadir links a las páginas o redes sociales de la PUCE y FEUCE	la página.
--	--	------------

Tabla 5 - Informe Resultados Iteración 1

Elaborado por: Alex Calvopiña, Stephanie García; 2016

Las sugerencias realizadas por los estudiantes que forman parte de la FEUCE fueron acogidas como se puede observar en la tabla de resultados. Estas sugerencias serán implementadas en durante el desarrollo de la segunda iteración.

3.4. Segunda Iteración

De acuerdo a la planificación realizada se definió un desarrollo por iteraciones incrementales, en el cual dentro de cada iteración se define los requerimientos funcionales y se realiza el diseño, la implementación y las pruebas de los módulos o funcionalidades planeadas para la iteración.

En la segunda iteración además de tomar en cuenta las observaciones obtenidas de la primera iteración y se desarrolla la Gestión de Candidatos que permite Ingresar, Modificar, Eliminar y Consultar los Candidatos, esta funcionalidad será agregada dentro de la página del Administrador.

A su vez se desarrolla la página de votante, en la cual se presentan los candidatos por cargo, para que el votante puede dar su voto por cada uno de ellos o pueda acceder a otro tipo de opciones de votación como el voto Nulo o Blanco.

Para Finalizar esta iteración se implementa la página de Bitácora dentro de las funcionalidades a las que puede acceder el administrador, en el cual puede observar los detalles de la Votación.

3.4.1. Requerimientos Funcionales

3.4.1.1. Casos de Uso

En cada iteración y empleando como base al Diagrama General desarrollado en los Requerimientos Funcionales posteriormente se presentan los Diagramas de Caso de Uso Detallados de las funcionalidades que serán desarrollados a lo largo de la segunda iteración. Estos diagramas no permiten realizar una representación del flujo de las acciones que realiza cada tipo de actor durante la manipulación del sistema y de cada una de las funcionalidades.

F3. Gestionar Candidatos

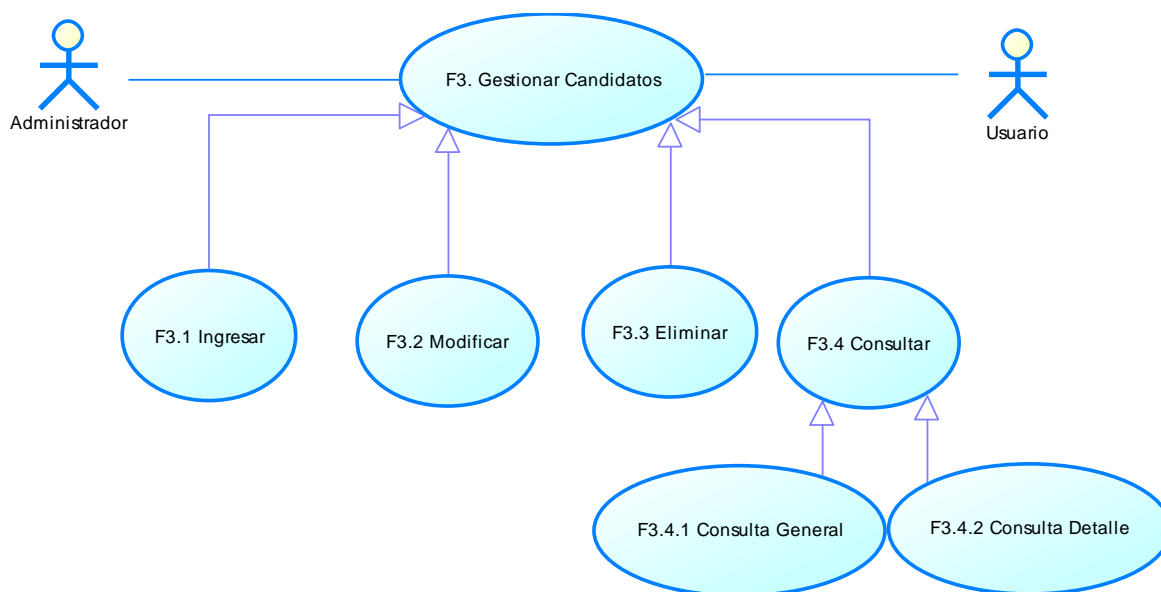


Diagrama 19 - Caso de Uso Gestionar Candidatos

Elaborado por: Alex Calvopiña, Stephanie García; 2016

F3.1 Ingresar Candidatos

Descripción: Permite realizar el ingreso de nuevos candidatos.

Actores: Administrador, Usuario

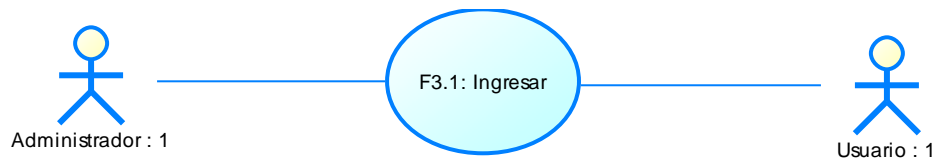


Diagrama 20 - Caso de Uso Ingresar Candidatos

Elaborado por: Alex Calvopiña, Stephanie García; 2016

Flujo Principal

1. Actor selecciona la opción Gestionar Listas
2. Actor selecciona la opción "Ingresar una Lista".
3. El sistema muestra la página de ingreso de Listas.
4. Actor ingresa datos de la Lista.
5. Actor selecciona botón "Agregar Candidato".
6. Actor Ingresa datos del Candidato.
7. Sistema verifica el tipo de datos ingresados. (E1)
8. Sistema almacena datos y muestra mensaje.

Excepciones:

E1: Verificación de datos, campos no válidos. La solución volver a ingresar.

F1.2 Modificar Candidatos

Descripción: Permite modificar los datos de los candidatos almacenadas en la Base de Datos.

Actores: Administrador, Usuario

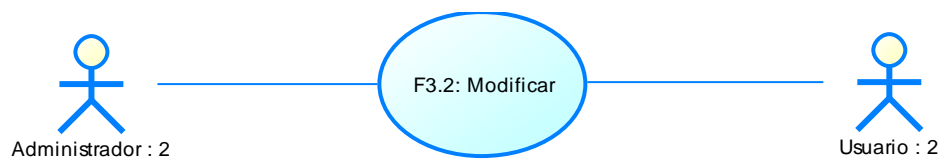


Diagrama 21 - Caso de Uso Modificar Candidatos

Elaborado por: Alex Calvopiña, Stephanie García; 2016

Flujo Principal

1. Actor selecciona la opción "Modificar una Lista"
2. El sistema muestra la página de modificación de Postulantes.
3. Actor selecciona el Código de la lista.
4. Sistema consulta los datos de la lista.
5. Sistema muestra los datos de la lista.
6. Actor selecciona botón "Modificar Candidatos".
7. Sistema muestra los candidatos pertenecientes a la lista.
8. Actor modifica los datos.
9. Sistema verifica el tipo de datos ingresados. (E1)
10. Sistema almacena datos y muestra mensaje.

Excepciones:

E1: Verificación de datos, campos no válidos. La solución volver a ingresar.

F1.3 Eliminar Candidatos

Descripción: Permite eliminar los datos de los candidatos almacenadas en la Base de Datos, mediante la especificación de su cedula.

Actores: Administrador, Usuario

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

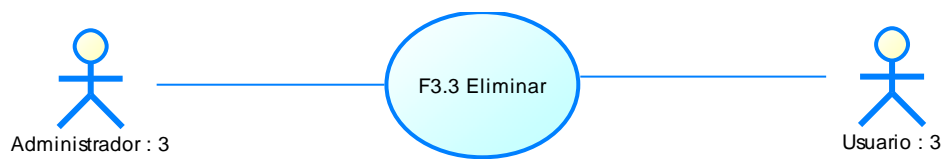


Diagrama 22 - Caso de Uso Eliminar Candidatos

Elaborado por: Alex Calvopiña, Stephanie García; 2016

Flujo Principal

1. Actor selecciona la opción "Eliminar y Consultar Candidatos"
2. El sistema muestra la página de Eliminación y Consulta de Candidatos.
3. Actor ingresa el numero de cédula del candidato.
4. Sistema comprueba el número de cédula. (F1)
5. Actor presiona opción "Eliminar".
6. Sistema elimina al candidato.

Flujo Alternativo:

F1: Si no existe el candidato ir al Caso de Uso F3.1 o F3.2

F1.4.1 Consulta General Candidato

Descripción: Permite consultar todos los candidatos almacenadas en la Base de Datos.

Actores: Administrador, Usuario

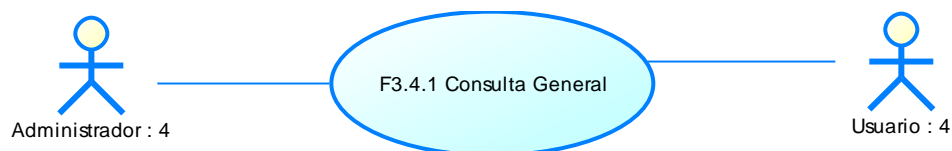


Diagrama 23 - Caso de Uso Consulta General Candidatos

Elaborado por: Alex Calvopiña, Stephanie García; 2016

Flujo Principal

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

1. El actor selecciona la opción "Consultar y Eliminar Candidatos"
2. El sistema muestra la página de Consulta de Candidatos.
3. El actor presiona el botón "Ver Candidato/s".
4. El sistema muestra información de los Candidatos. (E1)

Excepciones

E1: Problemas con la base de datos. La solución contactar al administrador de la base de datos (DBA).

F1.4.2 Consulta a detalle Candidatos

Descripción: Permite consultar la información de un candidato mediante una búsqueda por el nombre.

Actores: Administrador, Usuario

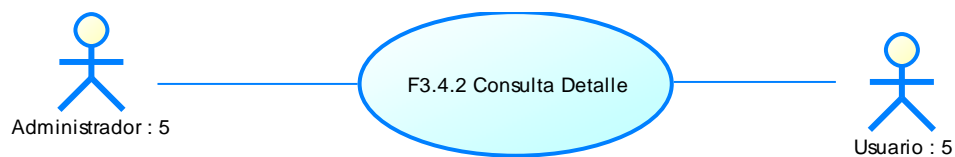


Diagrama 24 - Caso de Uso Consulta Detalle Candidato

Elaborado por: Alex Calvopiña, Stephanie García; 2016

Flujo Principal

1. El actor selecciona la opción " Consultar y Eliminar Candidatos "
2. El sistema muestra la página de Consulta de Candidatos.
3. El actor ingresa la cédula del candidato que desea buscar.
4. El actor presiona el botón consultar. (F1)
5. El sistema muestra la información del candidato especificado. (E1)

Flujo Alterno

F1: Si el candidato no existe ir al caso de uso: F3.1

Excepciones

E1: Problemas con la base de datos. La solución contactar al administrador de la base de datos (DBA).

F4. Votar

Descripción: Permite al actor realizar el proceso de sufragio.

Actores: Administrador, Votante

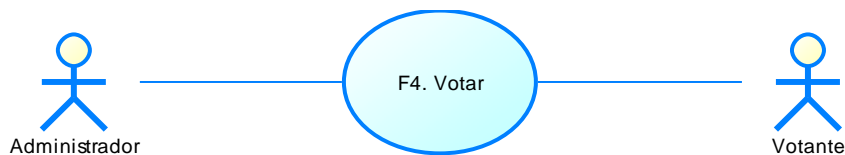


Diagrama 25 - Caso de Uso Votar

Elaborado por: Alex Calvopiña, Stephanie García; 2016

Flujo Principal

1. El actor ingresa a la página de votaciones.
2. El sistema muestra los candidatos de acuerdo al cargo. (E1)
3. El actor presiona votar en el o los candidatos a los que desea dar su voto.
4. El actor presiona el botón "Finalizar Voto".
5. El sistema verifica que el actor haya finalizado la votación.
6. El sistema almacena el puntaje de cada candidato.

Excepciones:

E1: Problemas con la base de datos. La solución llamar al administrador de la base de datos (DBA).

F5. Almacenar Detalles del Votante

Descripción: Sistema almacena una bitácora con los datos relevantes acerca del sufragio, como lo son sus datos, la hora y la ubicación desde donde se realizó la votación. (No presenta datos relacionados acerca de por quién o a quien dispuso su voto)

Actores: Administrador, Usuario

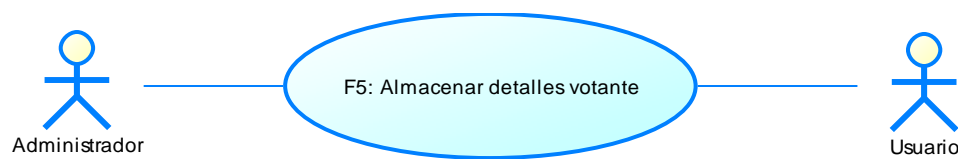


Diagrama 26 - Caso de Uso Almacenar Detalles del Votante

Elaborado por: Alex Calvopiña, Stephanie García; 2016

Flujo Principal:

1. Votante presiona "Finalizar Voto"
2. Sistema recibe el evento del botón "Votar". (E1)
3. Sistema almacena los datos de la Bitácora.
4. Actor ingresa en la pestaña de Bitácora.
5. Sistema muestra de la página de Bitácora.
6. Actor ingresa el numero de cédula del Votante y el periodo.
7. Sistema muestra la información. (E2)

Excepciones:

E1: Problemas al generar el evento del botón "Votar".

Solución:

- Comprobar conexión a internet.
- Comprobar el correcto ingreso de los campos.

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

- Contactar con el desarrollador.

E2: El votante buscado aun no ha realizado el proceso de sufragio.

3.4.2. Diseño

3.4.2.1. Diagrama de Clases

El Diagrama de Clases incorpora un conjunto de interfaces y colaboraciones permitiendo demostrar cómo se corresponden entre las clases.



Diagrama 27 - Diagrama de Clases

Elaborado por: Alex Calvopiña, Stephanie García; 2016

3.4.2.2. Diagramas de Secuencia

Después de elaborar los diagramas de Caso de Uso a Detalle de las Funcionalidades: F3, F4 y F5 se desarrollará los diagramas de secuencia de cada funcionalidad efectuada en esta iteración. Un diagrama de secuencia permite destacar la secuencia en que son realizadas las interacciones del sistema.

F3. Gestionar Candidatos

F3.1 Ingresar Candidatos

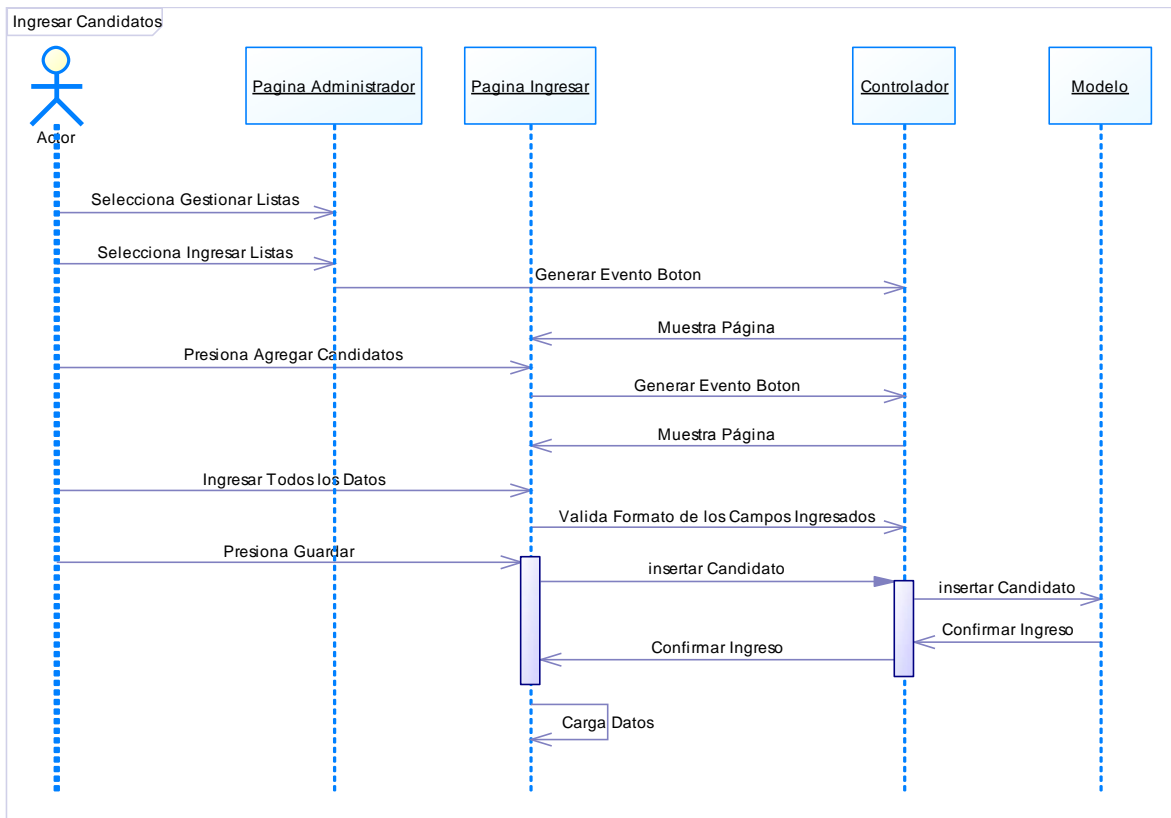


Diagrama 28 - Secuencia Ingresar Candidatos

Elaborado por: Alex Calvopiña, Stephanie García; 2016

F3.2 Modificar Candidatos

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

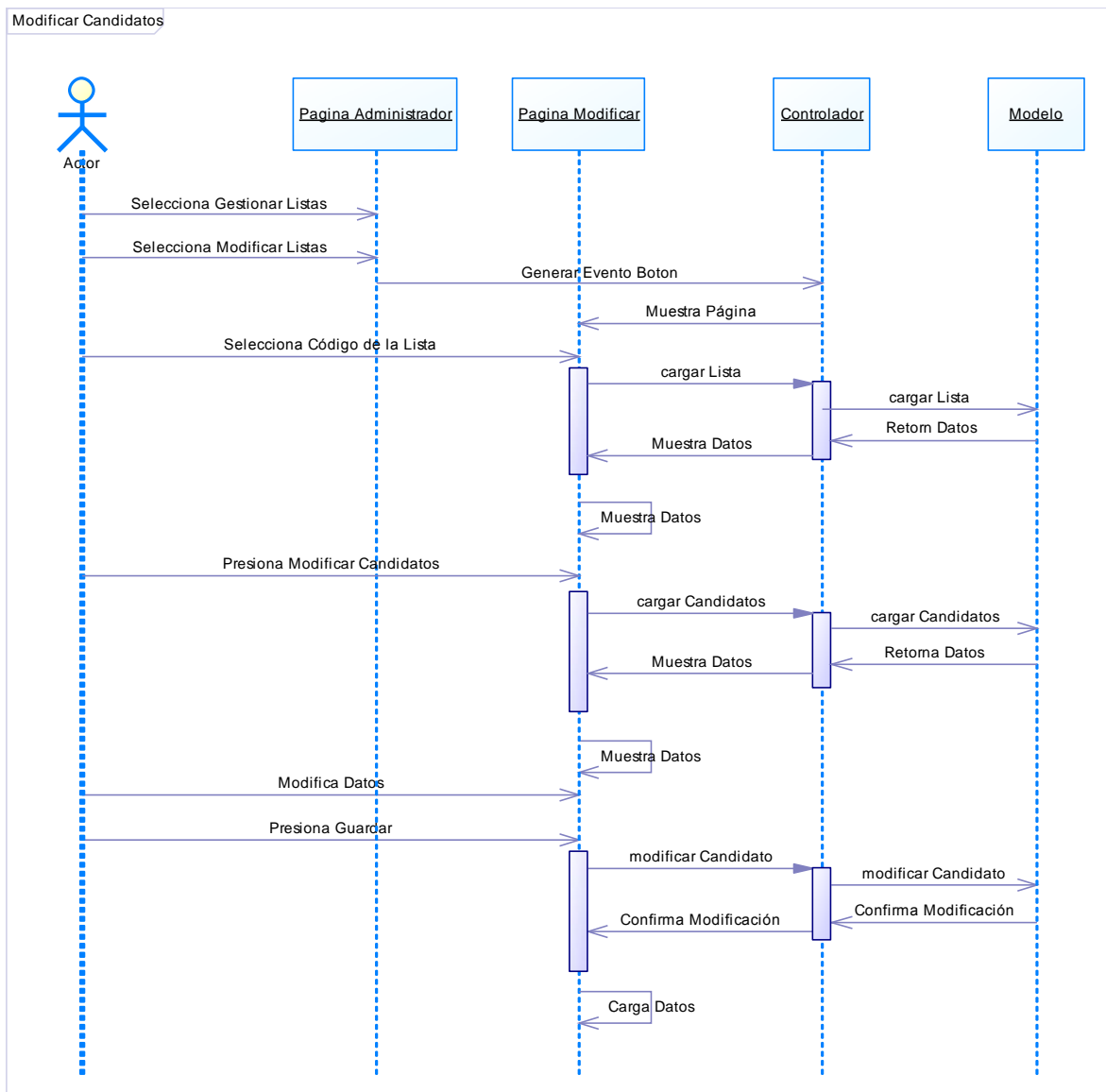


Diagrama 29 - Secuencia Modificar Candidatos

Elaborado por: Alex Calvopiña, Stephanie García; 2016

F3.3 Eliminar Candidatos

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

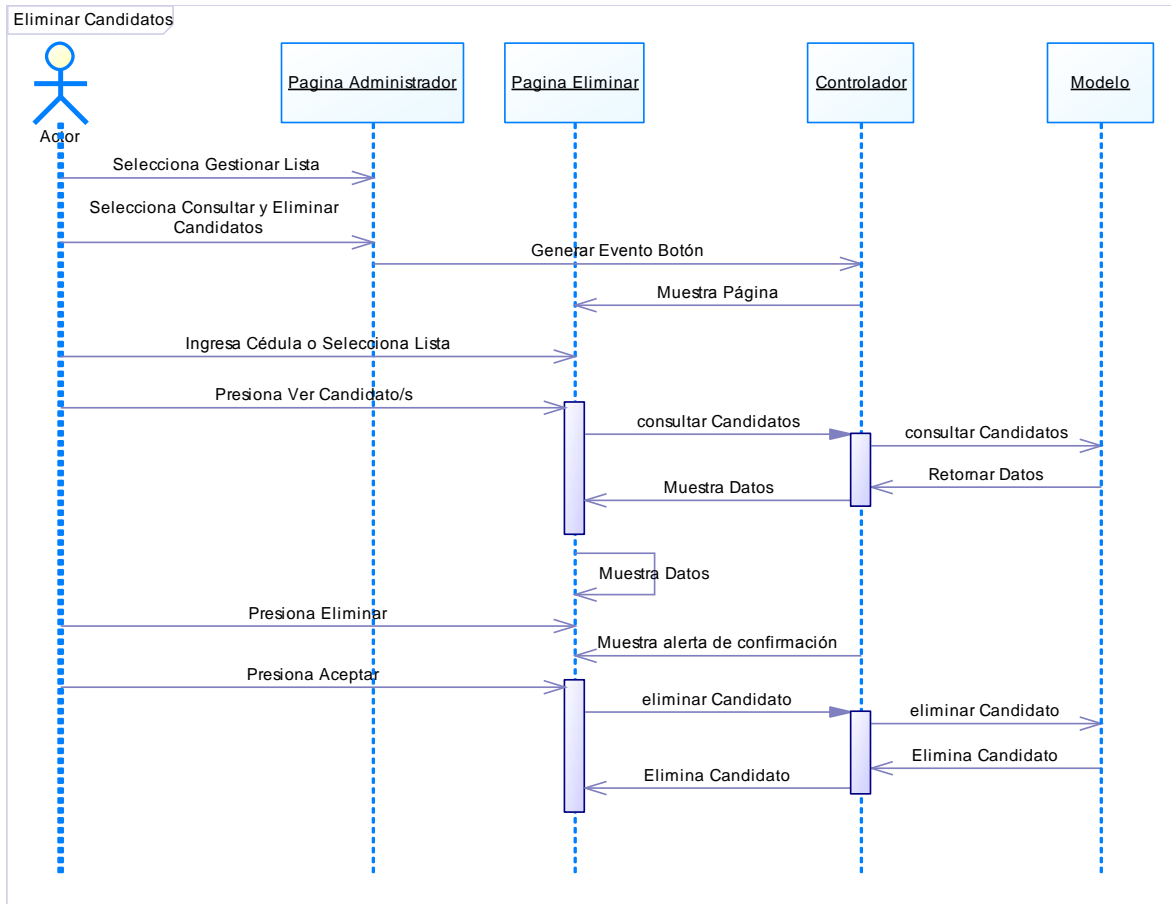


Diagrama 30 - Secuencia Eliminar

Elaborado por: Alex Calvopiña, Stephanie García; 2016

F3.4 Consultar Candidatos

F3.4.1 Consulta General

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

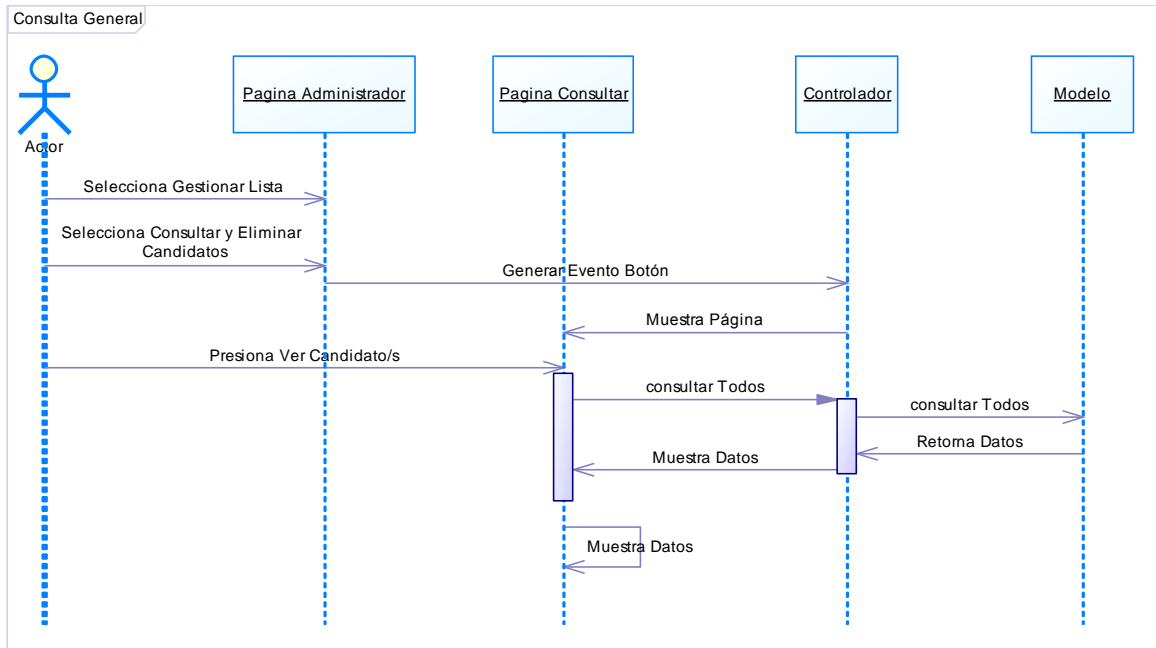


Diagrama 31 - Secuencia Consulta General Candidatos

Elaborado por: Alex Calvopiña, Stephanie García; 2016

F3.4.2 Consulta Detalle

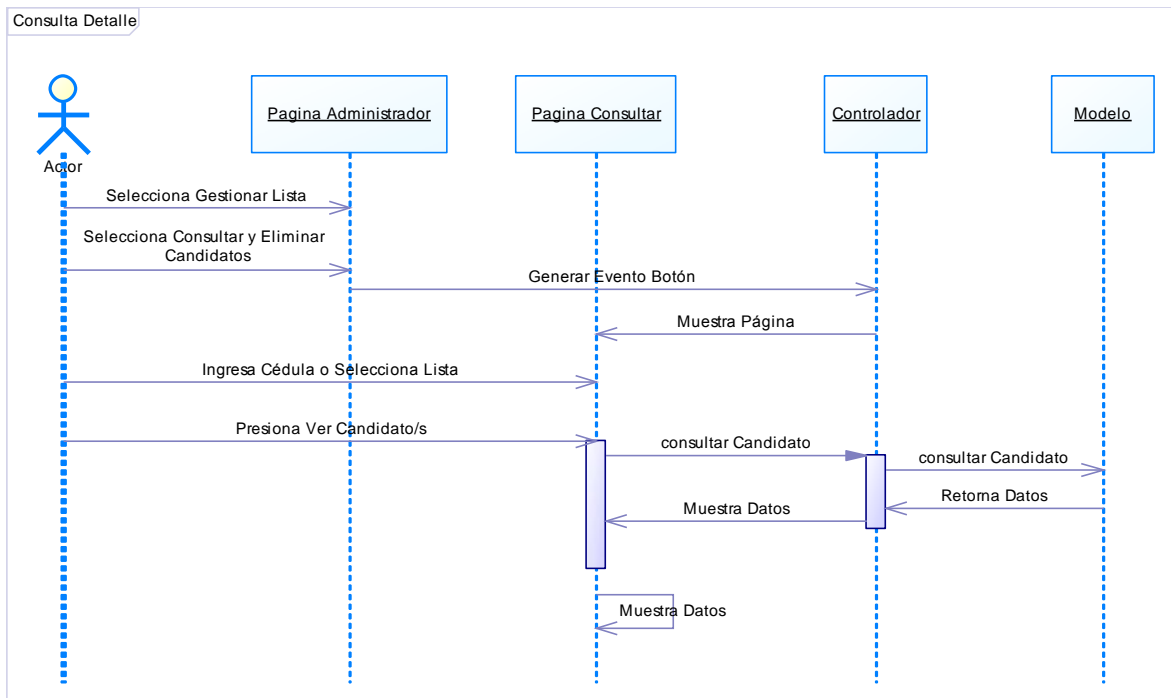


Diagrama 32 - Secuencia Consulta Detalle Candidatos

Elaborado por: Alex Calvopiña, Stephanie García; 2016

F4. Votar

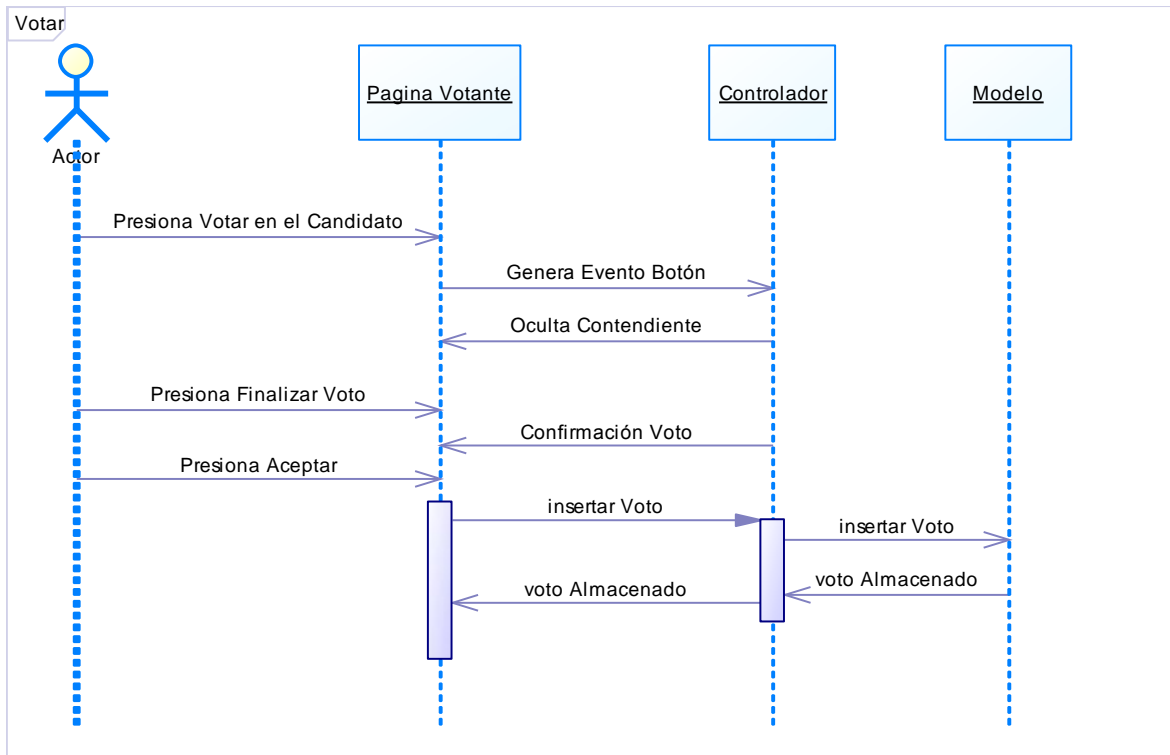


Diagrama 33 - Secuencia Votar

Elaborado por: Alex Calvopiña, Stephanie García; 2016

F5. Almacenar Detalles del Votante

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

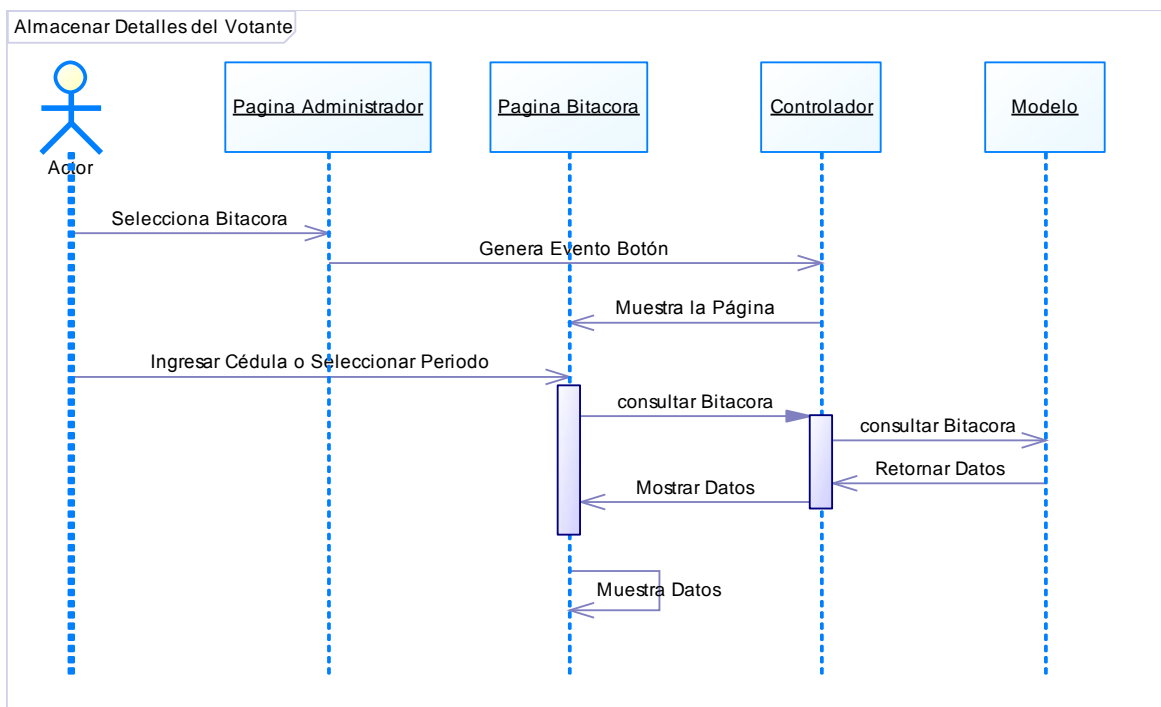


Diagrama 34 - Secuencia Almacenar Detalles del Votante

Elaborado por: Alex Calvopiña, Stephanie García; 2016

3.4.2.3. Diagrama de Actividades

Los Diagramas de Actividades nos permiten representar el flujo de actividades que son realizadas dentro de un proceso, a continuación, se detalla un diagrama de actividades del proceso de votar cada carril representa los actores que participan durante todo el proceso.

F4. Votar

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

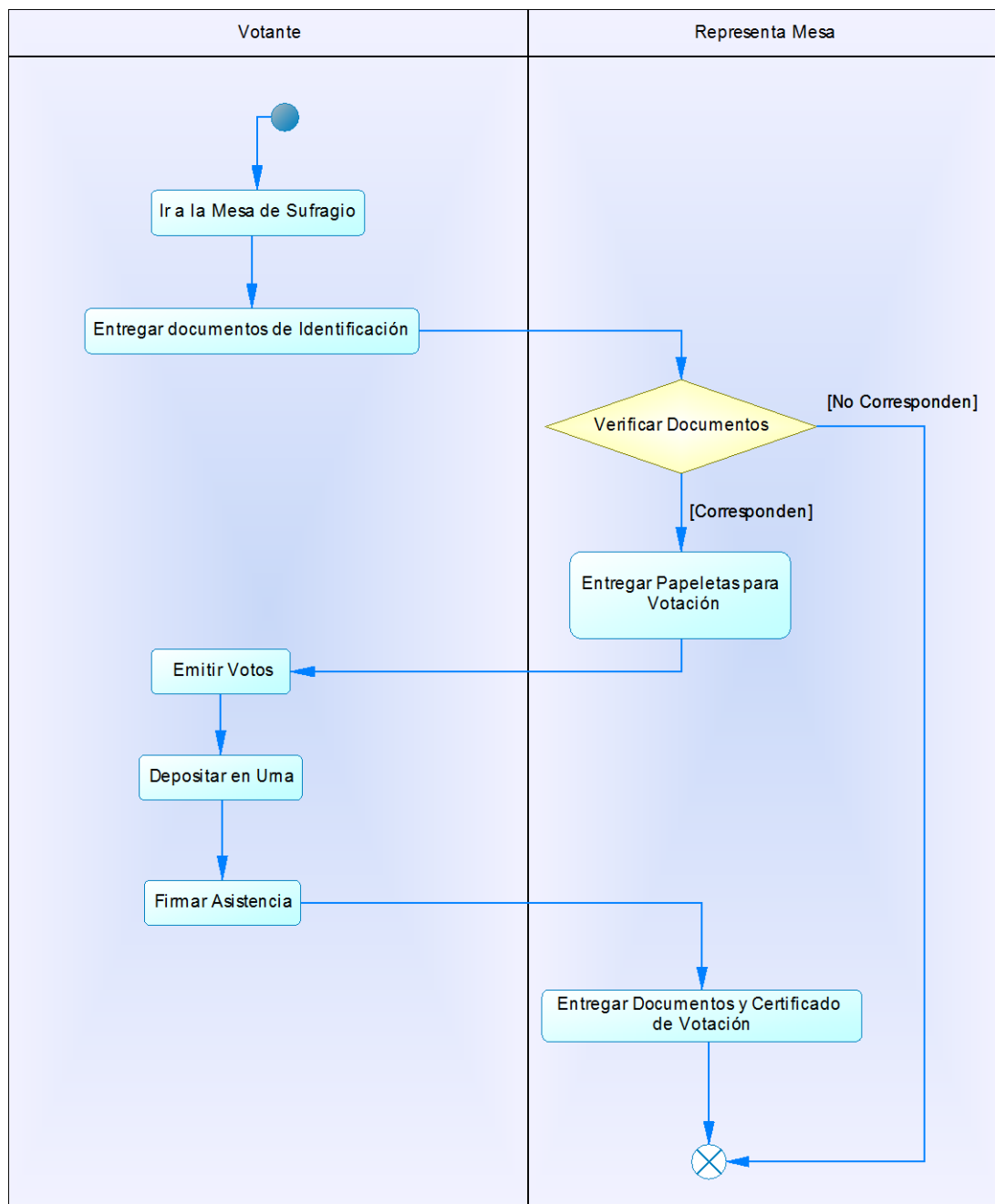


Diagrama 35 - Actividades Votar

Elaborado por: Alex Calvopiña, Stephanie García; 2016

3.4.2.4. Prototipos

F3. Gestionar Candidatos

F3.1 Ingresar Candidatos

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

Ingresar datos de un nuevo Candidato:

Cedula:

Nombre:

Apellido:

Cargo:

Correo:

Facultad:

Nivel:

Imagen: profile_placeholder.gif

Ilustración 29 - Ingresar Candidatos

F3.2 Modificar Candidatos

Modificar Candidatos

ID	CEDULA	NOMBRE	APELLIDO	CARGO	CORREO	FACULTAD	NIVEL	
38	112345678	Alex	Mena	CONSEJERO	Calex@gmail.com	Psicología	7	Edición
39	1223456789	Andrea	Polanco	CONSEJERO ACADÉMICO PRIMER SUPLENTE	andrea@hotmail.com	Hoteles y Turismo	6	Editar
49	9876543211	Andres	Bastidas	TERCER VOCALES SUPLENTE	andres@hotmail.com	Ciencias Humanas	3	Editar
36	1234567890	Daniel	Serrano	PRESIDENTE	daniel@hotmail.com	Administración	8	Editar
45	1234567889	David	Flores	PRIMER VOCALES	david@hotmail.com	Arquitectura	6	Editar

Ilustración 30 - Modificar Candidatos

F3.3 Eliminar Candidatos

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

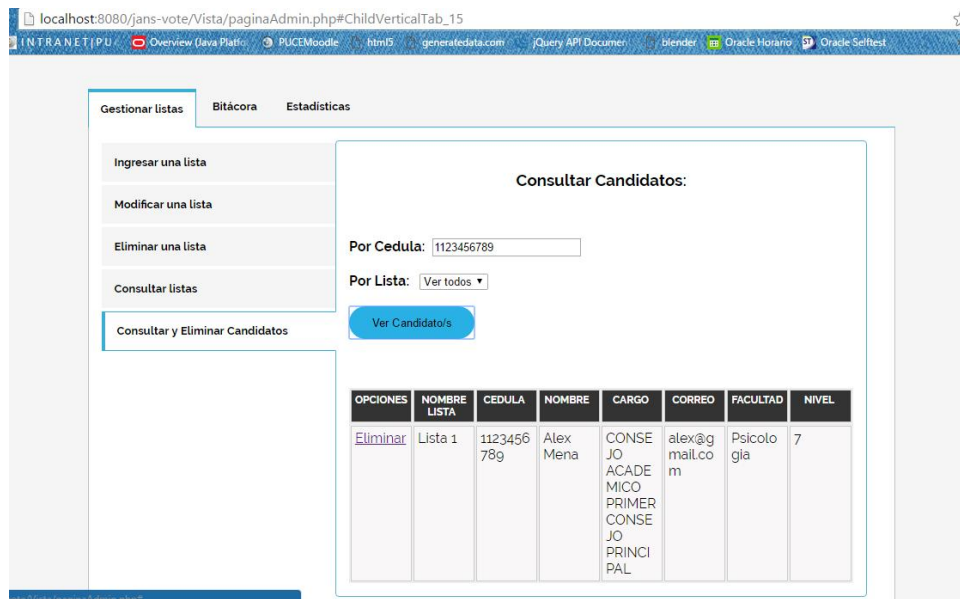


Ilustración 31 - Eliminar Candidatos

F3.4 Consultar Candidatos

F3.4.1 Consultar Candidatos General

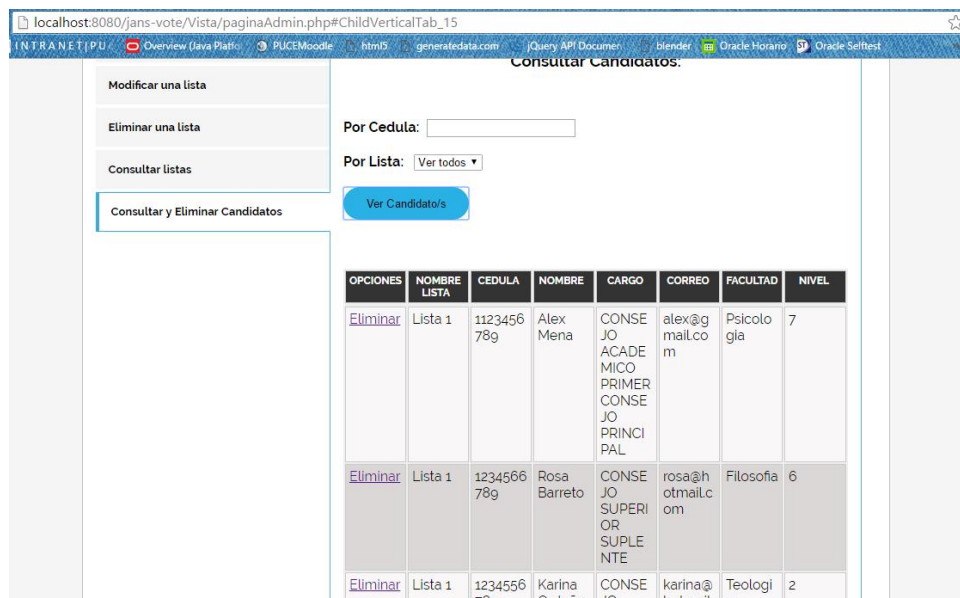


Ilustración 32 - Consultar Candidatos General

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

F3.4.2 Consultar Candidatos Detalle

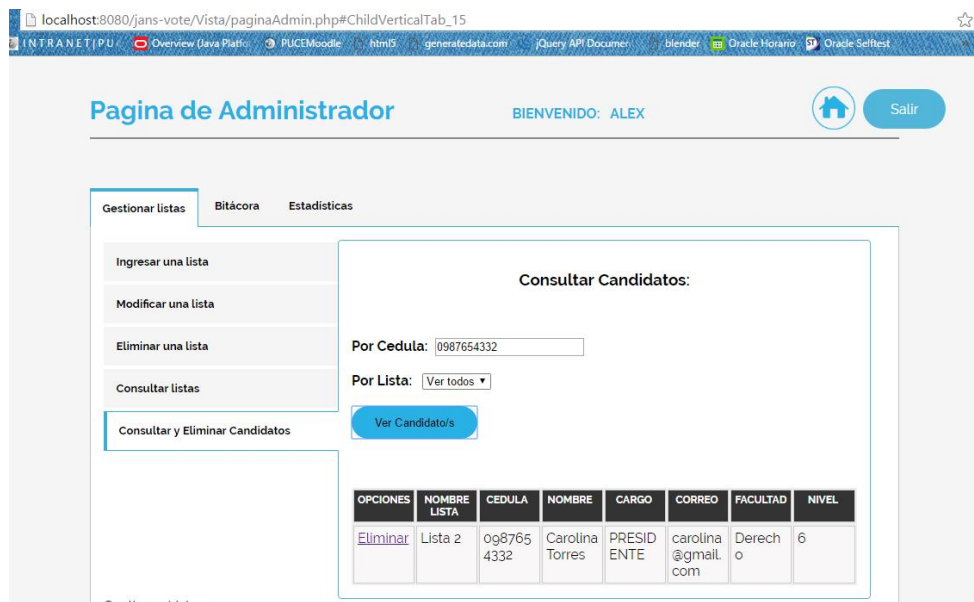


Ilustración 33 - Consultar Candidatos Detalle

F4. Votar

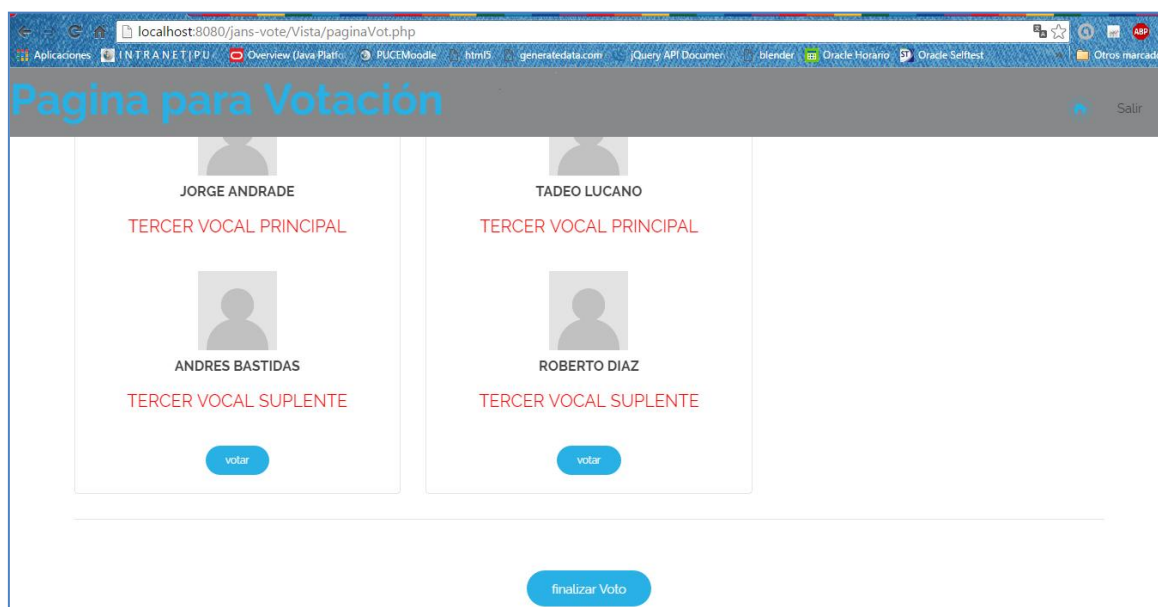


Ilustración 34 - Votar

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

F5. Almacenar Detalles del Votante



Ilustración 35 - Almacenar Detalles del Votante

3.4.3. Desarrollo

El desarrollo de la aplicación web, se realiza en base a lo descrito en el diseño de la segunda iteración, dentro de la cual se planificó realizar mejoras de las funcionalidades F0, F1, F2 de acuerdo a las observaciones obtenidas en las pruebas realizadas en la primera iteración. A su vez se desarrollarán las siguientes funcionalidades:

F3: Gestión de Candidatos: Ingresar, Modificar, Eliminar y Consultar Candidatos.

F4: Votar

F5: Almacenar Detalles del Votante.

El código fuente completo se podrá visualizar en el CD.

3.4.4. Pruebas

El realizar pruebas con un parte de los usuarios finales nos permiten obtener información sobre el funcionamiento del sistema y la calidad del mismo, a su vez nos permite conocer el punto de vista del usuario sobre lo desarrollado, en esta ocasión se realizarán pruebas de las funcionalidades desarrolladas en la segunda iteración y aquellas sugerencias acogidas en las pruebas de la primera iteración.

3.4.4.1. Informe de Pruebas

Las pruebas realizadas al finalizar el desarrollo de la segunda iteración nos permiten ampliar nuestra visión sobre las mejoras que se pueden desarrollar en la aplicación o errores que se pueden encontrar en su funcionamiento.

Los estudiantes que pertenecen a la FEUCE son aquellos que se encargaran de usar el sistema y probar sus funcionalidades, debido a que son considerados parte de los usuarios finales del mismo.

Las funcionalidades que se probaron en la segunda iteración son:

- F3: Gestión de Candidatos: Ingreso, modificación, eliminación, consulta de una o varios candidatos.
- F4: Votar: Muestra los candidatos de cada lista y agrupados por su cargo, permitiendo al usuario sufragar por su candidato preferido.
- F5: Bitácora: Muestra detalles relevantes acerca del proceso de sufragio, pero no muestra datos relacionados acerca de por quién o a quien dispuso su voto.

Resultados:

Funcionalidad	Sugerencias	Aceptación
F3: Gestión Candidatos		
F4: Votar	<ul style="list-style-type: none">• Efectuar voto blanco y	<ul style="list-style-type: none">• No se realizará por cada

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

	nulo por un solo candidato.	candidato si no en se mantendrá el votar nulo o blanco en general.
F2: Almacenar datos del Votante.		

Tabla 6 - Informe de Pruebas Iteración 2

Elaborado por: Alex Calvopiña, Stephanie García; 2016

Las sugerencias realizadas por los estudiantes que forman parte de la FEUCE sobre la votación no fueron acogidas como se puede observar en la tabla de resultados.

3.5. Tercera Iteración

Para el desarrollo de la tercera iteración se considerará las observaciones obtenidas en las iteraciones anteriores y continuamos con la implementación de la Página de Administrador en la cual se añade la Página de Estadística donde se muestran los resultados de las votaciones de acuerdo al período que se requiera consultar, se puede acceder al apartado de gráficos donde se muestran pasteles con los porcentajes obtenidos de las votaciones.

A su vez se realiza el desarrollo para la emisión de resultados, los cuales se presentan en un documento donde los usuarios pueden observar tablas con los resultados y gráficos con los porcentajes.

3.5.1. Requerimientos Funcionales

3.5.1.1. Casos de Uso

Después de realizar el desarrollo de los requerimientos funcionales de dos iteraciones, a continuación, se despliega los diagramas de caso de uso a detalle de las funcionalidades previstas para la tercera iteración las cuales son: F6, F7 y F8. Los diagramas de Caso de Uso a Detalle nos permiten visualizar la fluidez al interactuar el usuario con el sistema.

F6: Notificar Finalización Voto y Emitir Papeleta

Descripción: Sistema notifica la correcta finalización del proceso y emite una papeleta de votación mediante el uso de un correo electrónico al votante.

Actores: Usuario, Votante

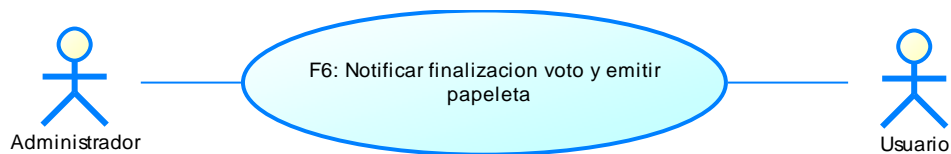


Diagrama 36 - Caso de Uso Notificar finalización del voto y emitir papeleta

Elaborado por: Alex Calvopiña, Stephanie García; 2016

Flujo Principal:

1. Actor presiona "Finalizar Voto".
2. Sistema muestra mensaje de confirmación.
3. Actor presiona Aceptar.
4. Sistema recibe el evento del botón. (E1)
5. Sistema consulta los datos del votante en la Base de Datos.
6. Sistema genera la papeleta de votación con los datos.
7. Sistema emite el correo electrónico al votante.
8. Actor recibe una notificación y la papeleta de votación al correo. (E2)
9. Actor puede observar o descargar la papeleta de votación.

Excepciones:

E1: Problemas al generar el evento del botón "Votar".

Solución:

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

- Comprobar conexión a internet.
- Comprobar el correcto ingreso de los campos.
- Contactar con el desarrollador.

E2: Correo electrónico no recibido.

Solución:

- Contactar con el Administrador.
- Verificar el correo, este correcto.

F7: Contar Votos

Descripción: Cada vez que se realice una votación se irá incrementando los votos en la tabla de Estadística.

Actores: Administrador, usuario.

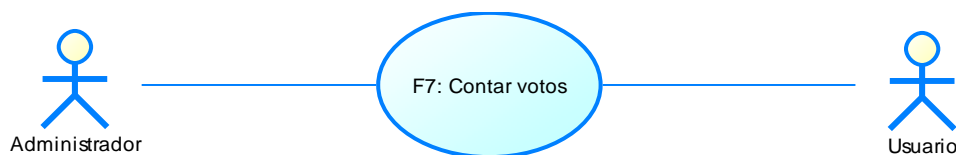


Diagrama 37 - Caso de Uso Contar Votos

Elaborado por: Alex Calvopiña, Stephanie García; 2016

Flujo Principal:

1. Actor presiona "Finalizar Voto"
2. Sistema muestra mensaje de confirmación.
3. Actor presiona Aceptar.
4. Sistema agrega un voto en la tabla de Estadística. (E1)

Excepciones:

E1: Sistema no agregó voto en la tabla. Solución: Contactar con el Administrador

F8: Emitir Resultados

Descripción: Una vez finalizado el proceso de sufragio, se emitirán los resultados sobre los votos que ha obtenido cada candidato.

Actores: Administrador, usuario, votante.

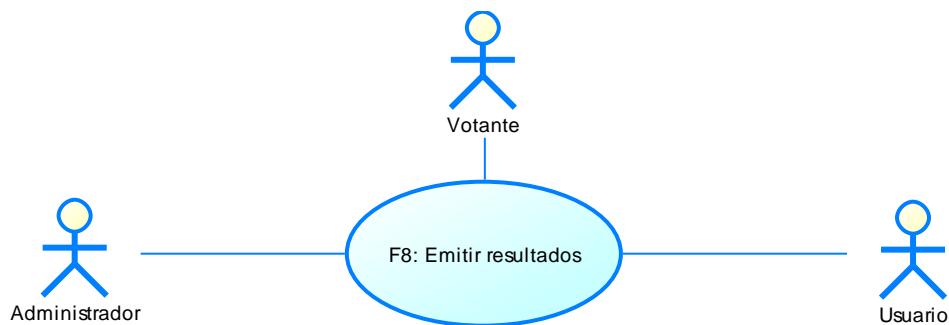


Diagrama 38 - Caso de Uso Emitir resultados y notifica los ganadores

Elaborado por: Alex Calvopiña, Stephanie García; 2016

Flujo Principal:

1. Actor selecciona Emitir Resultados.
2. Sistema obtiene los datos del conteo. (E1)
3. Sistema asocia los resultados mediante estadísticas.
4. Sistema cruza los resultados y presenta los cargos a ocupar por parte de los candidatos y los notifica.
5. Sistema genera un documento con los resultados obtenidos de las votaciones.

Excepciones:

E1: Sistema no pudo realizar el conteo de los votos. Solución: Contactar con el Administrador.

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

3.5.2. Diseño

3.5.2.1. Diagrama de Clase

El Diagrama 39 es un conjunto de clases e interacciones que representa como se maneja internamente el sistema a nivel de relaciones entre sus clases. Además de la definición de sus atributos y funciones propias de cada clase.



Diagrama 39 - Diagrama de Clases

Elaborado por: Alex Calvopiña, Stephanie García; 2016

3.5.2.2. Diagramas de Secuencia

En base al desarrollo de los diagramas de Caso de Uso a continuación se representan los diagramas de Secuencia de las funcionalidades planificadas en el desarrollo de la tercera

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

iteración. Estos diagramas nos permiten visualizar el orden en que se realizan las interacciones del usuario con el sistema.

F6. Notificar Finalización Voto y Emitir Papeleta

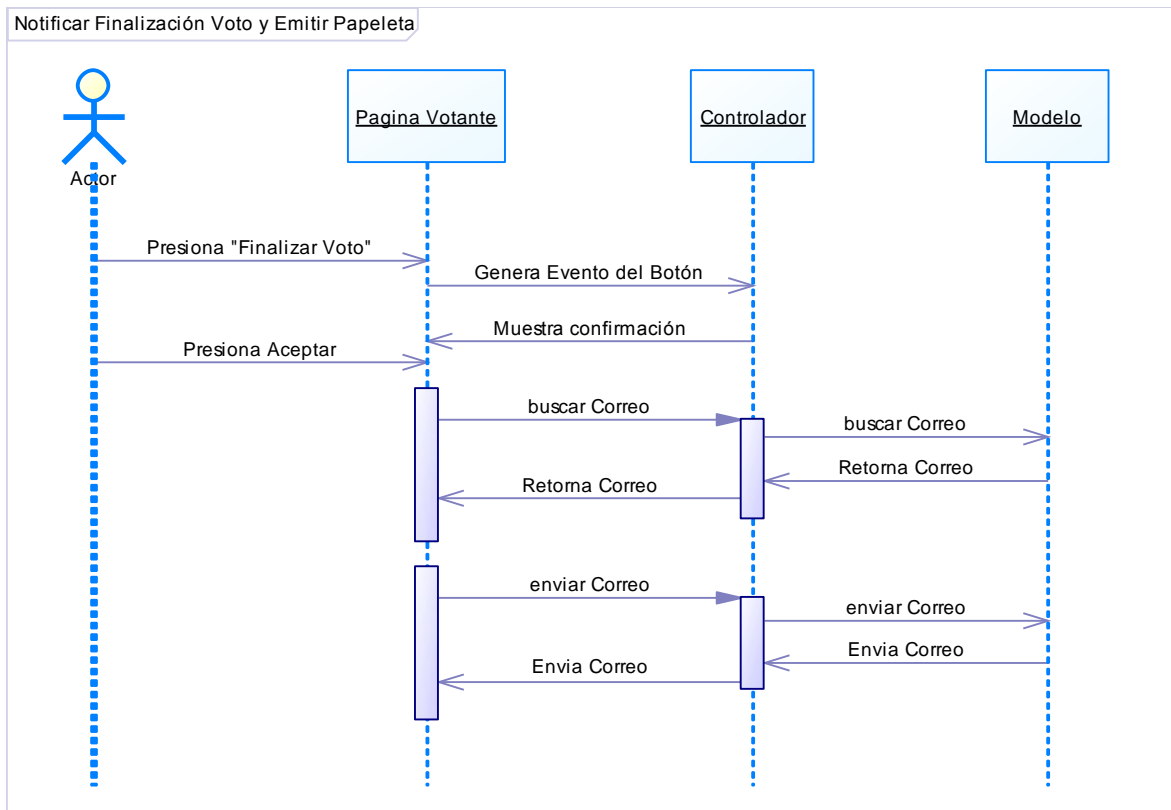


Diagrama 40 - Secuencia Notificar Finalización Voto y Emitir Papeleta

Elaborado por: Alex Calvopiña, Stephanie García; 2016

F7. Contar Votos

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

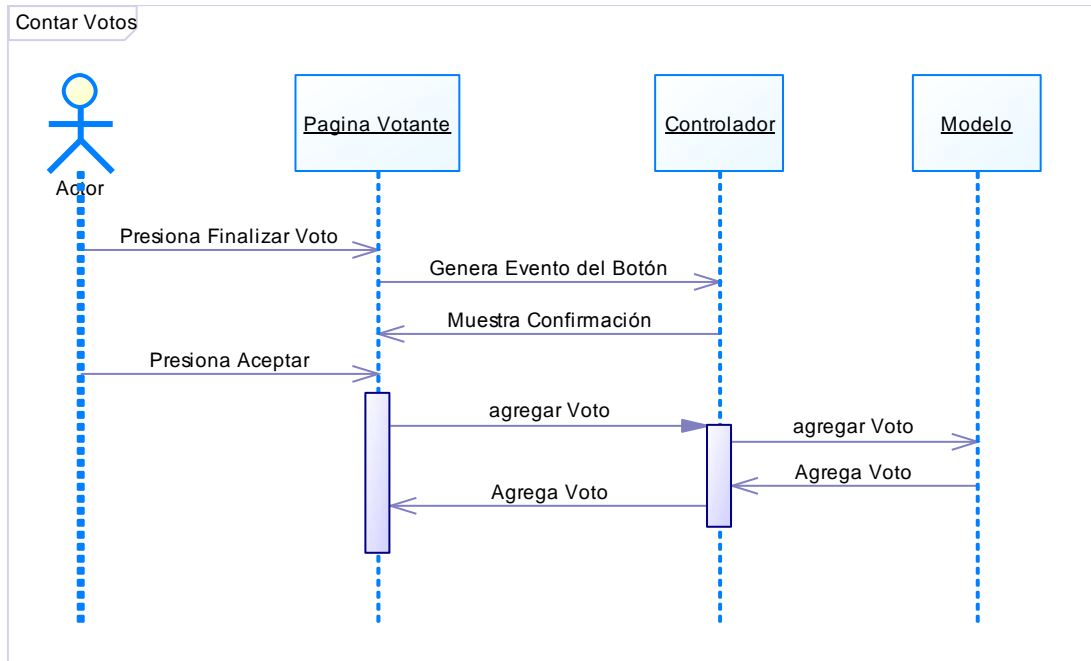


Diagrama 41 - Secuencia Contar Votos

Elaborado por: Alex Calvopiña, Stephanie García; 2016

F8. Emitir Resultados

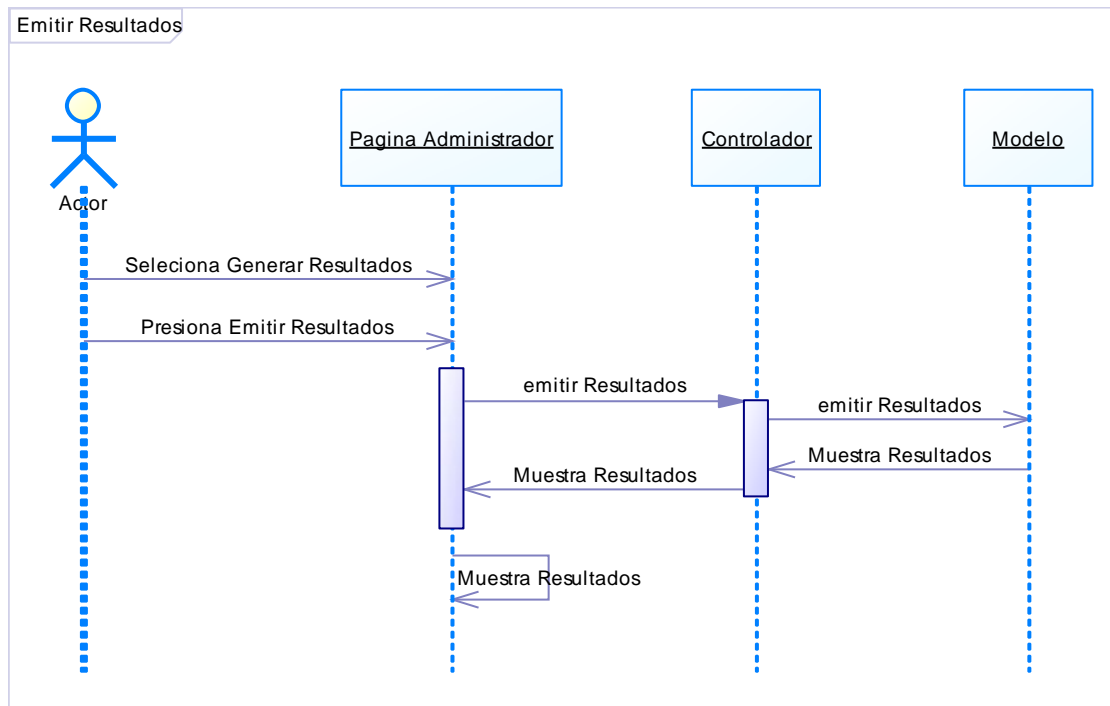


Diagrama 42 - Secuencia Emitir Resultados

Elaborado por: Alex Calvopiña, Stephanie García; 2016

3.5.2.3. Diagrama de Actividades

A continuación, se representa el flujo de actividades del proceso "Contar Votos", este diagrama nos permite conocer el orden en que son realizadas las actividades y tareas dentro del proceso, además permite definir quién es el responsable o encargado de cada actividad.

F7. Contar Votos

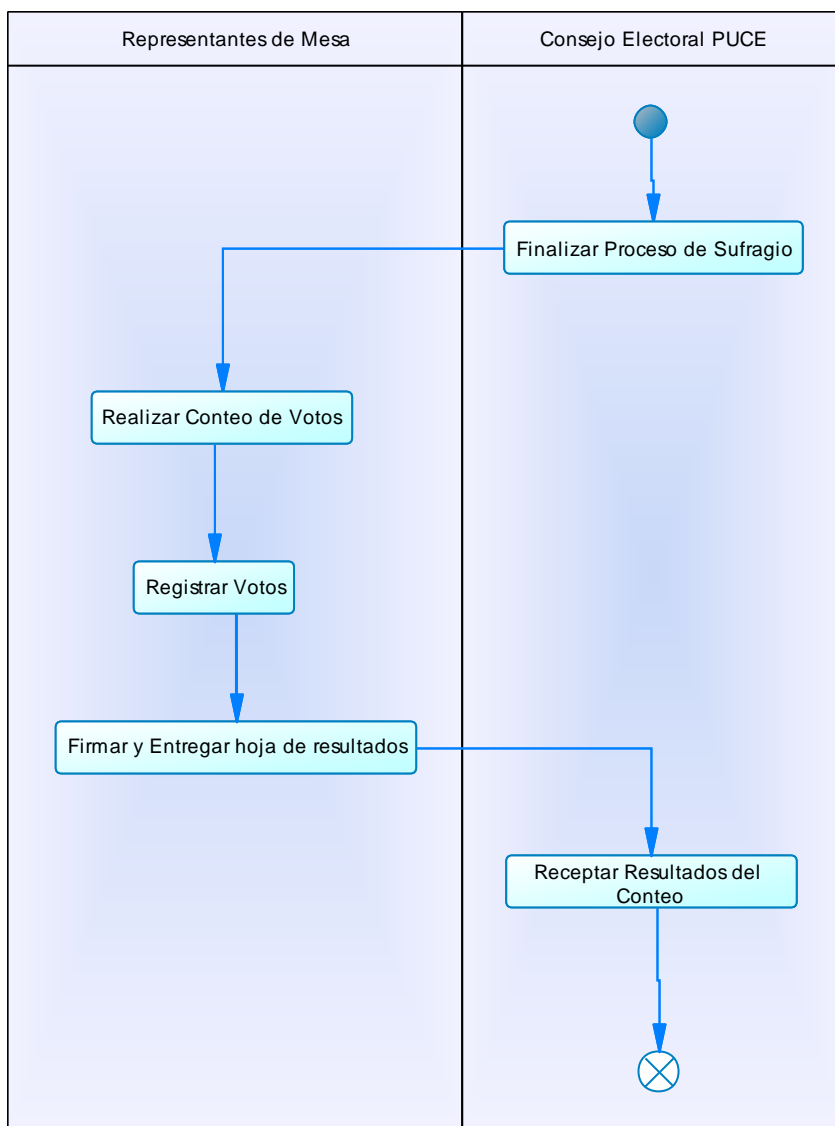


Diagrama 43 - Actividades Contar Votos

Elaborado por: Alex Calvopiña, Stephanie García; 2016

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

3.5.2.4. Diagrama Físico

El diagrama Físico nos permite visualizar como esta estructura la base de datos empleada por la aplicación web, en el diagrama podemos observar las entidades o tablas por las cuales están conformadas la base, los atributos de cada tabla y como están relacionadas las entidades entre sí.

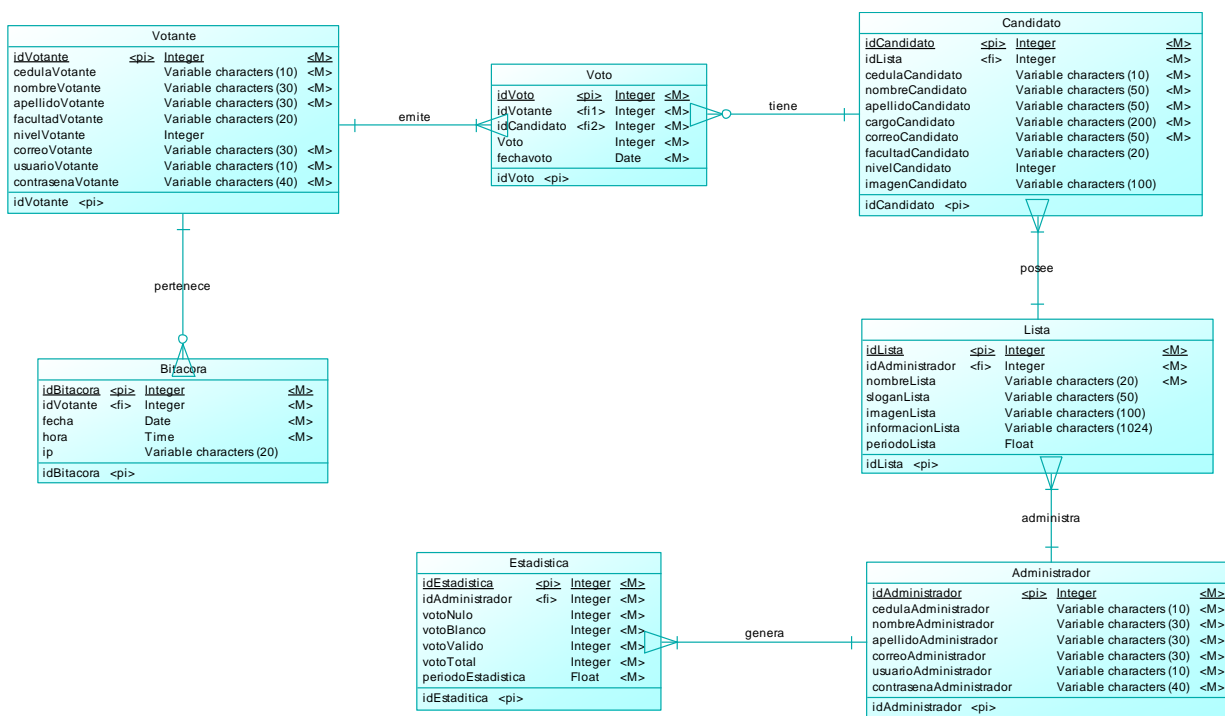


Diagrama 44 - Diagrama Físico

Elaborado por: Alex Calvopiña, Stephanie García; 2016

3.5.2.5. Prototipos

F6. Notifica Finalización Voto y Emite Papeleta

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

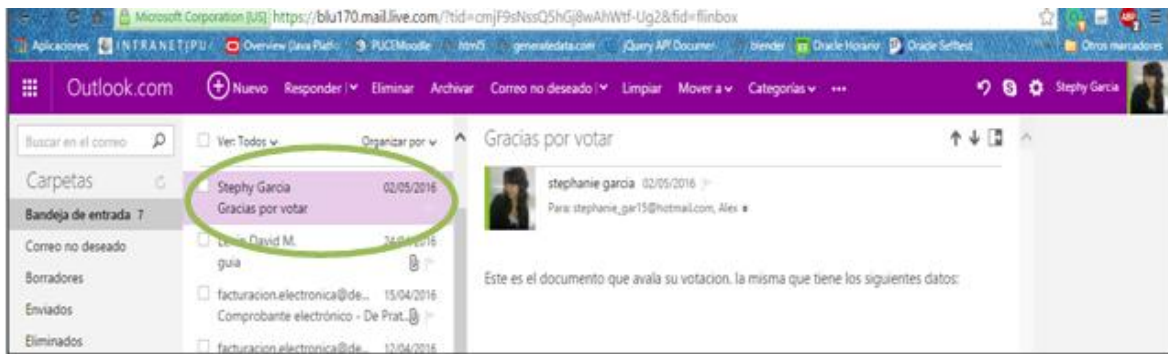


Ilustración 36 - Notifica Finalización Voto y Emite Papeleta

F7. Contar Votos

Totales:

Periodo: 2016

Buscar

ID	VOTOS NULO	VOTOS BLANCO	VOTOS VALIDOS	TOTAL VOTOS
3	0	0	4	4

Detallados:

Periodo: 2016

Buscar

LISTA	NOMBRE	CARGO	TOTAL VOTOS
Lista 1	Pedro Ruiz	PRIMER VOCAL PRINCIPAL	4
Lista 1	David Flores	PRIMER VOCAL SUPLENTE	4
Lista 2	Lenin Rodríguez	CONSEJO ACADEMICO SEGUNDO CONSEJO SUPLENTE	3

Ilustración 37 - Contar Votos Tablas

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

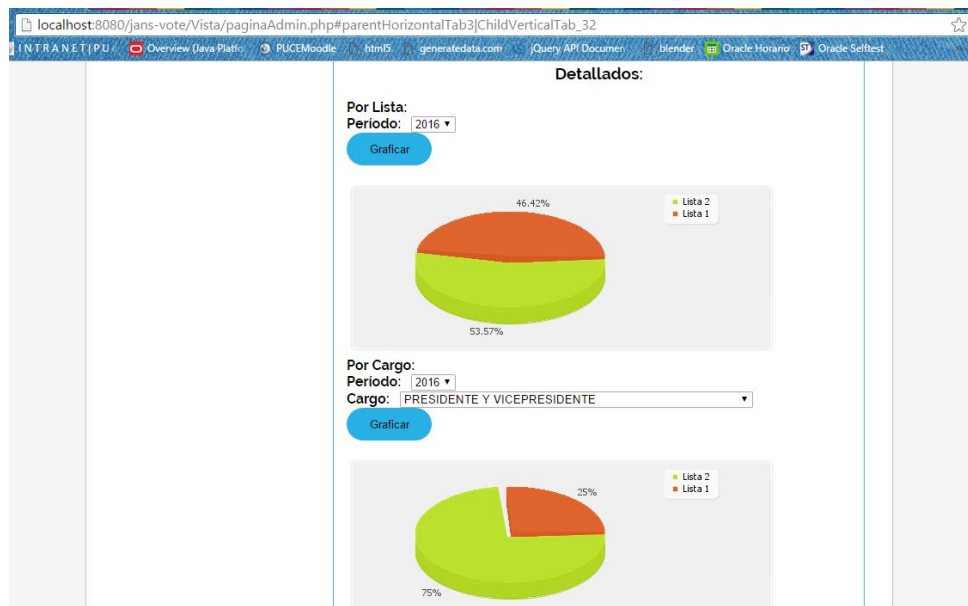


Ilustración 38 - Contar Votos Gráficos

F8. Emitir Resultados

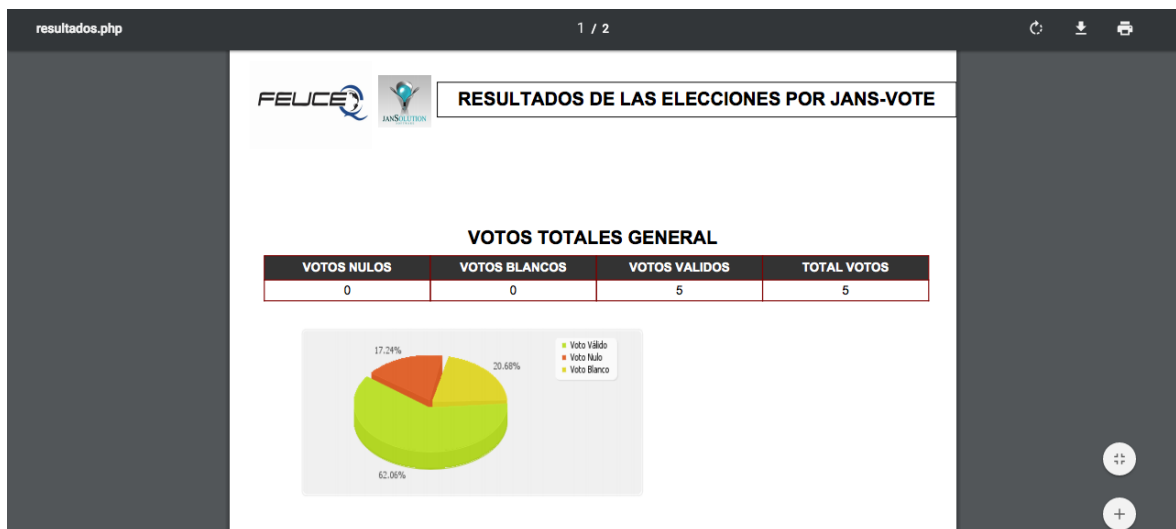


Ilustración 39 - Emitir Resultados 1

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

resultados.php 1 / 2

VOTOS TOTALES A DETALLE (POR CARGO)

Resultados para: CONSEJO ACADEMICO PRIMER CONSEJO PRINCIPAL

LISTA	NOMBRE	APELLIDO	TOTAL VOTOS
lista 1	Ikahsdkjh	kjkhjk	4
lista 3	Valeria	Rosero	1

Resultados para: CONSEJO ACADEMICO PRIMER CONSEJO SUPLENTE

LISTA	NOMBRE	APELLIDO	TOTAL VOTOS
lista 1	Leslie	Males	4
lista 3	Claudine	Cadena	1

Resultados para: CONSEJO SUPERIOR PRINCIPAL

LISTA	NOMBRE	APELLIDO	TOTAL VOTOS
lista 1	Valeria	Astudillo	5

Resultados para: CONSEJO SUPERIOR SUPLENTE

LISTA	NOMBRE	APELLIDO	TOTAL VOTOS
-------	--------	----------	-------------

Pag 1/2

Ilustración 40 - Emitir Resultados 2

3.5.3. Desarrollo

El desarrollo de las funcionalidades de la tercera iteración, se realizarán de acuerdo a lo establecido en el diseño, además se planificó realizar mejoras de las funcionalidades F3, F4, F5 pertenecientes a la segunda iteración, de acuerdo a las observaciones obtenidas en las pruebas realizadas. Las funcionalidades de esta iteración son las siguientes:

F6: Notificar Voto y Emitir Papeleta.

F7: Contar Votos

F8: Emitir Resultados.

El código completo se podrá visualizar en el CD.

3.5.4. Pruebas

Ejecutar el plan pruebas con los usuarios finales del sistema nos permiten tener información objetiva e independiente de la calidad del sistema y sus funcionalidades, en esta ocasión se realizarán pruebas de las funcionalidades desarrolladas en la tercera iteración.

3.5.4.1. Informe de Pruebas

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

Las pruebas realizadas al finalizar el desarrollo de la tercera iteración nos permiten ampliar nuestra visión sobre las mejoras que se pueden implementar en la aplicación o errores que se pueden encontrar en su funcionamiento, y así obtener un mejor desempeño del sistema.

Los estudiantes que pertenecen a la FEUCE son aquellos que se encargaran de usar el sistema y probar sus funcionalidades.

Las funcionalidades que se probaron en la tercera iteración son:

- F6: Notificar Finalización Voto y Emitir Papeleta: Envía un correo electrónico notificando que ha finalizado el proceso de sufragio.
- F7: Contar Votos: Realizar un incremento de los votos de cada candidato.
- F8: Emitir Resultados: Muestra un pdf con los datos de las votaciones, además muestra gráficos estadísticos.

Resultados:

Funcionalidad	Sugerencias	Aceptación
F6: Notificar Finalización Voto y Emitir Papeleta.	<ul style="list-style-type: none">• Enviar la notificación al correo de la PUCE y al personal	<ul style="list-style-type: none">• No se agregará otro correo debido a que en la base se tiene solo el institucional.
F7: Contar Votos		
F8: Emitir Resultados.		

Tabla 7 - Informe Pruebas Iteración 3

Elaborado por: Alex Calvopiña, Stephanie García; 2016

3.6. Producción o Despliegue

Los diagramas de despliegue permiten representar como se compone un sistema incluyendo su software y hardware, además muestra las relaciones que existen entre los componentes del sistema.

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

El diagrama 45 describe la configuración del sistema donde se muestran los nodos por los que está compuesto el sistema como son: Cliente Web, Servidor Web y Servidor DB, además de las relaciones que existen entre estos nodos.

Dentro de cada nodo se puede evidenciar sus componentes, en el nodo del Servidor Web podemos observar que la aplicación emplea el patrón de diseño MVC (Modelo - Vista - Controlador) y como están relacionados dichos componentes.

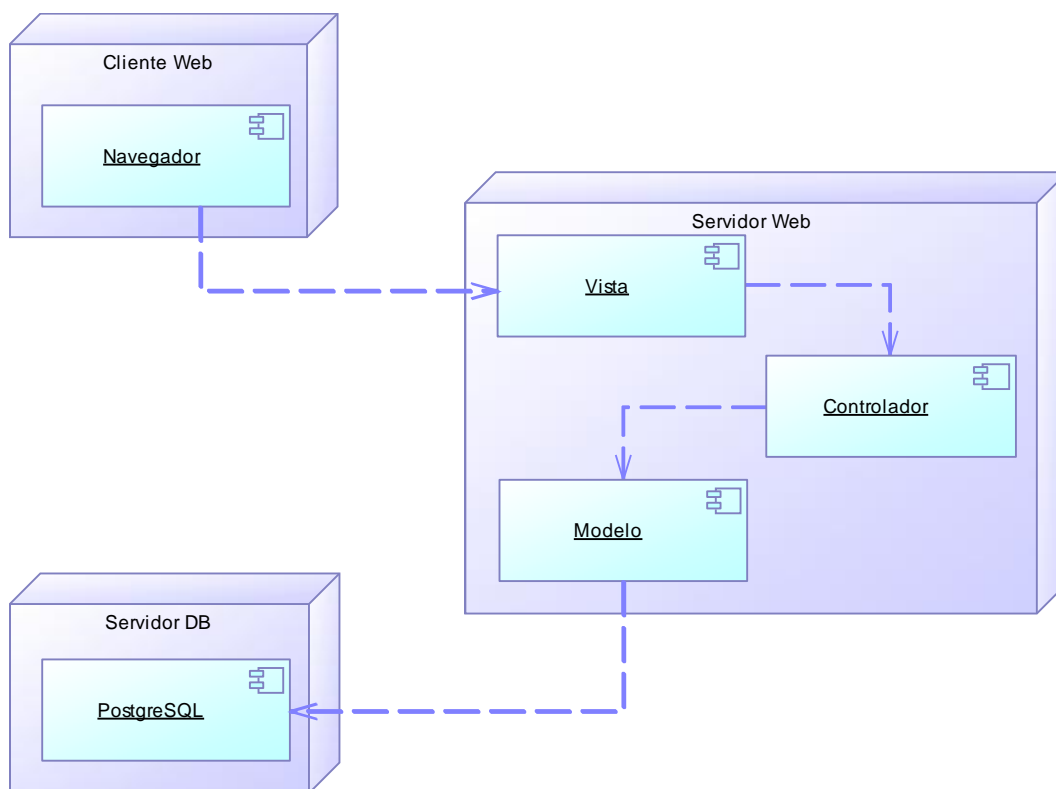


Diagrama 45 - Despliegue

Elaborado por: Alex Calvopiña, Stephanie García; 2016

3.7. Mantenimiento

Al haber concluido la etapa de codificación del proyecto y resuelto las necesidades del cliente, es necesario poner a la orden ciertos aspectos claves que serán de mucha

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

utilidad a la hora de realizar un mantenimiento al sistema, o desarrollar una nueva versión del mismo.

Al ser una aplicación orientada a web(cliente-servidor), con eventos e interacciones y con almacenamiento de información, los conocimientos técnicos necesarios para modificar, corregir o agregar código a la aplicación serían:

- **Lenguajes de programación:**

- Lado del cliente: Javascript (Jquery), HTML, CSS.
- Lado del servidor: PHP.

- **Editores de texto:**

Que soporten los lenguajes de programación explicados anteriormente con el fin de facilitar la escritura y lectura de código. Existe una infinidad de editores de texto, entre los cuales podemos sugerir los siguientes:

- **Windows:** Notepad++.
- **Mac OS X:** Atom.
- **Linux:** Vim.

- **Sistemas Gestores de Bases de Datos:**

Para administrar la Base de Datos del aplicativo, se requiere un SGBD, de los comúnmente utilizados, se recomienda PostgreSQL.

Para realizar sentencias (inserción, modificación, eliminación, etc.) dentro de un SGBD, se requiere tener conocimiento del lenguaje SQL.

Arquitectura

El aplicativo está construido mediante el patrón de arquitectura de Software MVC, de esta manera, al requerir modificar, o agregar código en algún documento del sistema, la

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

repartición de los documentos existentes en el proyecto se encuentra de la siguiente forma:

- **Controlador:** Eventos, utilización de datos.
- **Modelo:** Conexiones a la Base de Datos.
- **Vista:** Páginas presentadas al cliente.

En la siguiente imagen se pueden observar que el aplicativo está estructurado de acuerdo a la arquitectura MVC.

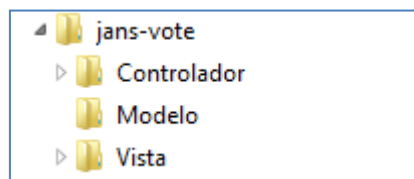


Ilustración 41 - Arquitectura MVC

Controlador: Dentro de la carpeta del controlador, se tiene los siguientes archivos.

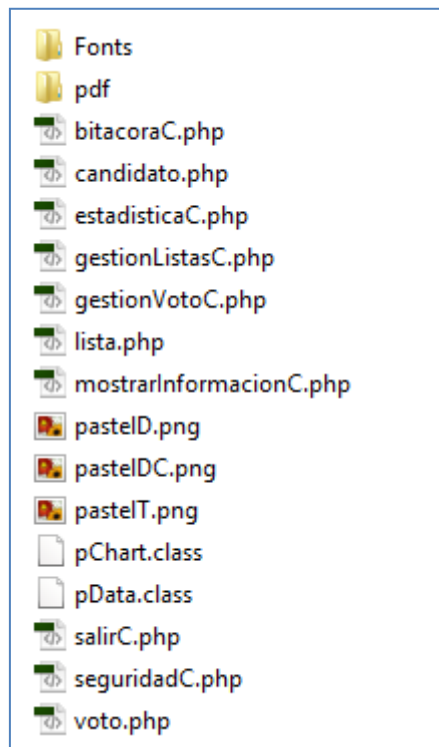


Ilustración 42 - Carpeta Controlador

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

Modelo: En el modelo tenemos los archivos donde se realiza las conexiones e interacciones con la base de datos.

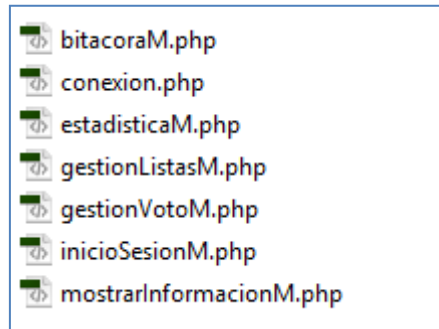


Ilustración 43 - Carpeta Modelo

Vista: En la parte de Vista se encuentra todo lo relacionado a las páginas con las que el cliente interactúa.

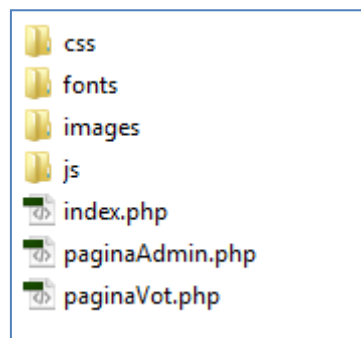


Ilustración 44 - Carpeta Vista

Inicio de sesión enlazado a datos de la PUCE

El aplicativo web, tiene como una de sus funcionalidades, dentro de la página principal del mismo, permitir el inicio de sesión de los estudiantes y administrativos con sus usuarios y contraseñas que han sido suministrados por la universidad, los mismos que son utilizados para otros aplicativos, como la Intranet.

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

El propósito de esta disertación es el de presentar la propuesta del voto electrónico para futuras elecciones dentro de la PUCE, por lo tanto, se requiere presentar los aspectos a tener en cuenta para llegar a hacer efectivo la utilización de este aplicativo web enlazado a los datos de los estudiantes dentro de la PUCE.

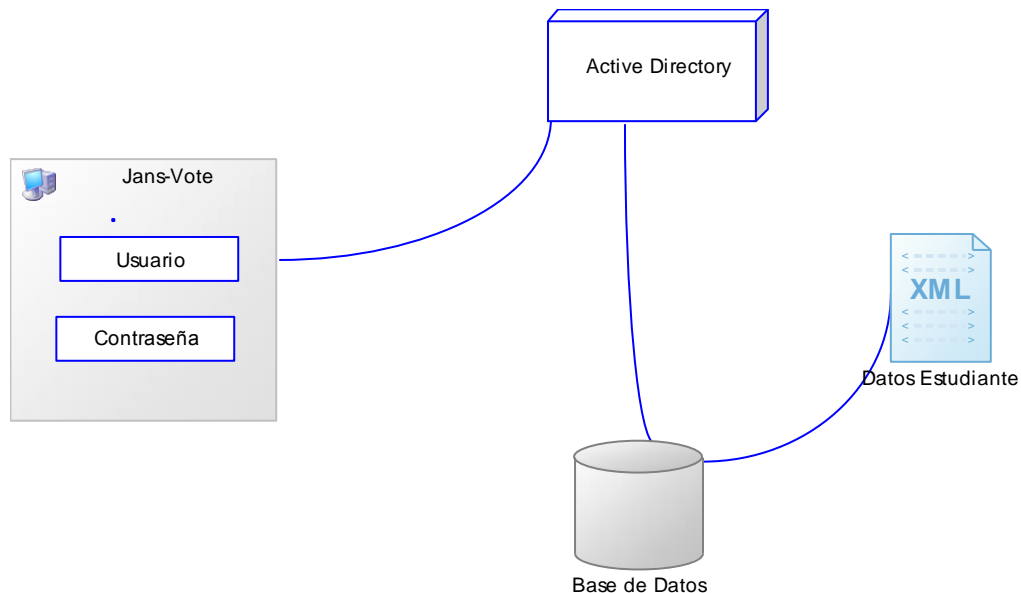


Diagrama 46 - Diagrama Consumo Web Services

Elaborado por: Alex Calvopiña, Stephanie García; 2016

Web Services

Son una tecnología utilizada actualmente que permiten el intercambio de información entre aplicaciones indistintamente de su lenguaje de programación o plataforma para la que han sido diseñadas.

Técnicamente, los servicios web utilizan como lenguaje de codificación y decodificación de la data, XML (Extensible Markup Language), mientras que para transportar estos, utiliza SOAP (Simple Object Access Protocol).

Para poder realizar el enlace pertinente y acceder a los datos de los estudiantes, se requiere consumir el Web Service que tiene la Institución, desde el aplicativo web, el

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

mismo que al ser desarrollado en PHP, permite realizar este consumo utilizando el siguiente código:

Opción 1: CLASE SOAPCLIENT

```
$webservice="http://..... "; //url del dominio de la institución.
```



```
//Parámetros que se envía desde la página del aplicativo mediante el método post.
```

```
$parametros=array ();
```

```
$parametros['usuario']=$_POST['usuario'];
```

```
$parametros['contrasena']=$_POST['contrasena'];
```



```
// Invocación al web service
```

```
$wb = new soapclient ($webservice, $parametros);
```



```
// Llamamos al método dentro del web service que nos devolverá si el usuario y la
```

```
contraseña ingresados por el usuario efectivamente pertenecen a la institución.
```

```
$result = $wb->getmetodo($parametros);
```

Se obtiene un XML en la variable \$result, con la que se puede trabajar para poder utilizar o no el aplicativo Web, dependiendo de si el usuario es estudiante o no de la Institución.

Opción 2: LIBRERÍA CURL

```
//Recuperamos los datos mediante POST
```

```
$usuario=POST['usuario'];
```

```
$Contrasena=POST['contrasena'];
```



```
//Enlazamos una variable al link del WebService enviando los parámetros que este
```

```
requiera
```

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

```
$WebService="http://www.ServicioWeb/prueba.asmx/GetMetodo?usuario=$usuario?contrasena=$contrasena";
```

```
//Utilizamos e instanciamos la librería CURL de PHP.
```

```
$ch = curl_init ();
```

```
curl_setopt ($ch, CURLOPT_URL, $WebService);
```

```
curl_setopt ($ch, CURLOPT_RETURNTRANSFER, 1);
```

```
//Guardamos la respuesta del Web Service en una variable.
```

```
$result = curl_exec($ch);
```

```
//Cerramos CURL.
```

```
curl_close($ch);
```

SoapClient permite soportar las peticiones que han sido creadas en Curl, la diferencia reside en que al emplear SOAP con Curl el programador debe encargarse de todo el desarrollo línea por línea, en el caso de SoapClient este lo realiza todo como las llamadas a los métodos lo cual facilita realizar las conexiones.

La información que se requiere consumir del Servicio Web es: la confirmación de existencia del usuario “logueado”, el correo electrónico del mismo, junto con otro campo conocido como Mascara. Este atributo permite diferenciar de la siguiente forma a los usuarios:

- Estudiantes: 1
- Docentes: 2
- Administrativos: 4

La suma de estos números , permite resolver si por ejemplo, un Docente es a su vez un Administrativo(6), un estudiante es un Docente también(3), entre otros tipos.

A su vez, otro de los atributos pertinentes para el correcto desenvolvimiento del aplicativo web, es el conocido como Estado, el mismo que diferencia a los estudiantes que se encuentran activos en la Institución, frente a los que ya no cursan ningún nivel de estudios.

3.8. Fin de Proyecto

Al concluir las distintas fases planificadas para el desarrollo del software de acuerdo con la metodología Extreme Programming, consideramos necesario el realizar un análisis de todo el proceso de desarrollo. A continuación, tenemos una tabla que muestra el avance de la codificación de acuerdo a lo desarrollado en las iteraciones.

	Primera Iteración	Segunda Iteración	Tercera Iteración
Tamaño	1313	1404	653
Tiempo	74,5	80	53
% Avance	34%	33%	33%

Tabla 8 - Detalle Fin de Proyecto

Elaborado por: Alex Calvopiña, Stephanie García; 2016

En la primera iteración, se planificó el desarrollo de tres funcionalidades.

F0: Verificar Usuario y Contraseña.

Para realizar el desarrollo de la funcionalidad F0, tomamos en cuenta que los usuarios realizarían el inicio de sesión mediante su usuario y contraseña que son empleados en la intranet de la PUCE, esta funcionalidad fue la que tomo más tiempo para ser implementada, debido al cuidado que se tiene con la seguridad y confidencialidad de los datos ingresados por el usuario. Además del manejo de las sesiones, determinando si el usuario se mantiene en la sesión o ya ha salido.

F1: Gestión de Listas: Ingresar, Modificar, Eliminar y Consultar Listas.

Dentro del desarrollo de la Gestión de Listas se realiza la codificación de los métodos de Ingreso, Modificación, Eliminación y Consultas de las Listas. Cada uno de estos métodos están detallados en la página del administrador, la dificultad que se tuvo al desarrollar estos métodos fue el realizar el manejo de imágenes, como son la carga y

modificación de las imágenes en sus respectivos métodos. Después de desarrollar las pruebas de la primera iteración pudimos encontrar mejoras para esta funcionalidad, al realizar la carga de las tablas para las consultas.

F2: Mostrar información de las Listas.

El mostrar información de las Listas es similar al consultar listas con la diferencia que se este se muestra en la página principal donde todos los usuarios tienen acceso y su formato para ser visualizado es diferente, ya que se muestra de una manera más agradable al usuario. Para lograr ese formato se empleó CSS y JQuery.

Para el desarrollo de la segunda iteración, se desarrolla las mejoras que se obtuvieron al realizar las pruebas de la primera iteración de acuerdo a los sugerencias realizadas por los sujetos de prueba. A su vez se realiza la construcción de tres funcionalidades planificadas para esta iteración.

F3: Gestión de Candidatos.

Para realizar la construcción del Gestionar Candidatos, se lo hace en base a lo desarrollado en la Gestión de Listas, de igual manera se codifica el ingreso, modificación, eliminación y consulta de uno o varios candidatos. Estas funcionalidades van de la mano con las funcionalidades del Gestionar Lista, debido que ambas están dentro del mismo contexto.

F4: Votar.

El desarrollo de la votación es la más relevante ya que es el proceso, que principalmente se desea automatizar, en la página de votación se muestran los candidatos de cada lista y dependiendo del cargo que ocupan, esto permite al usuario votar por su candidato preferido, además se tiene la opción de emitir voto nulo o voto blanco. En el desarrollo de este módulo los conflictos se tornaron al mostrar los candidatos de acuerdo como se tiene en las papeletas para votación, lo cual se solucionó empleando CSS para el formato en que se muestran los candidatos. Y para mostrar u ocultar los candidatos se empleo Javascript.

F5: Almacenar detalles del Votante

La inserción de los datos del votante en la Base de Datos se realiza cuando el votante ha finalizado el proceso de votación. De esta manera se puede almacenar la información de la fecha y hora en que se efectuó la votación, pero cabe recalcar que dentro de esta información no se encuentra a quién emitió su voto.

El administrador es el único que puede acceder a la visualización de la bitácora, para observar como avanza el proceso de sufragio. Para obtener esta información debe acceder con su usuario y contraseña, posteriormente dirigirse a la pestaña de bitácora y ahí podrá consultar la información de todo un periodo o por votante, empleando su cédula.

Para finalizar con la construcción del sistema, se realizó una tercera iteración que de igual manera a las anteriores iteraciones se desarrollaron tres funcionalidades, en esta iteración no se acogieron las sugerencias obtenidas al aplicar las pruebas.

F6: Notificar Finalización Voto y Emitir Papeleta

Para notificar la finalización del voto, se lo realiza mediante el envío de un correo electrónico donde el votante puede verificar que su voto fue efectuado correctamente. Previo al desarrollo, se configuro al servidor para el envío de correos.

F7: Contar Votos:

El conteo de los votos es un proceso que se realiza para obtener los resultados y las estadísticas de las votaciones. Permitiendo conocer al usuario los candidatos que han obtenido la mayoría de votos, además se puede observar la cantidad de votos que obtuvo cada candidato.

F8: Emitir Resultados

La emisión de resultados se realiza mediante la elaboración de un documento, empleando tablas y gráficos para una mejor visualización. La visualización de los

resultados se lo puede hacer en el documento o directamente en la página de Administrador en la pestaña de Estadística, para los gráfico se empleó clases de Pchart.

4. CONCLUSIONES Y RECOMENDACIONES

El presente capítulo, tiene como propósito ayudar a los lectores a conocer un poco más a fondo los inconvenientes encontrados a la hora de realizar el Software, y que gracias a estas recomendaciones y conclusiones se los puede resolver y permitirán trabajar de una manera más eficiente en futuros proyectos que tengan características similares a éste.

4.1. Conclusiones

- Actualmente la FEUCE trabaja con el sistema de conteo de las votaciones tradicional, lo que conlleva mayor cantidad de tiempo y esfuerzo del personal para entregar los resultados a los estudiantes.
- Cada año la FEUCE invierte económicamente en la adquisición de papeletas para la votación, las mismas que por no ser reciclables generan grandes problemas al medio ambiente.
- Al ser un aplicativo orientado a la Web, la seguridad de la información juega un papel sumamente importante, lo que conlleva implementar mayores medidas de seguridad de los datos.
- La implementación de las votaciones electrónicas deben ser rigurosas en cómo comprobar que efectivamente es el usuario quien está votando, para descartar el fraude en las mismas. (Autenticación)
- Implementar un aplicativo web en lenguajes como PHP sin la ayuda de un Framework de desarrollo, en ocasiones presenta dificultades: de tiempo, de organización, pero da una libertad de desarrollo más amplia, ya que no necesariamente hay que adaptarse a un patrón específico.
- El patrón de arquitectura MVC, permite organizar de mejor manera el desarrollo de una aplicación donde se requiere el uso de interfaces de usuario, las mismas que gracias a este modelo, se encuentran separadas de la lógica del negocio.

- Al trabajar con una metodología de desarrollo como XP dentro de un proyecto, siempre se debe tener claros los requerimientos del cliente, a su vez, poner énfasis en la etapa de diseño, para que una vez que se empiece la codificación, no tener inconvenientes y codificar plenamente.
- Existe preocupación en las personas al emplear un sistema de votaciones electrónicas respecto a la transparencia durante el proceso electoral ya que no existe una manera física de corroborar los resultados obtenidos.
- Los procesos electorales conllevan sentimientos encontrados por parte de los candidatos y sus votantes, al ser una votación electrónica, puede producir conflictos por ataques a la integridad del sistema con fin de conseguir el puesto deseado.
- Se considera difícil y conflictivo la implantación de un sistema de votación electrónica debido a las diferentes opiniones que tienen las personas acerca del tema.
- El voto electrónico puede ser empleado como un complemento al sistema normal de votación permitiendo mejorar el desempeño del proceso.

4.2. Recomendaciones

- La codificación en parejas es una buena opción dentro de la metodología XP, pero para poder cumplir con el objetivo del equipo, es necesario definir estándares que se deben cumplir a cabalidad para no tener inconvenientes futuros. Una buena opción que ayuda a la programación en parejas es el uso de un software para manejo de versiones.
- A lo largo del desarrollo del aplicativo y es de suma importancia que el usuario esté involucrado en cada una de las iteraciones ya que esto permite a los desarrolladores poder cumplir con todas las necesidades del cliente.
- Para la implantación del sistema en un futuro es necesario revisar la fase de Mantenimiento de la presente disertación en donde detalla los pasos a seguir para enlazar el aplicativo a la Institución.

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

- Para mejorar la transparencia y confiabilidad del sistema en una futura versión se puede implementar una opción que permita imprimir el voto y depositarlo en una urna teniendo así un respaldo físico de la votación, teniendo en cuenta que esto iría en contra del pensamiento ecologista.
- Para proyectos de corto alcance es recomendable emplear una metodología de desarrollo como XP.
- Para obtener un buen nivel de calidad de un producto de software, XP propone el desarrollo en parejas.
- Para futuras versiones del aplicativo puede añadirse la opción de iniciar sesión mediante la lectura de huellas dactilares, facilitando este proceso.
- Al ser un proyecto escrito en lenguaje PHP, se debe tener en cuenta que este es un lenguaje utilizado del lado del servidor, y por este motivo existe la necesidad de utilizar otros lenguajes de programación que sirvan de apoyo para interactuar dinámicamente con el usuario, como JavaScript.

5. BIBLIOGRAFÍA

CMS. (27 de Marzo de 2008). *SELECTING A DEVELOPMENT APPROACH*. Obtenido de Sitio Web CMS: <https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>

Paloma Cáceres, E. M. (s.f.). *Procesos Ágiles para el desarrollo de Aplicaciones Web*. Universidad Rey Juan Carlos, Madrid.

Hernán, S. M. (2004). *diseño de una Metodología Ágil de Desarrollo de Software*. Universidad de Buenos Aires. Argentina: Universidad de Buenos Aires.

Raya, R. (24 de Julio de 2014). *¿Qué son las metodologías ágiles?* Recuperado el 19 de Noviembre de 2015, de <http://blog.leanmonitor.com/es/que-son-las-metodologias-agiles/>

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

Mas Sobre UML. (2005). En S. Schach, *Ingeniería de software clásica y orientada a objetos* (págs. 497 - 511). México: México.

Que es UML? (1999). En M. Fowler, *UML gota a gota* (págs. 1-13). México: México.

García, J., Moreira, A., & Rossi, G. (Marzo de 2004). *UML: el lenguaje estándar*. Recuperado el 2016 de 01 de 08, de UML e Ingeniería de Modelos:
<http://www.ati.es/novatica/2004/168/168-4.pdf>

Booch, G., Rumbaugh, J., & Jacobson, I. (2000). *El Lenguaje Unificado de Modelado*. Madrid: Addison Wesley.

EcuRed. (18 de 01 de 2016). *Lenguaje de Programación*. Obtenido de EcuRed:
http://www.ecured.cu/index.php/Lenguaje_de_Programaci%C3%B3n

CCM. (18 de 01 de 2016). *Lenguajes de programación*. Obtenido de CCM:
<http://es.ccm.net/contents/304-lenguajes-de-programacion>

Lenguaje de Programación. (18 de 01 de 2016). Obtenido de EcuRed:
http://www.ecured.cu/index.php/Lenguaje_de_Programaci%C3%B3n

Roman, C. (20 de 09 de 2015). *Lenguaje de Modelado Unificado*. Obtenido de Lenguaje de Modelado Unificado: <http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html>

DYNA, R. (18 de Enero de 2016). *Caso de Estudio*. Obtenido de Revista DYNA:
http://www.scielo.org.co/scielo.php?pid=S0012-73532007000300028&script=sci_arttext#fig02

Heurtel, O. (2015). *PHP 5.6 Desarrollar un sitio web dinámico e iterativo*. Barcelona: ENI.

Popov, N. (30 de Septiembre de 2014). *PHP RFC: Exceptions in the engine (for PHP 7)*. Obtenido de PHP: https://wiki.php.net/rfc/engine_exceptions_for_php7

Ken, D., & Pool, J. (11 de Mayo de 2015). *Características, ventajas y desventajas de las tecnologías de desarrollo Ruby on Rails y PHP*. Obtenido de Blog para las publicaciones de nuestras investigaciones ITIC92:
<https://jditic92.wordpress.com/2015/05/11/caracteristicas-ventajas-y-desventajas-de-las-tecnologias-de-desarrollo-ruby-on-rails-y-php/>

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

- Bustamante, D., & Rodriguez, J. (3 de Marzo de 2014). *Metodología XP*. Obtenido de Metodología de Desarrollo de Software:
<http://blogs.unellez.edu.ve/dsilva/files/2014/07/Metodologia-XP.pdf>
- Penadés, M. C., & Letelier, P. (06 de 2006). *Agile methodology for the development of the software: eXtreme Programming (XP)*. Obtenido de
http://www.cyta.com.ar/ta0502/b_v5n2a1.htm
- Tedeschi, N. (26 de Enero de 2016). *¿Qué es un Patrón de Diseño?* Obtenido de Microsoft | Developer Network: <https://msdn.microsoft.com/es-es/library/bb972240.aspx>
- Mestras, J. P. (09 de 2008). *Estructura de las Aplicaciones Orientadas a Objetos*. Obtenido de El patrón Modelo-Vista-Controlador (MVC):
<https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>
- Bascón, E. P. (2004). *El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing*. Obtenido de
<https://181.188.189.42/universidad/Publicaciones/revistas/actanova/documentos/v2n4/v2.n4.bascon.pdf>
- Date, C. (2001). *Introducción a los Sistemas de Bases de Datos*. Pearson.
- Ramírez, R. Z. (14 de 01 de 2008). *SISTEMAS GESTORES DE BASE DE DATOS*. Obtenido de
http://www.csi-csif.es/andalucia/modules/mod_ense/revista/pdf/Numero_14/RAQUEL_ZAMBRANO_2.pdf
- Gutierrez, P. (05 de 11 de 2013). *Fundamento de las bases de datos: Modelo entidad-relación*. Obtenido de <http://www.genbetadev.com/bases-de-datos/fundamento-de-las-bases-de-datos-modelo-entidad-relacion>
- Bruce, M. (2001). *PostgreSQL: Introduction and Concepts*. Addison-Wesley .
- Arana, B. O. (2014). *PostgreSQL, La base de datos más potente*. Madrid: Altaria.
- Gutierrez, D. (2010). *MVC*. Universidad de los Andes.
- Sánchez, J. (2004). *Principios sobre Bases de Datos Relacionales*. USA. Obtenido de
<http://www.jorgesanchez.net/bd/bdrelacional.pdf>

DISEÑO E IMPLEMENTACIÓN DEL VOTO ELECTRÓNICO MEDIANTE EL USO DE UNA APLICACIÓN WEB, PARA LAS ELECCIONES DE LA FEDERACIÓN DE ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

Fundamentos de Bases de Datos. (s.f.). Obtenido de Fases y resultados en el diseño de BDs:

<http://www.fing.edu.uy/inco/cursos/bdatos/teorico/3-DisenoConceptual.pdf>

LibrosWeb. (s.f.). Obtenido de La arquitectura MVC:

http://librosweb.es/libro/jobeeet_1_4/capitulo_4/la_arquitectura_mvc.html

Pantoja, E. B. (2004). *El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing.* Acta Nova.

Alicante, U. d. (s.f.). *Modelo vista controlador (MVC).* Obtenido de ASP.NET MVC 3 Framework:

<http://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

Alicante, U. d. (s.f.). *ASP.NET MVC 3 Framework.* Obtenido de Modelo vista controlador (MVC):

<http://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

Ciclo de Vida - Metodología XP. (s.f.). Obtenido de

http://cursa.ihmc.us/rid=1224363621984_1758846095_1407/PROCESO%20%20XP.cm
ap

Patron Modelo Vista-Controlador. (19 de 02 de 2013). Obtenido de

<https://victdelr.wordpress.com/2013/02/19/patron-modelo-vista-controladormodelo-de-4-capas/>

Lyle, W. F. (2013). *Programming Languages.*

jQuery Intro. (15 de 06 de 2016). Obtenido de w3schools:

http://www.w3schools.com/jquery/jquery_intro.asp