

Universidad de las Ciencias Informáticas

Facultad 15



Título: Sistema de Registro Electrónico Directo para las votaciones en la Universidad de las Ciencias Informáticas.

Trabajo de Diploma para optar por el título de Ingeniería en Ciencias Informáticas.

Autoras: Yunet Álvarez Almeida

Lisbey González Cárdenas

Tutor: Ing. Deibys Greguas Navarro

Ciudad de La Habana, 2010.

"Año 52 de la Revolución"

Dedicamos este sueño hecho realidad a:

A mi tesoro máspreciado, mi mamita linda, por su infinito amor, por el apoyo y la confianza que me dio en todo momento y por haberme dado los mejores consejos del mundo.

A mi papito, por creer en mí, por el amor que siempre me ha profesado, por enseñarme a luchar por mis sueños y darme fuerzas para seguir adelante en mis momentos más vulnerables.

A mi ángel de la guarda, Yusnier, por haberme regalado todo su cariño, amor, comprensión, y apoyo principalmente en los momentos más difíciles.

Yunet

A mi mamá que este título es y será por siempre de ella, es la persona más importante de mi vida.

A mi familia que me guiaron por los mejores caminos, me brindaron su mano en los momentos tristes, por hacerme ver la vida más clara y por demostrarme que podía ser capaz de alcanzar nuevas metas para el futuro.

A Isabel que la quiero como una hermana, por todos esos momentos de alegría que compartimos y espero vernos pronto.

A mis verdaderos amigos, por estar a mi lado cuando los necesité.

A mi compañera de tesis Yunet por ser mi brazo derecho en los momentos difíciles.

A todos los que de una forma u otra me ayudaron a hacer mi sueño realidad.

Lisbey

Agradezco de todo corazón:

A mi tutor, Deibys Greguas Navarro, por tanta dedicación, por enseñarme a ser cada día mejor y por el apoyo incondicional que me dio en todo momento.

A mi hermana Yanet, por estar ahí siempre cuando la necesité y a mis sobrinos Bryan y Ana Karla, por ser la fuente de mi inspiración.

A toda mi familia, por confiar en mí y contribuir infinitamente en la realización de mis sueños.

A mi tía Ailyn, por su constancia, preocupación y sus valiosos consejos.

A Marine y Cuco, por ser mi segunda familia, por todo el apoyo y la comprensión que me dieron en estos 5 años de la carrera.

A Yanet y Dunia, por su ayuda y preocupación en los momentos más difíciles.

A mis amigos Pedro, Franklín y Jorge por ayudarme siempre que lo necesité.

A mis amigas, Dayana, Lisbey, Yisel e Igniris por ser tan especiales.

A mi compañera de tesis, por tanta paciencia ante mis ataques de estrés.

A todas las personas que me han apoyado en el desarrollo del presente trabajo y de manera general durante todo el transcurso de mi vida universitaria. Gracias por influir positivamente en mi formación profesional.

A la Revolución por darme la oportunidad de realizar mi sueño.

A todos ustedes, muchas gracias.

Yunet

A mi mamá, por su ayuda incondicional en todos estos años, por darme siempre ese cariño tan especial y a ella le debo lo que soy.

A toda mi familia que aunque no las he tenido a mi lado en estos años de la carrera estuvieron ayudándome en los momentos más difíciles.

A mis amistades gracias por estar preocupados por mí y en especial a Yunet por brindarme su apoyo en todo momento que lo necesité.

A mi tutor por tanta dedicación y por estar siempre disponible ante cualquier duda.

Gracias a todos lo que aportaron su granito en la realización de esta tesis.

A todos de corazón ¡Muchas Gracias!

Lisbey



La posibilidad de realizar un sueño es lo que hace que la vida sea interesante.

Paulo Coelho

Declaramos que Yunet Álvarez Almeida y Lisbey González Cárdenas somos las únicas autoras del trabajo de diploma Sistema de Registro Electrónico Directo para las votaciones en la Universidad de las Ciencias Informáticas y otorgamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo, para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Yunet Álvarez Almeida

Lisbey González Cárdenas

Firma del Autor

Firma del Autor

Ing. Deibys Greguas Navarro

Firma del Tutor

Ing. Deibys Greguas Navarro

Correo: dgreguas@uci.cu

Profesor adjunto.

Instructor recién graduado en el 2008 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI).

Miembro del grupo de desarrolladores del proyecto Match Mind.

Diseñador gráfico de los proyectos Telebanca y Sistema de Gestión Penitenciaria (SIGEP), ambos en actual funcionamiento.

Miembro del grupo de desarrolladores de la Dirección de Informatización de la Infraestructura Productiva de la UCI.

Resumen

En la Universidad de las Ciencias Informáticas (UCI) se desarrollan procesos electorarios frecuentemente. La gestión de la información relacionada con estos procesos se realiza de forma manual, al igual que las votaciones se efectúan mediante el voto tradicional, pues no existe un software que facilite la realización de dichos procesos.

Dentro del proceso de informatización que se está desarrollando en la UCI, se ha concebido la realización del Sistema de Registro Electrónico Directo para las votaciones en la UCI, como una solución informática para apoyar el desarrollo de los procesos electorarios de las diferentes organizaciones políticas y de masas existentes en la universidad.

El presente trabajo de diploma expone el desarrollo del Sistema de Registro Electrónico Directo para las votaciones en la UCI, en el mismo quedan plasmadas las características de las herramientas analizadas, y las seleccionadas para el desarrollo del sistema. Se especifican cada uno de los procesos que se pretenden automatizar, así como las principales funcionalidades que debe brindar el sistema. Se realiza la descripción de los casos de uso del sistema generando los artefactos que dan entrada al diseño, el cual facilitará la implementación del sistema propuesto.

La utilización de este sistema en los procesos electorarios que se desarrollan en la universidad tiene como propósito optimizar la utilización de los recursos materiales y humanos empleados en el desarrollo de estos procesos.

Palabras claves

Sistema de Registro Electrónico Directo, Procesos Electorarios.

Tabla de contenidos

Introducción.....	1
Capítulo 1.....	5
1.1. Introducción del capítulo.....	5
1.2. Sistemas electorales.....	5
1.2.1. Sistema de Votación Tradicional.....	5
1.2.2. Sistemas de Votación Digital.....	6
1.2.3. Sistemas de Registro Electrónico Directo.....	8
1.2.4. Justificación del tipo de Sistema de Votación Digital seleccionado.....	12
1.3. Tecnologías, metodologías y herramientas de desarrollo.....	13
1.3.1. Tipos de sistemas informáticos.....	13
1.3.2. Servicios Web.....	15
1.3.3. Tecnologías del cliente.....	17
1.3.4. Tecnologías del servidor.....	19
1.3.5. Metodologías de desarrollo.....	21
1.3.6. Lenguajes de modelado de software.....	24
1.3.7. Herramientas CASE (Computer Aided Software Engineering).....	26
1.3.8. Sistemas Gestores de Base de Datos.....	27
1.3.9. Entornos de desarrollo.....	29
1.3.10. Framework.....	30
1.3.11. Fundamentación de la selección de las tecnologías, metodología y herramientas de desarrollo.....	32
1.4. Conclusiones parciales.....	34
Capítulo 2.....	35
2.1. Introducción del capítulo.....	35
2.2. Objeto de automatización.....	35
2.3. Información que se manipula.....	36
2.4. Propuesta de sistema.....	36
2.5. Modelo de negocio.....	37
2.5.1. Descripción del negocio basado en procesos.....	37

2.6.	Especificación de requisitos.....	39
2.6.1.	Requerimientos funcionales.....	39
2.6.2.	Requerimientos no funcionales	41
2.6.3.	Validación de requisitos.....	42
2.7.	Modelo de sistema	44
2.7.1.	Patrones de Casos de Uso	46
2.8.	Conclusiones parciales.....	47
Capítulo 3.....		48
3.1.	Introducción del capítulo.....	48
3.2.	Modelo del diseño	49
3.2.1.	Fundamentación del uso de Patrones.....	49
3.2.2.	Diagrama genérico de las clases del diseño.....	51
3.2.3.	Diagramas de interacción.....	52
3.3.	Diseño de la Base de Datos	53
3.3.1.	Modelo conceptual	53
3.3.2.	Modelo lógico de datos.....	54
3.3.3.	Modelo físico de datos.....	54
3.4.	Conclusiones parciales.....	56
Capítulo 4.....		57
4.1.	Introducción	57
4.2.	Diagrama de despliegue.....	57
4.3.	Diagrama de componentes	58
4.4.	Métodos de prueba	59
4.4.1.	Resultados de las pruebas de caja negra	60
4.4.2.	Resultados de las pruebas de caja blanca.....	61
4.5.	Conclusiones parciales.....	64
Conclusiones Generales		65
Recomendaciones.....		66
Referencias Bibliográficas.....		67
Anexos		69

Índice de figuras

Ilustración 1: Diagrama de los flujos de trabajo a lo largo del proyecto	23
Ilustración 2: Diagrama del proceso de negocio Organizar Proceso Eleccionario	38
Ilustración 3: Diagrama del proceso de negocio Efectuar Voto	38
Ilustración 4: Diagrama del proceso de negocio Emitir Resultados	39
Ilustración 5: Calidad de la especificación de requisitos	44
Ilustración 6: Descripción de los actores del sistema	45
Ilustración 7: Diagrama de Casos de Usos del Sistema	46
Ilustración 8: Diagrama genérico de las clases del diseño	52
Ilustración 9: Modelo Entidad-Relación de la Base de Datos	55
Ilustración 10: Diagrama de despliegue.....	58
Ilustración 11: Diagrama de componentes genéricos.....	59
Ilustración 12: Resultados de la primera iteración de los CP	60
Ilustración 13: Resultados de la segunda iteración de los CP	61
Ilustración 14: Resultado insatisfactorio del método Buscar Candidato.....	62
Ilustración 15: Resultado satisfactorio del método Buscar Candidato.....	62
Ilustración 16: Resultado insatisfactorio del método Registrar	63
Ilustración 17: Resultado satisfactorio del método Registrar Urna.....	63
Ilustración 18: Autenticarse en el sistema	69
Ilustración 19: Funcionalidad Registrar Candidato	69

Introducción

El derecho al voto o sufragio universal es un privilegio que se le otorga a los ciudadanos, que concede gran importancia al resultado de las elecciones, debido a su capacidad para legitimar la conformidad con la toma de decisiones y el respaldo que los electores dan a la administración imperante mediante procesos electorales.

En las primeras elecciones que se convocaron en Cuba el 16 de junio de 1900, bajo el gobierno interventor norteamericano, con el fin de elegir a las autoridades municipales, luego del cese de la dominación española, votó el siete por ciento de la población. En este proceso eleccionario primó el fraude, lo que obligó a la oposición a alzarse en armas contra el gobierno. Desde entonces se mantuvieron las elecciones fraudulentas hasta el cambio de gobierno el 1ero de enero de 1959, Triunfo de la Revolución Cubana, lo que hizo posible la incorporación del pueblo en la conducción real de la sociedad.

Los cubanos son privilegiados por el derecho al voto y por la realización de las elecciones mediante un proceso electoral transparente, en las que el pueblo postula y elige a sus representantes; éstos son elegidos si alcanzan más del 50 por ciento de los votos válidos emitidos, rinden cuenta a sus electores y pueden ser revocados en cualquier momento de su mandato. Todos los electores tienen el derecho a ser elegidos como delegados a partir de los 16 años, y como diputados a la Asamblea Nacional a los 18.

El cuestionamiento al sistema político y electoral cubano, constituye uno de los pilares fundamentales de la campaña enemiga contra nuestro país liderada por los Estados Unidos. La actividad contra Cuba en materia de democracia y derechos humanos, no sólo constituye la principal herramienta utilizada para tratar de “legitimar” su política de hostilidad y agresión hacia la isla; sino que responde también al interés de los principales países capitalistas industrializados de imponer a los países en desarrollo un modelo de organización política que facilite la dominación.

El sistema de democracia participativa cubana continúa fortaleciéndose, lo cual se demuestra en los procesos de elecciones generales en los que por voto directo y secreto se postulan los candidatos a delegados a las Asambleas Municipales, Provinciales y Nacional del Poder Popular; y en los procesos eleccionarios que se desarrollan en organizaciones políticas y de masas, tales como: la Unión de Jóvenes Comunistas (UJC), la Federación Estudiantil Universitaria (FEU), el Partido Comunista de Cuba (PCC) y el Sindicato.

En la Universidad de las Ciencias Informáticas (UCI), para desarrollar los procesos electorales correspondientes a cada una de las organizaciones políticas y de masas, se utiliza el sistema de votación tradicional, el desarrollo de estos procesos electorales requiere una planificación estratégica y operativa, lo que implica: un manejo masivo de información como el listado de los candidatos seleccionados con sus nombres, apellidos, grupo, facultad y currículos correspondientes; además, se necesita utilizar gran cantidad de recursos materiales para la impresión de las boletas y los padrones electorales y de personas para el acondicionamiento de los escrutinios, el control del personal que accede a las urnas y el conteo manual de los votos. La realización manual de estos procesos obstaculiza el monitoreo del progreso de las votaciones, lo que induce retrasos en la comunicación de los resultados.

Por la importancia del desarrollo de procesos eleccionarios eficientes y la necesidad del ahorro de recursos materiales y humanos se plantea como **Problema Científico**: ¿Cómo mejorar el Proceso Eleccionario en la Universidad de las Ciencias Informáticas de forma que se optimicen la utilización de los recursos materiales y humanos?

Para el desarrollo de la investigación se plantea como **Objeto de Estudio**: El proceso de Desarrollo de Software.

El cual tiene como **Campo de Acción**: El desarrollo de aplicaciones Web para la gestión de la información de las votaciones en la UCI.

Planteando como **Idea a defender**: el empleo de una aplicación Web que gestione la información de las votaciones en la UCI, optimizará la utilización de recursos materiales y humanos.

Por lo que el **Objetivo General** de la investigación es: desarrollar una aplicación Web que gestione la información de las votaciones en la UCI y optimice la utilización de recursos materiales y humanos.

Para lograr el cumplimiento del objetivo propuesto se trazaron las siguientes **Tareas de la Investigación**:

1. Fundamentación teórica de la investigación.
2. Análisis y selección de la metodología, tecnologías y herramientas para desarrollar la aplicación Web.
3. Modelamiento del negocio.
4. Realización de las actividades pertenecientes al flujo de trabajo de requerimientos del producto.

5. Realización de las actividades pertenecientes al flujo de trabajo de análisis y diseño del sistema.
6. Implementación de la aplicación Web.
7. Realización de las actividades de prueba de liberación del producto implementado.

En sentido general, se puede destacar que el cumplimiento de las tareas de la investigación, facilitará las labores de la administración electoral de la UCI, perfeccionando los procesos electorales por medio de nuevas técnicas de automatización, ofreciendo un servicio más confiable, eficiente, seguro y ágil a los electores, un ejemplo de ello lo constituye la automatización de los padrones electorales, proceso que facilita la obtención de una lista actualizada de los votantes habilitados, que inciden directamente en la calidad del padrón. Brinda, además, la posibilidad de selección inmediata de los candidatos y conocer la cantidad de votos en tiempo real, lo que acelera la etapa de escrutinio y consecuentemente la entrega de los resultados. Se aspira obtener un sistema reutilizable, capaz de llevar a cabo la ejecución de las distintas actividades relacionadas con el proceso electoral en la UCI, homogéneo, fácil de usar y que obtenga la satisfacción de los usuarios finales.

Métodos Científicos utilizados en la investigación:

Se utilizan diferentes métodos científicos para realizar la investigación. Como método **Teórico** se emplea el Analítico-Sintético pues uno de los primeros pasos para llevar a cabo la presente tesis es buscar y analizar documentos e información sobre el desarrollo de los procesos eleccionarios en la UCI y extraer los elementos más importantes relacionados con el objeto de estudio. Otro método teórico que se aborda en la investigación es el método Histórico-Lógico para analizar a nivel internacional y nacional el empleo de sistemas informáticos similares al que se propone, así como investigaciones realizadas anteriormente sobre el tema. Los métodos inductivo y deductivo se utilizan en diferentes etapas de la investigación, especialmente cuando de las ideas generales se deducen las particulares.

Dentro de los métodos **Empíricos** se usan las Entrevistas, estas se les realizan a los trabajadores de la administración electoral de la UCI, en específico, a las personas relacionadas con las votaciones dentro de las organizaciones políticas y de masas de la UCI, para de esta forma recoger toda la información referente al funcionamiento del proceso eleccionario. También se

utiliza la Encuesta para conocer la opinión de las personas que van a interactuar con la aplicación y poder complementar los requerimientos del cliente.

El documento presenta una estructura por capítulos como se muestra a continuación:

CAPÍTULO 1. Fundamentación Teórica: Se exponen los conceptos fundamentales asociados al desarrollo de aplicaciones Web para la gestión de la información de los procesos electorarios. Se formaliza un estudio del estado del arte de los sistemas de votación digital, tanto a nivel nacional como internacional, realizando un estudio de las soluciones ya existentes; lo que permite tener un juicio valorativo para seleccionar el tipo de sistema de votación digital para solucionar la problemática planteada. Fundamentación de las tendencias, tecnologías, metodología y herramientas de desarrollo en las que se apoya la solución al problema.

CAPÍTULO 2. Características del Sistema: Se hace una descripción de la situación problemática, se identifican los procesos a automatizar, creando un entorno de trabajo en el que se desarrollará el sistema, se realiza un estudio de la información que se manipula y se hace una propuesta del sistema a desarrollar. Se define el Modelo del negocio, se especifican los requerimientos, validándolos mediante métricas y se confecciona el Modelo de Sistema.

CAPÍTULO 3. Diseño del Sistema: Se define el diseño del sistema a desarrollar, realizando un diagrama genérico del diseño basado en la utilización del framework a utilizar y especificando uno por cada caso de uso, se detalla un diagrama de secuencia por cada sección de caso de uso y se describen de las clases del diseño. Se realiza el modelo de datos describiéndose las tablas existentes logrando un diseño que cumpla con las necesidades del cliente.

CAPÍTULO 4. Implementación y Prueba: Se confecciona el diagrama de despliegue y el de componentes con la explicación detallada de cada componente y su funcionalidad, detallando un diagrama de componentes para subsistema de implementación. Se describe el modelo de prueba, realizando un Caso de Prueba por cada Caso de Uso del Sistema.

Capítulo 1

1.1. Introducción del capítulo

La utilización de un sistema electoral le permite a los electores expresar su voluntad política mediante votos, para seleccionar el candidato que los representará. En este capítulo se expondrán definiciones sobre los sistemas electorales, se analizará la evolución y los diferentes tipos de Sistemas de Votación Digital existentes en el mundo. Además, se realizará un estudio de las principales tecnologías, metodologías y herramientas de desarrollo de software, con el objetivo de seleccionar las que serán usadas para desarrollar la aplicación.

1.2. Sistemas electorales

1.2.1. Sistema de Votación Tradicional

El término voto se utiliza en el lenguaje natural como sinónimo de sufragio, sin embargo, desde el punto de vista jurídico, la relación entre ambas no es necesariamente de identidad.

No obstante, el íntimo enlace entre ambos hace que en el lenguaje llano se entienda como sufragio al voto del elector, englobando ambos como “al hecho de declarar su voluntad en los comicios o sea las reuniones de los ciudadanos para sufragar”. (Rico, 1997)

Un **sistema electoral** es el conjunto de principios, normas, reglas, procedimientos técnicos enlazados entre sí, y legalmente establecidos, por medio de los cuales los electores expresan, su voluntad política en votos que a su vez se convierten en escaños o poder. (Electoral, 2005)

Los **procesos electorales** son actos que permiten transformar la voluntad de las personas o ciudadanos en cargos representativos mediante votaciones. (Electoral, 2005)

En el **Sistema de Votación Tradicional** el elector se presenta en el colegio electoral con su documento de identidad, el presidente de mesa revisa un padrón impreso para habilitarlo a emitir el sufragio. Le entrega la boleta electoral impresa con las opciones de candidatos y la persona ingresa al cuarto oscuro. De acuerdo con su decisión, el ciudadano podrá votar la lista completa, realizar corte de boleta, votar la lista incompleta o votar en blanco. Sale, ingresa la boleta dentro de la urna, recupera su documento de identidad y firma el padrón electoral, quedando inhabilitado para volver a votar. A la hora del cierre de los comicios se abren las urnas para el recuento

manual de los votos y finalmente comunicar los resultados del proceso electoral. (Bunge, 2004)

En la UCI los procesos electorarios se realizan mediante el sistema de votación tradicional, para el desarrollo de los mismos se han identificado dos tipos de votaciones:

Abiertas: Los electores pueden elegir el número de candidatos para miembro que considere necesario, sin anularse la boleta electoral, además brinda la posibilidad de elegir al presidente, con la condición de que este candidato debió ser elegido como miembro previamente. Un ejemplo de organización que realiza las votaciones Abiertas es la FEU.

Cerradas: Solo se puede votar por un número cerrado de candidatos, que debe ser especificado por el organizador del proceso electoral antes de efectuar la votación, las boletas electorales que no cumplan este requisito, serán anuladas. En este tipo de votación cerrada se elige al presidente mediante el voto a mano alzada, entre los miembros una vez que estos fueron seleccionados. Un ejemplo de organización que realiza las votaciones cerradas es la UJC.

1.2.2. Sistemas de Votación Digital

Las nuevas tecnologías de información y comunicación se han masificado al inicio del siglo XX, alcanzando la esfera de los procesos políticos y sociales. La incorporación de tecnología en la función pública no es un hecho nuevo. Los Sistemas de Votación Digital son uno de los resultados alcanzados por los procesos de informatización que se vienen desarrollando a nivel mundial.

1.2.2.1. Evolución de los Sistemas de Votación Digital

Durante décadas, en la construcción de la democracia, han imperado las formas de votación manuales, por medio de la utilización de bolígrafos o marcadores, pero consecuentemente con el avance en los métodos de votación se han desarrollado sistemas de votaciones digitales, que agilizan los procesos electorarios, optimizan la utilización de recursos materiales y humanos, aceleran la etapa de escrutinio y la entrega de los resultados. (Rojas, 2000)

Unos de los primeros intentos de automatización lo constituye la máquina de votación por palanca. Esta máquina de votación fue patentada por el inventor Alfred J. Gillespie y fabricada por Norma Votar Machine Compañía de Rochester, en Nueva York en 1890. (Busaniche, et al., 2008)

En la década de 1960 un nuevo sistema de votación fue introducido en el mercado y finalmente triunfó sobre las máquinas mecánicas. Estas fueron las tarjetas perforadas, se usaron en la elección presidencial de 1964 en Estados Unidos. En 1982, aproximadamente la mitad del electorado estadounidense efectuó la votación mediante sistemas de tarjetas perforadas. (Busaniche, et al., 2008)

Seguidamente se introdujeron los sistemas de votación mediante escaneo óptico. Estas papeletas son leídas por una máquina que escanea los votos y en la década de 1990 se inició el uso de los Sistemas de Registro Electrónico Directo (RED) mediante la pantalla táctil de votación. (Busaniche, et al., 2008)

Los sistemas RED son un medio electrónico de la aplicación de los viejos sistemas mecánicos de palanca. Al igual que con las máquinas de palanca, las opciones posibles son visibles para el votante en la parte delantera de la máquina. El votante entra directamente en las opciones de almacenamiento electrónico con el uso de una pantalla táctil, botones, o dispositivo similar y efectúa el voto electrónico. (Busaniche, et al., 2008)

1.2.2.2. Tipos de Sistemas de Votación Digital

La tecnología de voto electrónico incluye fundamentalmente tres tipos de sistemas, estos difieren en su implementación y en los riesgos y beneficios que proporcionan a los electores.

Actualmente, la mayoría de los **Sistemas de Recuento Automático** de votos se basan en el reconocimiento óptico de las marcas hechas por el votante sobre la boleta, ya sea de forma directa o a través de una máquina de marcar boletas. Este tipo de sistema permite contabilizar los votos mediante un sistema de reconocimiento óptico de caracteres. El principio de que la voluntad del elector se expresa en un papel, no permite el ahorro de recursos materiales y facilita la posibilidad de alterar o anular el voto. Este mecanismo aumenta los costos de la aplicación del sistema, ya que requiere un escáner por cada colegio electoral. El paso de cada boleta electoral por el escáner antes de ser introducida en la urna, permite registrar la secuencia en la que se emitieron los votos, y así reconstruir la relación de cada votante con su voto, perdiendo de esta forma el anonimato. (Busaniche, et al., 2008)

Los **Sistemas de Votación a Distancia** a través de internet, son un mecanismo que permite emitir el sufragio desde una computadora común conectada a la red de redes, desde un teléfono celular o desde un sistema de televisión digital. Este tipo de sistema no permite el control sobre la correcta identificación de los votantes, por lo que es imprescindible asegurar que alguien vote en

nombre de otra persona. La máquina que recibe el voto tiene conocimiento de quien lo está emitiendo, la información almacenada en este servidor permite conocer la relación entre el elector y el voto, violando así el secreto del mismo. (Garza, 2008)

Los **Sistemas de Registro Electrónico Directo** (RED) son los que más se corresponden con las denominadas urnas electrónicas. Estos sistemas se caracterizan por realizar simultáneamente el registro y la tabulación del voto, a través de un dispositivo informático que es operado directamente por el votante, mediante un teclado o una pantalla táctil. También pueden configurarse de tal modo que permitan al usuario corregir sus opciones, de forma tal que impidan invalidar o anular el voto. Estos sistemas optimizan la utilización de recursos al no ser necesaria la utilización de hojas para la impresión de las boletas y los padrones electorales. Además, disminuye la intervención de personas en este proceso, debido a que el conteo de los votos se efectúa de forma automática, agilizando la comunicación de los resultados. (Busaniche, et al., 2008)

1.2.3. Sistemas de Registro Electrónico Directo

La constante evolución que experimenta la tecnología de cómputo desde finales de la década de 1990, desembocó en el desarrollo de los sistemas de votación electrónica más recientes: los Sistemas de Registro Electrónico Directo (RED).

Un Sistema RED es una máquina que utiliza boletas electrónicas para registrar por medio de una pantalla de votación los datos emitidos por los electores a través de un programa informático, este sistema es el que más se asemeja a la votación tradicional. (Electoral, 2005)

Una de las características más salientes de este tipo de sistema RED reside en que permite unificar la emisión y el escrutinio de los votos en una misma máquina.

1.2.3.1. Ventajas de los Sistemas de Registro Electrónico Directo

Generalmente, la aspiración asociada a la automatización de las votaciones, es agilizar este proceso así como el de la transmisión de los resultados, optimizando la utilización de recursos tanto materiales como humanos.

La incorporación de tecnología en la etapa del proceso electoral de emisión del voto, mediante el uso de las urnas electrónicas, proporciona ventajas sobre el sistema electoral tradicional, tales como:

- ✓ Ofrecen mayores facilidades al elector para el ejercicio del derecho al sufragio.
- ✓ Los resultados se caracterizan por su alta precisión.
- ✓ Aceleran el proceso de obtención de los resultados del escrutinio.
- ✓ Cuentan con un sistema de respaldo automático.
- ✓ Disminuyen la abstención de votos.
- ✓ Optimizan la utilización de recursos materiales y humanos. (Electoral, 2005)

Una posible **desventaja** es que los electores no están exentos a la ocurrencia de fallos electrónicos durante el proceso de votación.

1.2.3.2. Seguridad de los Sistemas de Registro Electrónico Directo

La seguridad es el requisito fundamental para lograr la confiabilidad en un sistema cualquiera.

Para que un sistema se pueda definirse como seguro debe basarse en estos principios:

Integridad: Se refiere a la validez y consistencia de los elementos de información almacenados y procesados en un sistema informático. Basándose en este principio, las herramientas de seguridad informática deben asegurar que los procesos de actualización estén sincronizados y no se dupliquen, de forma que todos los elementos del sistema manipulen adecuadamente los mismos datos. Este principio es particularmente importante en sistemas descentralizados, es decir, aquellos en los que diferentes usuarios, computadores y procesos comparten la misma información.

Confidencialidad: Se refiere a la privacidad de los elementos de información almacenados y procesados en un sistema informático. Basándose en este principio, las herramientas de seguridad informática deben proteger al sistema de invasiones, intrusiones y accesos, por parte de personas o programas no autorizados. Este principio es particularmente importante en sistemas distribuidos, es decir, aquellos en los que usuarios, computadores y datos residen en localidades diferentes, pero están física y lógicamente interconectados.

Disponibilidad: Se refiere a la continuidad de acceso a los elementos de información almacenados y procesados en un sistema informático. Basándose en este principio, las

herramientas de Seguridad Informática deben reforzar la permanencia del sistema informático, en condiciones de actividad adecuadas para que los usuarios accedan a los datos con la frecuencia y dedicación que requieran. Este principio es particularmente importante en sistemas informáticos cuyo compromiso con el usuario, es prestar servicio permanente.

Irrefutabilidad (No repudio): El uso y/o modificación de la información por parte de un usuario debe ser irrefutable, es decir, que el usuario no puede negar dicha acción.

Los Sistemas de Registro Electrónico Directo pueden cumplir con algunos de estos principios, si en ellos se aplican los siguientes mecanismos de seguridad informática:

Control de Acceso: La autenticación debe ser la primera acción del usuario en el sistema, permitiendo suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica. Los elementos lógicos como ficheros y la base de datos, así como los físicos como los servidores y las urnas electrónicas deben estar protegidos de accesos no autorizados, previniendo que un atacante haga uso indebido de los mismos.

Terminales de votación seguros: Las urnas electrónicas donde los electores emiten el sufragio deben ser supervisadas por los Responsables de los Colegios Electorales, apoyados por medidas de seguridad como: Antivirus y Firewall. Los antivirus ejercerán el control preventivo y correctivo sobre ataques de virus al sistema y el Software “firewall” lo realizará sobre intrusiones no deseadas.

Canales de comunicación seguros: Las transacciones llevadas a cabo durante el proceso de elecciones entre las urnas digitales y el servidor de aplicaciones, y entre la computadora que administra el proceso electoral y el servidor de aplicaciones, deberá realizarse mediante el Protocolo de Transferencia de Hipertexto Seguro (HTTPS).

Prevenir la ejecución de código remotamente: La ejecución de código remotamente es una vulnerabilidad que permite al atacante ejecutar código en el servidor vulnerable y obtener información almacenada en él. Se debe configurar el sistema, de forma que se pueda controlar la disponibilidad de las variables globales, para evitar que estas sean inicializadas remotamente.

Verificar las trazas: Los sistemas RED deben utilizar un mecanismo que permita comprobar la ejecución de las validaciones de datos previstas.

Un sistema informático nunca va hacer totalmente seguro, la aplicación de estas medidas de seguridad solo van a dificultarle la entrada a personas que deseen poner en jaque la seguridad del sistema.

1.2.3.3. Soluciones similares de Sistemas de Registro Electrónico Directo

A nivel internacional existe un crecimiento acelerado del empleo de sistemas de votaciones digitales mediante RED. Estos sistemas RED son el modelo preferido de una gran cantidad de países. Las urnas electrónicas usadas en Brasil, así como en varios estados de EE.UU. o en las últimas elecciones de Venezuela son sistemas RED.

Existen muchos proveedores para desarrollar y comercializar sistemas de Registro Electrónico Directo para efectuar votaciones. A continuación se citan algunos ejemplos:

AccuVote-TS

Su proveedor es Diebold Election Systems. Este sistema incluye pantalla táctil, lector de tarjetas, teclado, de auriculares, y una impresora de cinta de papel. El votante elige a su candidato favorito en la pantalla táctil, y la votación se imprimirá en de la cinta de papel. La información electoral, incluyendo la identidad de autenticación, auditoría o recuento de los votos, se almacenan en datos de Microsoft Access sin configuración de la contraseña por lo que hay de alta los riesgos de ataque. Fue utilizada en las elecciones del 2004 en los Estados Unidos, muestra como en cuestión de minutos se puede vulnerar la maquina, inyectando código malicioso que altera de forma imperceptible los resultados finales de las elecciones, con el menor riesgo de ser detectado. (Tula, 2007)

IVotronic

La empresa proveedora de este sistema es Election Systems and Software (ES & S). IVotronic proporciona una plataforma multilenguaje, y utiliza memoria flash para guardar los registros de votación. Los trabajadores electorales usan PEB (Voto Electrónico Personal, un dispositivo que es similar en el disco) para iniciar la máquina de votación. Cuando la elección ha terminado, se usan los trabajadores PEB para acceder a los registros de voto máquina, luego se entregan los registros al centro electoral o se transmiten los datos desde la red.

Existe el riesgo de romper la contraseña de los trabajadores PEB. Este sistema ha presentado fallas, tales como: el número de los votantes no se corresponden con los almacenados en el

servidor, el candidato seleccionado en la boleta electoral no era el votante de selección, y así sucesivamente. (Tula, 2007)

SAVIOC

SAVIOC es una fuente abierta, está escrito en C, y puede ser guardado en el disco con FreeDOS. Funciona desde el disco, no está conectado a ninguna red y la mayoría de de las teclas del teclado son para discapacitados, los atacantes no pueden encontrar el camino para la invasión. SAVIOC tiene un costo bajo, pero carece de interfaz gráfica de usuario y facilidad de uso. (Tula, 2007)

En Cuba no se ha utilizado un sistema RED para efectuar las votaciones, los procesos eleccionarios desarrollados en las Asambleas Municipales, Provinciales y Nacional del Poder Popular; y en las organizaciones políticas y de masas, tales como: la Unión de Jóvenes Comunistas (UJC), la Federación Estudiantil Universitaria (FEU), el Partido Comunista de Cuba (PCC) y el Sindicato se realizan mediante el proceso eleccionario tradicional, de igual forma sucede en la Universidad de las Ciencias Informáticas.

1.2.4. Justificación del tipo de Sistema de Votación Digital seleccionado

Después de realizar un análisis exhaustivo de las características de los Sistemas de Votación Digital existentes a nivel mundial, se decide seleccionar a los Sistemas de Registro Electrónico Directo como el tipo de Sistema de Votación Digital, más adecuado para efectuar las votaciones en la UCI.

La selección está basada en:

- ✓ La utilización de un Sistema de Votación a Distancia permitirá efectuar las votaciones desde una PC conectada a Internet, pero la máquina que recibe el voto tiene conocimiento de quien lo está emitiendo, la información almacenada en este servidor permite conocer la relación entre el elector y el voto, violando así el secreto del mismo. Por la necesidad de efectuar procesos eleccionarios seguros no se decide utilizar este tipo de Sistema de Votación Digital.
- ✓ La utilización de un Sistema de Recuento Automático se basa en el reconocimiento óptico de las marcas hechas por el votante sobre la boleta, el principio de que la voluntad del elector se expresa en un papel, no permite el ahorro de recursos materiales y facilita la

posibilidad de alterar o anular el voto. Este mecanismo aumenta los costos de la aplicación del sistema, ya que requiere un escáner por cada colegio electoral.

- ✓ Las votaciones mediante los Sistemas de Registro Electrónico Director son las que más se asemejan al proceso tradicional, forma en que se realiza actualmente el proceso eleccionario en la UCI.
- ✓ La utilización de un Sistema de Registro Electrónico Directo para efectuar las votaciones en la UCI, optimizará la utilización de recursos materiales y humanos, pues ya no será necesaria la impresión de las boletas electorales ni la utilización de gran cantidad de personas para controlar y desarrollar el proceso eleccionario.
- ✓ Los datos obtenidos serán precisos, se mantiene el anonimato del votante, no se anularán boletas y se podrá monitorear el progreso de las votaciones en tiempo real, permitiendo acelerar el tiempo de comunicación de los resultados.

Se decide desarrollar un Sistema de Registro Electrónico Directo para efectuar las votaciones en la UCI, debido a que los procesos eleccionarios que se realizan en la UCI, ya sean Abiertos (como en la FEU) o Cerrados (como en la UJC), no se corresponden con los diseños de los sistemas ya existentes, además, estos sistemas están desarrollados con tecnologías propietarias, no utilizan Sistemas Gestores de Base de Datos potentes y se han demostrado fallas y vulnerabilidades de estos sistemas durante el proceso de votación.

Por estas razones se ofrece a continuación un análisis de diferentes tecnologías, metodologías y herramientas de desarrollo de software, para poder decidir cuáles utilizar en la implementación del Sistema de Registro Electrónico Directo para las votaciones en la UCI.

1.3. Tecnologías, metodologías y herramientas de desarrollo

1.3.1. Tipos de sistemas informáticos

Los **sistemas informáticos** son un conjunto de elementos interconectados o relacionados para el tratamiento de información. Estos sistemas permiten representar la información de forma eficiente y automatizada, transmitirla sin errores ni pérdidas, almacenarla para poder acceder a ella, recuperarla tantas veces como sea preciso y procesarla para obtener datos mejor elaborados. (Mateu, 2004)

1.3.1.1. Aplicaciones de escritorio

Por aplicaciones de escritorio o desktop se entiende toda aplicación que ha sido desarrollada para ser ejecutada en una plataforma específica, ya sea Windows, Linux o Mac. El desarrollo sobre una plataforma, normalmente, implica que la aplicación "no" pueda ser ejecutada en otras. Entre las ventajas de las aplicaciones de escritorio están: (Mateu, 2004)

- ✓ Mayor capacidad gráfica visual
- ✓ Menor tiempo de respuesta
- ✓ Mayor personalización

Estas aplicaciones están siempre disponibles en cualquier momento desde el escritorio a través de la bandeja del sistema o la barra de tareas de Windows. A diferencia de las aplicaciones web que requieren un navegador antes de que pueda comenzar a usarlo.

Algunas de las desventajas de estas aplicaciones son:

- ✓ Para que estas funcionen correctamente necesitan de la instalación en el cliente.
- ✓ Si se realiza una nueva versión de la aplicación el cliente debe adquirirla nuevamente.
- ✓ Duplicidad de datos por la falta de unificación de los mismos.
- ✓ Diseminación de la información y lógica en muchas partes (cada computador que la use).
- ✓ Falta de portabilidad de la aplicación a diferentes sistemas operativos.
- ✓ Traumas a la hora de realizar actualizaciones o correcciones al programa ya que las instalaciones están diseminadas.
- ✓ Dificultad para configurar cada una de las instalaciones dependiendo de las necesidades de cada usuario.

1.3.1.2. Aplicaciones Web

La aparición de la Web hizo posible que cualquier persona pueda ofrecer información particularizada a los demás y encontrar documentos interactivos sobre cualquier tema, relacionados unos con otros mediante enlaces que permiten navegar de página en página alrededor del mundo.

Las aplicaciones web son populares debido a la practicidad del navegador como cliente ligero (navegador de Internet), contiene elementos que permiten una comunicación activa entre el usuario y la información lo cual posibilita un modo interactivo con el cliente. La habilidad para

actualizar y mantener aplicaciones Web sin distribuir e instalar software es otra razón de su popularidad. (Mateu, 2004)

Utilizan lenguajes interpretados del lado del cliente, tales como Java Script, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página Web individual es enviada al cliente como un documento estático, pero la secuencia de páginas provee de una experiencia interactiva.

La creciente popularidad de las aplicaciones Web se debe a sus múltiples ventajas, entre las cuales se pueden citar: (Mateu, 2004)

Multiplataforma: Una misma versión de la aplicación puede correr sin problemas en múltiples plataformas como Windows, Linux, Mac, etc.

Actualización instantánea: Debido a que todos los usuarios de la aplicación hacen uso de la última versión del programa que radica en el servidor, los usuarios siempre utilizarán la versión más actualizada del sistema.

Acceso inmediato y desde cualquier lugar: Las aplicaciones basadas en tecnologías web no necesitan ser descargadas, instaladas y configuradas. Además, pueden ser accedidas desde cualquier computadora conectada a la red.

Menos requerimientos de hardware: Este tipo de aplicación no consume (o consume muy poco) espacio en disco y también es mínimo el consumo de memoria RAM en comparación con los programas instalados localmente. Tampoco es necesario disponer de computadoras con poderosos procesadores ya que la mayor parte del trabajo se realiza en el servidor donde reside la aplicación.

En las aplicaciones Web la información no es presentada de la misma forma en todos los navegadores, por ejemplo Mozilla Firefox e Internet Explorer.

1.3.2. Servicios Web

Los Servicios Web permiten la comunicación entre aplicaciones o componentes de aplicaciones de forma estándar a través de protocolos comunes (como http) y de manera independiente al lenguaje de programación, plataforma de implantación, formato de presentación o sistema

operativo. Un Servicio Web es un contenedor que encapsula funciones específicas y hace que estas funciones puedan ser utilizadas en otros servidores.

Existen organizaciones como OASIS y W3C que son los comités responsables para la arquitectura y orden de los Servicios Web. Con el objetivo de mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I que se encarga de ampliar diversos perfiles para una mejor definición estos estándares. (Sotés Mesa, y otros, 2009)

Web Services Protocol Stack, se denomina así al conjunto de servicios y protocolos, de los servicios Web:

XML (Extensible Markup Language): Es el formato estándar para los datos que se vayan a intercambiar (Aquino Leiva, y otros, 2009).

SOAP (Simple Object Access Protocol): Protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

UDDI (Universal Description, Discovery and Integration): Protocolo para publicar la información de los servicios Web. Permite comprobar qué servicios web están disponibles.

TCP/IP (Transmission-Control-Protocol/Internet Protocol): Es uno de los protocolos fundamentales para la comunicación, está diseñado para enrutar y tiene un grado muy elevado de fiabilidad, es adecuado para redes grandes y medianas, así como en redes empresariales. Se utiliza a nivel mundial para conectarse a Internet y a los servidores web. Es compatible con las herramientas estándar para analizar el funcionamiento de la red.

HTTP (HyperText Transfer Protocol): Es el protocolo usado en cada transacción de la Web, define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Además, es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. (Ricardo Velázquez, y otros, 2009)

Entre las **ventajas** que ofrecen estos servicios web se tienen:

- ✓ Brindan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- ✓ Los servicios Web provocan los estándares y protocolos basados en texto, ya que es más fácil acceder a su contenido y entender toda su tarea.

- ✓ Teniendo como apoyo HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- ✓ Admiten que servicios y software de distintas compañías ubicadas en diferentes lugares geográficos puedan ajustarse fácilmente para proveer servicios integrados.
- ✓ Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.

1.3.3. Tecnologías del cliente

Las tecnologías del cliente se caracterizan por la rapidez que brindan a los usuarios en comparación con las tecnologías por parte del servidor, pues se validan posibles errores antes que los datos sean enviados al servidor y de esta forma se gana en consistencia y velocidad, aunque es siempre recomendable hacer validaciones en el servidor. La programación del lado del cliente depende del navegador o explorador Web, si existe una mínima alteración la aplicación puede colapsar.

1.3.3.1. Java Script

Lenguaje de programación script que se utiliza fundamentalmente para crear páginas Web dinámicas, estas pueden implementar funcionalidades, acciones que se ejecutan al pulsar botones u otros elementos HTML¹, operaciones sencillas o complejas que luego son mostradas al usuario, así como mensajes de aviso, efectos visuales o animaciones.

Es un lenguaje interpretado, por lo que no es necesario compilar los programas para ejecutarlos. Los programas escritos en Java Script se pueden probar directamente en cualquier navegador sin necesidad de utilizar herramientas de compilación o cualquier proceso intermediario.

Utiliza un gestor automático de memoria dinámica, que reserva espacio para crear objetos y los elimina cuando éstos ya no se utilizan. Está basado en un conjunto predefinido de objetos, que pueden ser extendidos. Sin embargo, no es posible crear nuevas clases, o establecer relaciones de herencia. Permite utilizar funciones, al estilo de los lenguajes de programación orientados a procedimientos.

Java Script tiene algunas **limitaciones**, tales como:

¹**HTML**: por sus siglas en inglés **HyperText Markup Language** traducido al español como Lenguaje de Marcas de Hipertexto

- ✓ No puede interactuar con el servidor: trabaja enteramente dentro del navegador del cliente y no se comunica con el servidor ni con otras máquinas.
- ✓ No puede crear sus propios gráficos, solamente puede manipular gráficos hechos en otros programas.

1.3.3.2. CSS (Cascade Style Sheets)

Es un lenguaje de hojas de estilos creado con el objetivo de controlar la presentación de los documentos electrónicos definidos con HTML y XHTML². Es la mejor forma de separar contenido y presentación, y es indispensable para crear aplicaciones Web complejas. Separar contenido y presentación brinda numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

El lenguaje CSS³ se utiliza para definir el aspecto de todos los contenidos, el formato de tablas, la separación, el color, tamaño y tipo de letra de titulares y/o textos, la tabulación con la que se muestran los elementos de una lista o menú. (Pérez, 2009)

Ventajas:

- ✓ Control centralizado de la presentación de un sitio Web completo, con lo que se agiliza de forma considerable la actualización del mismo.
- ✓ Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario.
- ✓ El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño (siempre y cuando no se utilice estilo en línea).

Una de las **limitaciones** del uso de CCS es la incompatibilidad entre diferentes navegadores, por lo que si algo se ve excelente en un navegador es posible que en otro no se vea igual, o que varíe la visualización hasta en las diferentes versiones de un mismo navegador.

²XHTML: por sus siglas en inglés eXtensible Hypertext Markup Language traducido al español como Lenguaje Extensible de Marcado de Hipertexto

³CSS: por sus siglas en inglés Cascade Style Sheets traducido al español como Hojas de Estilo en Cascada

1.3.3.3. AJAX (Asynchronous JavaScript and XML)

No es una tecnología en sí mismo. En realidad, se trata de la unión de varias tecnologías que se desarrollan de forma autónoma y que se unen de forma sorprendente. Las tecnologías que forman AJAX⁴ son:

- ✓ XHTML y CSS, para crear una presentación basada en estándares.
- ✓ DOM, para la interacción y manipulación dinámica de la presentación.
- ✓ XML, XSLT y JSON, para el intercambio y manipulación de información.
- ✓ XML Http Request, para el intercambio asíncrono de información.
- ✓ Java Script, para unir todas las demás tecnologías.

AJAX se ha desarrollado para mejorar la interacción entre el usuario y la aplicación, evitando las constantes recargas de la página, debido a que el intercambio de información con el servidor se produce en un segundo plano. Este segundo plano es la creación de un elemento intermedio, una nueva capa que mejora la respuesta de la aplicación, y que no permite que el usuario encuentre una ventana del navegador en blanco, esperando la respuesta del servidor.

Demasiado código AJAX hace lento el navegador, más uso de código Java Script, trae por consiguiente mayor trabajo del navegador y el rendimiento de la Unidad Central de Procesamiento (CPU⁵) puede verse afectado.

1.3.4. Tecnologías del servidor

Las tecnologías por parte del servidor tienen como elemento común: el control. El diseñador de un sistema posee control total sobre el servidor, por tanto, se puede detallar organizadamente cómo ejecutar un programa del lado del servidor, pues estos programas son invisibles a los cambios que puedan surgir en el lado del cliente, por lo que una aplicación que funciona interactuando con el servidor puede producir páginas a cualquier navegador.

La principal desventaja es la velocidad de ejecución debido a que toda la interacción está relacionada con el servidor, lo que provoca demoras con respecto al tiempo de respuesta o al tiempo de recorrido por la red.

⁴**AJAX**: por sus siglas en inglés **A**synchronous **J**ava **S**cript **A**nd **X**ML traducido al español como Java Script asíncrono y XML

⁵**CPU** por sus siglas en inglés **C**entral **P**rocessing **U**nit traducido al español como Unidad Central de Procesamiento

1.3.4.1. JSP (Java Server Pages)

JSP⁶ es la forma más común de desarrollar páginas dinámicas en Java. Las JSP son archivos HTML o XML⁷ a los cuales se insertan etiquetas con código Java. Cuando un cliente pide una página JSP del servidor y no se ha ejecutado antes, la página es inicialmente pasada al motor de JSP, el cual compila la página convirtiéndola en servlet⁸, la ejecuta y devuelve el contenido de los resultados al cliente. (García, 2002)

La tecnología JSP puede ser analizada desde dos puntos de vista: el diseñador Web y el programador de interfaz.

El uso de páginas JSP presenta algunas **desventajas**, como la codificación de las interfaces lleva muchas líneas de código y deben hacerse usando scripts, además si sólo se emplean páginas JSP, el código de script que realiza la lógica de negocio puede llegar a ser demasiado grande y confuso, haciendo difícil su depuración.

1.3.4.2. PHP (Personal Home Page)

Es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la Web. Dentro de una página Web se puede incrustar código PHP⁹ que se va a ejecutar cada vez que se visite la página. El código PHP es interpretado en el servidor Web y genera código HTML conjuntamente con otro contenido. Este es un producto de tipo open source (código abierto) por lo que se puede acceder a su código, utilizarlo, modificarlo y sin coste alguno.

La comunidad de desarrolladores y programadores de este lenguaje de programación es muy amplia por lo que continuamente aparecen mejoras en su código de lo cual se deriva la popularidad que hoy día tiene. Es un lenguaje que está disponible en gran cantidad de sistemas operativos diferentes, desde todos los sistemas operativos de Unix tanto gratuitos como comerciales, hasta las diferentes versiones de Windows.

⁶ **JSP**: por sus siglas en inglés **J**ava **S**erver **P**ages traducido al español como Páginas Java del lado del Servidor

⁷ **XML**: por sus siglas en inglés **E**xtensible **M**arkup **L**anguage al español como Lenguaje de Marcas Extensible

⁸ **Servlet**: programa que se ejecuta en un servidor y que genera páginas Web de forma dinámica a partir de los parámetros de la petición que envíe el navegador Web

⁹ **PHP**: por sus siglas en inglés **P**ersonal **H**ome **P**age traducido al español como Páginas Web Personales

Ventajas:

- ✓ Es un lenguaje multiplataforma.
- ✓ Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- ✓ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos o extensiones.
- ✓ Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Permite las técnicas de Programación Orientada a Objetos.
- ✓ No requiere definición de tipos de variables. (Zubyc, 2009)

1.3.5. Metodologías de desarrollo

El desarrollo software es un proceso realmente complejo, en el cual es necesario el uso de una metodología adecuada, para poder obtener como resultado clientes y desarrolladores satisfechos con el producto. Las metodologías son un proceso disciplinado sobre la evolución de los programas informáticos, con el objetivo de hacerlos más entendibles y eficientes. Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software. (Aquino Leiva, y otros, 2009)

1.3.5.1. Scrum

Esta metodología de desarrollo software es ágil como la Programación Extrema, cuyo ciclo de vida está compuesto por cinco etapas (Concepto, Especulación, Exploración, Revisión y Cierre).

Tiene carácter adaptable y orienta a las personas antes que a los procesos, controla de forma empírica y adaptable la evolución del proyecto con las siguientes **prácticas de la gestión ágil**:

- ✓ Revisión de las iteraciones, desarrollo incremental (en el proyecto no se trabaja con diseño o abstracciones).
- ✓ Desarrollo evolutivo (intenta predecir en las fases iniciales como será el resultado final y sobre dicha predicción realizar el diseño).
- ✓ La estructura del producto no es realista, porque las circunstancias obligarán a remodelarlo muchas veces.

- ✓ Auto-organización (los equipos son auto-organizados, con margen de decisión suficiente para tomar las decisiones que consideren oportunas).
- ✓ Colaboración entre todos según su conocimiento y no según su rol o puesto.

El resultado final en esta metodología se consigue de forma iterativa e incremental. Al comienzo de cada iteración (sprint) se determina que partes se van a construir, tomando como criterios la prioridad para el negocio, y la cantidad de trabajo que se podrá abordar durante la iteración. Dichas iteraciones se presentan en las etapas de Especulación, Exploración y Revisión; y debido a que, según este tipo de modelos de desarrollo nunca se termina un producto, se presentan de forma infinita pudiendo llegar a la etapa de cierre solo cuando se desee enviar al mercado una versión funcional del producto. (Rodríguez González, 2008)

1.3.5.2. RUP (Rational Unified Process)

El Proceso Racional Unificado es un proceso de desarrollo de software que junto con el Lenguaje Unificado de Modelado UML¹⁰, constituyen la metodología estándar más utilizada para el análisis, la implementación y la documentación de sistemas orientados a objetos.

El ciclo de vida de RUP se caracteriza por ser:

Iterativo e incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración. Aunque cada iteración tiene que proponerse un incremento en el proceso de desarrollo, todas deben aportar al principal resultado de la fase en la que se desarrolla.

Centrado en la arquitectura: El Proceso Unificado asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema.

Guiado por los casos de uso: Se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc. (Presman, 2002)

¹⁰UML: por sus siglas en inglés **U**nified **M**odeling **L**anguage traducido al español como Lenguaje Unificado de Modelado

La vida de un sistema transcurre a través de ciclos de desarrollo, desde su nacimiento hasta su muerte, en cada ciclo se repite el proceso unificado de desarrollo. Cada ciclo consta de cuatro fases:

Inicio (Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema).

Elaboración (Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen).

Construcción (Se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario).

Transición (Se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados. Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes).

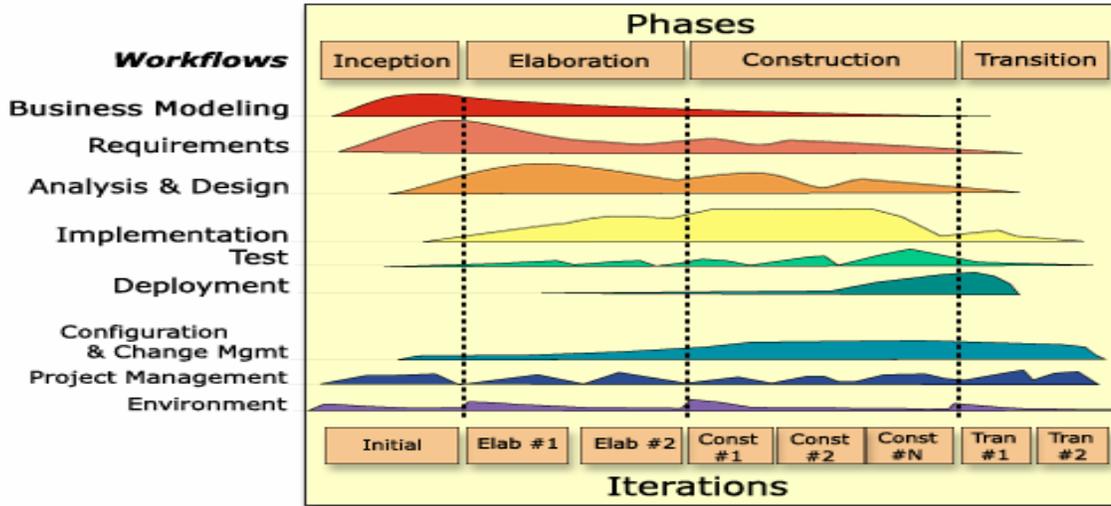


Ilustración 1: Diagrama de los flujos de trabajo a lo largo del proyecto

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

1.3.6. Lenguajes de modelado de software

Es un conjunto estandarizado de símbolos para modelar parte de un diseño de software orientado a objetos. Generalmente se usa extensivamente en combinación con una metodología de desarrollo de software para avanzar de una especificación inicial a un plan de implementación y para comunicar dicho plan a todo un equipo de desarrolladores. El uso de un lenguaje de modelado es más sencillo que la auténtica programación, pues existen menos medios para verificar efectivamente el funcionamiento adecuado del modelo.

1.3.6.1. BPMN (Business Process Modeling Notation)

BPMN¹¹ es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo.

Proporciona la capacidad de entender y definir dichos procesos, ya sean internos o externos, a través de un diagrama de procesos de negocio. Se diseñó con el objetivo de facilitar la comprensión por parte de todos los implicados (expertos TIC, analistas de negocio, directivos, etc.) que participan en el proceso.

BPMN sólo tiene tres tipos de artefactos:

- ✓ Data Object: los objetos de datos son un mecanismo para mostrar como los datos son requeridos o producidos por las actividades. Están conectados a las actividades a través de asociaciones.
- ✓ Group: un grupo es representado por un rectángulo redondeado con línea discontinua. El agrupamiento se puede usar documentación o análisis, pero no afecta al flujo de secuencia.
- ✓ Annotation: las anotaciones son mecanismos para que un modelador pueda dar información textual adicional. (Barriento, 2008)

BPMN está planeada para dar soporte únicamente a aquellos procesos que sean aplicables a procesos de negocios. Esto significa que cualquier otro tipo de modelado realizado por una organización con fines distintos de los del negocio no estará en el ámbito de BPMN. Por ejemplo, los siguientes tipos de modelado no estarían en el ámbito de BPMN:

¹¹**BPMN**: por sus siglas en inglés **B**usiness **P**rocess **M**odeling Notation traducido al español como Notación para el Modelado de Procesos de Negocio

- ✓ Estructuras organizacionales
- ✓ Descomposición funcional
- ✓ Modelos de datos

1.3.6.2. UML (Unified Modeling Language)

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software, desde una perspectiva Orientado a (OO).

El UML está basado en el uso extensivo del método Booch, OMT y Jacobson; es decir, el UML es la evolución de éstas y otras aproximaciones para modelado de procesos de negocios, objetos, y el modelado de componentes. (Presman, 2002)

El Lenguaje Unificado de Modelado establece un conjunto de notaciones y diagramas estándar para la modelación de los sistemas orientados a objetos, y describe la semántica de lo que significan. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. Las características más generales de UML son:

- ✓ Tecnología de orientación a objetos.
- ✓ Viabilidad en la corrección de errores.
- ✓ Desarrollo incremental e iterativo.

Sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura de hardware donde se ejecuten.

Permite establecer la serie de requerimientos y estructuras necesarias para diseñar un sistema de software previo al proceso de escribir código. Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa. Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto. (Presman, 2002)

1.3.7. Herramientas CASE (Computer Aided Software Engineering)

Las herramientas CASE¹² representan una forma que permite Modelar los Procesos de Negocios de las empresas y desarrollar los Sistemas de Información Gerenciales. Además, facilita organizar y manejar la información de un proyecto informático. Le permite a los participantes de un proyecto, que los sistemas (especialmente los complejos), se tornen más flexibles, más comprensibles y además mejorar la comunicación entre los participantes.

1.3.7.1. Rational Rose Enterprise

Rational Rose Enterprise es una herramienta CASE de diseño, unificada, orientada para el software que utiliza para la modelación el Lenguaje Unificado de Modelado.

Características de Rational Rose Enterprise

- ✓ Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- ✓ Capacidad de análisis de calidad de código.
- ✓ Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- ✓ Integración con otras herramientas de desarrollo de Rational.
- ✓ Publicación Web y generación de informes para optimizar la comunicación dentro del equipo.

Es una herramienta compatible solamente con sistemas operativos de Microsoft, se integra con varios entornos de desarrollo, sobre todo con diversas versiones del Visual Studio. (Aston, 2007)

1.3.7.2. Visual Paradigm

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste; brindando ayuda a arquitectos, analistas, diseñadores y

¹²**CASE**: por sus siglas en inglés **C**omputer **A**ided **S**oftware **E**ngineering traducido al español como Ingeniería de Software Asistida por Ordenador

desarrolladores. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (Presman, 2002)

Incluye herramientas muy interesantes para ingeniería inversa de bases de datos, además es una herramienta colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto y brinda la posibilidad de integrarse con los principales entornos de desarrollo y en múltiples plataformas.

Presenta varios **beneficios** entre los que se destacan:

- ✓ Poderosa herramienta de generación de PDF/HTML a partir de diagramas UML.
- ✓ Sincronización entre el código fuente y el modelo en tiempo real.
- ✓ Soporte para toda la notación UML.

1.3.8. Sistemas Gestores de Base de Datos

Un Sistema de Gestión de Bases de Datos (SGBD) consiste en un conjunto de programas, procedimientos y lenguajes que proporcionan las herramientas necesarias para trabajar con una base de datos. Incorpora una serie de funciones que permiten definir los registros, sus campos, sus relaciones, además de insertar, suprimir, modificar y consultar los datos.

1.3.8.1. MySQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario. MySQL Database Server es la base de datos de código fuente abierto más usada del mundo. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación mínima para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue. (Valdés Palacios, y otros, 2009)

Características:

- ✓ Acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones.
- ✓ Seguridad, en forma de permisos y privilegios, determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas.
- ✓ Potencia: Lenguaje muy potente para consulta de bases de datos, usar un motor ahorra una enorme cantidad de trabajo.

- ✓ Portabilidad: Es también un lenguaje estandarizado, de modo que las consultas hechas usando MySQL son fácilmente portables a otros sistemas y plataformas.
- ✓ Conectividad: Permite conexiones entre diferentes máquinas con distintos sistemas operativos.
- ✓ Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema. Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- ✓ Tiene una probabilidad muy reducida de corromper los datos.

MySQL presenta algunas **desventajas**, tales como: que un gran porcentaje de las utilidades de MySQL no están documentadas.

1.3.8.2. Postgre SQL

Actualmente es uno de los gestores de base de datos de código abierto más potente, soporta consultas complejas, integridad referencial, fue diseñado y creado para tener un mantenimiento y un ajuste menor que el de otros productos. Cuenta también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, Perl, Tcl y Python).

Postgre SQL es multiplataforma, extensible, diseñado además para ambientes de alto volumen de información con satisfactorios resultados, ofrece varios modos de bloqueos para controlar el acceso concurrente a los datos, múltiples tipos de datos redefinidos, y sobre todo lo más importante, ahorros considerables en costos de operación. (Garcia Calderón, y otros, 2009)

Postgre SQL ofrece muchas **ventajas** respecto a otros sistemas gestores de bases de datos:

- ✓ Instalación ilimitada
- ✓ Soporte
- ✓ Ahorros considerables en costos de operaciones
- ✓ Estabilidad y confidencialidad legendarias
- ✓ Extensible
- ✓ Multiplataforma
- ✓ Diseñado para ambientes de alto volumen
- ✓ Herramientas gráficas gestión de BD.de diseño y administración de BD

1.3.9. Entornos de desarrollo

Un entorno de desarrollo integrado, es un programa compuesto por un grupo de herramientas para un programador. Un IDE¹³ es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica de usuario (GUI¹⁴). Los IDE pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

1.3.9.1. ZendStudio

Es un editor PHP dirigido a la elaboración de Aplicaciones Web en este lenguaje, muy utilizado por la comunidad del PHP a nivel mundial, ofrece depuración de código, además soporta lenguajes utilizados en la web como son Java Script, CSS y HTML. Es un editor multiplataforma elaborado en software libre, el editor ofrece herramientas de programación y ayuda contextual para las sentencias del lenguaje.

El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

Mediante herramientas de edición, análisis, optimización y bases de datos, incrementa la velocidad de los ciclos de desarrollo y facilita el desarrollo de proyectos de gran complejidad.

ZendStudio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda, permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración. (Alvarez, 2003)

1.3.9.2. NetBeans

Es un entorno de desarrollo integrado de código abierto, de gran éxito con una gran base de usuarios y de distribución gratuita que proporciona herramientas muy cómodas y de fácil uso para el desarrollo de aplicaciones sobre la plataforma Java. Es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Entre las características que

¹³IDE: por sus siglas en inglés Integrated Development Environment traducido al español como Entorno de Desarrollo Integrado

¹⁴GUI: por sus siglas en inglés Graphical User Interface traducido al español como Interfaz Gráfica de Usuario

proporciona este entorno está el desarrollo de aplicaciones multiplataforma sobre: Mac OS, Windows, Linux. Incluye además:

- ✓ Soporte para Java, C, C++, XML7 y lenguajes HTML
- ✓ Incluye CVS (control de versiones) y Ant (compilación avanzada)
- ✓ Posibilidad de utilizar otras versiones de compiladores y depuradores
- ✓ Creación visual de componentes gráficos
- ✓ Dispone de todo un entorno para crear documentación javadoc

Permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las API¹⁵ de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Las aplicaciones basadas en esta plataforma pueden ser extendidas fácilmente por otros desarrolladores de software. (Alberto Ruiz, 2009)

NetBeans permite crear aplicaciones Web con PHP, además tiene soporte para la tecnología cliente AJAX.

1.3.10. Framework

En general, el término framework, se refiere a una estructura de software compuesta por componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se puede añadir piezas para construir una aplicación concreta.

Un framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo clases en java, descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. (Cárdenas Medina, y otros, 2009)

1.3.10.1. Symfony

Symfony es un framework diseñado para optimizar el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una

¹⁵API: por sus siglas en inglés **A**pplication **P**rogramming **I**nterface traducido al español como Interfaz de Programación de Aplicaciones

aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

A continuación se muestran algunas de sus características. Symfony se diseñó para que se ajustara a los siguientes requisitos:

- ✓ Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y Unix)
- ✓ Independiente del sistema gestor de bases de datos
- ✓ Sigue la mayoría de las *mejores prácticas* y patrones de diseño para la web
- ✓ Código fácil de leer que incluye comentarios de php Documentor y que permite un mantenimiento muy sencillo
- ✓ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros

La parte que más afecta al rendimiento de Symfony es Propel, el sistema ORM¹⁶ utilizado por defecto en Symfony. (Cárdenas Medina, y otros, 2009)

1.3.10.2. Codelgniter

Codelgniter es un conjunto de herramientas para personas que construyen su aplicación web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría si lo escribiese desde cero, proveyéndole un rico juego de librerías para tareas comúnmente necesarias, así como una interface simple y estructura lógica para acceder a esas librerías.

Codelgniter le permite creativamente enfocarse en su proyecto minimizando la cantidad de código necesaria para una tarea dada.

Codelgniter gestiona la seguridad de diferentes formas. Es justamente restrictivo sobre que caracteres permitir en las cadenas URI para ayudar a minimizar la posibilidad de que datos maliciosos puedan ser pasados a su aplicación.

Una de las características más interesantes de Codelgniter es el elevado número de clases que incluye para trabajar con distintos objetos: calendario, bases de datos, correo electrónico, manipulación de imágenes, FTP, lenguaje, tablas, sesiones, compresión ZIP, entre otros. A diferencia de otros frameworks, Codelgniter cuenta con una documentación excelente que permite conocer todos los secretos de este entorno de trabajo. Está liberado bajo la licencia de

¹⁶ORM: por sus siglas en inglés **O**bject **R**elational **M**apping traducido al español como Mapeo Objeto Relacional

código abierto Apache-BSD, lo que significa que es totalmente libre y puede ser usado. (Rivera, 2007)

Facilidades:

- ✓ Se encuentra bajo una licencia open source Apache/BSD-style.
- ✓ Verdaderamente Liviano. El núcleo del sistema sólo requiere unas pocas librerías. Realmente rápido.
- ✓ Contiene librerías que le permiten realizar las tareas de desarrollo web más comúnmente necesarias, como acceder a una base de datos, mandar un email, validar datos de un formulario, mantener sesiones, manipular imágenes, entre otras.
- ✓ Es extensible. El sistema puede ser fácilmente extendido a través del uso de plugins y librerías asistentes, o a través de extensión de clases o ganchos del sistema.
- ✓ Posee una amplia y completa documentación.
- ✓ Baja curva de aprendizaje para empezar a usarlo.

1.3.11. Fundamentación de la selección de las tecnologías, metodología y herramientas de desarrollo.

Después de realizar un estudio de diferentes tecnologías, metodologías y herramientas de desarrollo de software se ha decidido implementar una aplicación WEB para dar solución a la problemática planteada. Una aplicación web proporciona un rápido acceso a la información de forma actualizada e inmediata, sin necesidad de instalación en las máquinas clientes de algún software, son muy fáciles de utilizar y puede haber varios usuarios conectados a la aplicación concurrentemente.

Se ha decidido utilizar para el desarrollo del Sistema de Registro Electrónico Directo para las votaciones en la UCI tecnologías cliente/servidor sobre plataforma web. Como tecnologías cliente se seleccionó Java Script, CSS y AJAX y como tecnología servidor PHP, debido a que es un producto de Código Abierto, multiplataforma y tiene conexión con el potente SGBD Postgre SQL, que es el seleccionado para gestionar la BD del sistema por ser de Código Abierto, multiplataforma y permite manipular grandes volúmenes de información.

NetBeans fue seleccionado como IDE para el desarrollo del sistema por todas las ventajas que ofrece, proporciona herramientas muy útiles para los programadores, agiliza el ciclo de desarrollo y simplifica la complejidad del proyecto.

Teniendo en cuenta los argumentos antes expuestos y dado que se va a hacer uso de tecnologías cliente/servidor sobre plataforma web, luego de haberse realizado un análisis de los principales frameworks de PHP más utilizados en la actualidad, se llega a la conclusión de que el más apropiado para el desarrollo del sistema es el CodeIgniter, debido a su gran integración con el gestor de base de datos que se usará (Postgre SQL), además es un poderoso Framework para PHP que facilita la escritura de código repetitivo, y en comparación de otros Frameworks es más rápido y más fácil, es totalmente extensible y altamente compatible con gran variedad de versiones y configuraciones de PHP. Ofrece un alto rendimiento, es ligero y fácil de instalar. Viene con varias librerías para gestionar el acceso a datos, sesiones de usuarios, formularios y la seguridad. Utiliza el patrón arquitectónico Modelo-Vista-Controlador (MVC) como paradigma de desarrollo. Sin dudas NetBeans junto a CodeIgniter es la propuesta ideal para el desarrollo de la aplicación web.

Por disímiles razones se escoge la metodología RUP para la realización de este trabajo, por su nivel al describir los procesos y crear diagramas UML. Es muy conveniente porque durante su ciclo de vida se desarrolla toda una metodología iterativa e incremental que va eliminando los errores cometidos en las iteraciones previas, logrando que al final del proceso se obtenga como resultado un producto de calidad. Además, esta robusta metodología constituye un estándar probado a nivel mundial, hace énfasis en realizar una buena elicitación de requisitos y por ende un fructuoso diseño.

Luego de estudiar los diferentes lenguajes de modelado se seleccionaron BPMN para modelar los procesos de negocio y UML para el resto de los procesos.

BPMN es elegido debido a la complejidad que presentan los procesos de negocio del proceso eleccionario, ya que este lenguaje considera un único diagrama para representar los procesos de negocio, que está basado en la técnica de diagramas de flujo y adaptado para graficar las operaciones de los procesos de la organización. No utilizando UML para modelar estos procesos debido a que su enfoque orientado a objetos contradice el enfoque orientado al negocio, además dificulta la comunicación con el cliente porque está pensado para un público predominantemente técnico, a diferencia de BPMN que está compuesto por un conjunto de elementos gráficos que facilitan que tanto los miembros del equipo de desarrollo como el cliente entiendan con claridad sus diagramas.

UML por su parte está más orientado a diseñadores de software. Es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. El mismo está definido como

estándar en el diseño de sistemas. Los diseños realizados se pueden implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos).

Como herramienta de modelado se seleccionó Visual Paradigm, ya que constituye una de las herramientas más potentes que existen en la actualidad. La misma soporta UML y BPMN, que son los lenguajes de modelado seleccionados. Es multiplataforma, soporta el ciclo de vida completo del desarrollo de software, facilitando la tarea de dibujar diagramas, así como su corrección sintáctica. Permite dibujar todos los tipos de diagramas de clases y genera documentación partir de los mismos.

1.4. Conclusiones parciales

Después de analizar los principales conceptos asociados a los Sistemas Electorales se profundizó en el estudio de los Sistema de Registro Electrónico Directo como tipo de Sistema de Votación Digital más adecuado a utilizar para las votaciones en la UCI. Para el desarrollo de esta aplicación se fundamentó la selección de las tecnologías, metodologías y herramientas más apropiadas a utilizar. Debido a que se está luchando en la Universidad por alcanzar la independencia tecnológica se seleccionaron tecnologías potentes en su mayoría multiplataforma y de software libre. De ahí la elección de PHP como lenguaje de programación y CodeIgniter como framework. Además, se designó como metodología RUP porque es la que más se ajusta a las necesidades del Sistema de Registro Electrónico Directo para las votaciones en la UCI. Para el modelado se utilizan BPMN y UML ya que cuentan con características específicas para cada uno de los flujos de trabajo que se modelarán. Con el objetivo de mejorar y estandarizar la documentación se escogió Visual Paradigm, herramienta que soporta todo el ciclo del desarrollo de un proyecto.

Capítulo 2

2.1. Introducción del capítulo

Para comprender cómo se realizan las votaciones en la UCI se hace necesario estudiar el desarrollo del proceso electoral en cada organización para la cual se va a desarrollar la aplicación. Este estudio permitirá modelar los procesos de negocios, describir los artefactos generados durante este flujo de trabajo, especificar los requerimientos, identificando los actores, trabajadores y casos de usos.

2.2. Objeto de automatización

Actualmente la UCI no cuenta con un sistema que automatice los procesos electorales, capaz de gestionar la información relacionada con las votaciones que se desarrollan en la misma. Debido a la necesidad de mejorar el proceso electoral, se decide desarrollar un Sistema de Registro Electrónico Directo para las votaciones.

El desarrollo de este sistema propone automatizar las actividades que involucra la gestión de la información de las votaciones, incluyendo el proceso de votación, tales como:

Recoger datos de Candidato: El Organizador (Facultad) del proceso electoral recoge los datos (nombre, apellido, grupo, currículum) de los candidatos seleccionados.

Listar Candidato: El Organizador (Facultad) del proceso electoral confecciona un listado de los candidatos seleccionados con sus datos y lo divulga para que los electores (los miembros de la organización) conozcan a sus candidatos.

Enviar listado de Candidatos: El Organizador (Facultad) del proceso electoral envía el listado de sus candidatos con sus datos correspondientes al Organizador (UCI) del proceso electoral.

Enviar cantidad de electores por facultad: El Organizador (Facultad) del proceso electoral envía un listado con la cantidad de electores por facultad al Organizador (UCI) del proceso electoral.

Listar Colegio: El Organizador (UCI) del proceso electoral confecciona un listado con los nombres de todos los colegios y la cantidad de electores. Los colegios son conformados por

facultades, en dependencia de la cantidad de personas vinculadas a la misma, son ubicadas una o más facultades en un mismo colegio electoral.

Adicionar Urna: El Organizador (UCI) del proceso eleccionario asigna las urnas a los colegios correspondientes.

Efectuar voto: Es el proceso en el que los electores acuden al colegio electoral correspondiente a su facultad y emiten su voto, siempre que el Responsable del Colegio Electoral afirme que ese elector está en el padrón electoral correspondiente a ese colegio.

Emitir resultados: El Responsable del Colegio Electoral cuenta los votos y comunica los resultados.

2.3. Información que se manipula

Padrón Electoral: Es el listado actualizado de todos los electores habilitados para efectuar las votaciones en un determinado colegio electoral. Es utilizado para verificar que coincidan los datos del Carnet de Identidad del elector con los registrados en el padrón, además en él se registra la firma del elector como constancia de su voto.

Currículo del candidato: Es el documento donde se encuentra la biografía de cada candidato, lo que permite al elector conocer el historial de los candidatos.

Boleta electoral: Es un listado con el nombre y los apellidos de los candidatos del proceso electoral.

Resultados de las votaciones: Son los datos finales que se obtienen luego de realizar el escrutinio de las votaciones.

2.4. Propuesta de sistema

Para darle solución a la situación problemática se ha decidido diseñar un Sistema de Registro Electrónico Directo para las votaciones en la Universidad de las Ciencias Informáticas de forma que optimice la utilización de los recursos materiales y humanos.

El sistema contará con tres módulos. El módulo de **Administración** permitirá Gestionar Usuario, Gestionar Roles, y Gestionar Funcionalidades, el de **Proceso Eleccionario** es el encargado de Gestionar Candidato, Gestionar Colegio, Gestionar Urna y Configurar Boleta, también permitirá consultar el currículo de los candidatos y efectuar el voto. En el módulo de **Reportes** se podrán

obtener los reportes en tiempo real de los datos almacenados en el sistema, tales como: Resultados de las votaciones por candidato, Registro de comparecencia, Resultados generales por urna y Resultados generales por colegio.

2.5. Modelo de negocio

La disciplina de modelado de negocio es la primera que propone RUP dentro del ciclo de desarrollo de un software, tiene su mayor peso durante la fase de inicio debido a que permite conocer los procesos existentes actuales de cualquier entidad o empresa para la que se vaya a desarrollar el sistema. Es en este flujo de trabajo donde se conocen a fondo cómo son iniciadas cada una de las actividades de un sector determinado, y a través del modelado de estos procesos se obtiene una visión más amplia del negocio existente. (Presman, 2002)

2.5.1. Descripción del negocio basado en procesos

Para conseguir sus objetivos, una empresa organiza su actividad por medio de un conjunto de procesos de negocio. Cada uno de ellos se caracteriza por una colección de datos que son producidos y manipulados mediante un conjunto de tareas, en las que ciertos agentes (por ejemplo, trabajadores o departamentos) participan de acuerdo con un flujo de trabajo determinado. Además, estos procesos se hallan sujetos a un conjunto de reglas de negocio, que determinan las políticas y la estructura de la información de la empresa. Por tanto, la finalidad del modelado del negocio es describir cada proceso, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas de negocio. El primer paso del modelado del negocio consiste en capturar los procesos de negocio de la organización bajo estudio. La definición del conjunto de procesos del negocio es una tarea crucial, ya que define los límites del proceso de modelado posterior. (Barriento, 2008)

2.5.1.1. Proceso de negocio: Organizar Proceso Eleccionario

Este proceso comienza cuando el Presidente de la comisión electoral de la facultad recibe el nombre, apellidos, grupo y currículum de los candidatos seleccionados, para confeccionar un listado con estos datos. Luego se le envía este listado de candidatos al Presidente de la comisión electoral a nivel UCI para que verifique que no existan errores en la información recibida. Una vez que no exista ningún error se procede a crear la boleta y se adiciona las urnas necesarias por

colegios electorales, sino se le manda al Presidente de la comisión electoral de la facultad el listado para que sea actualizado correctamente.

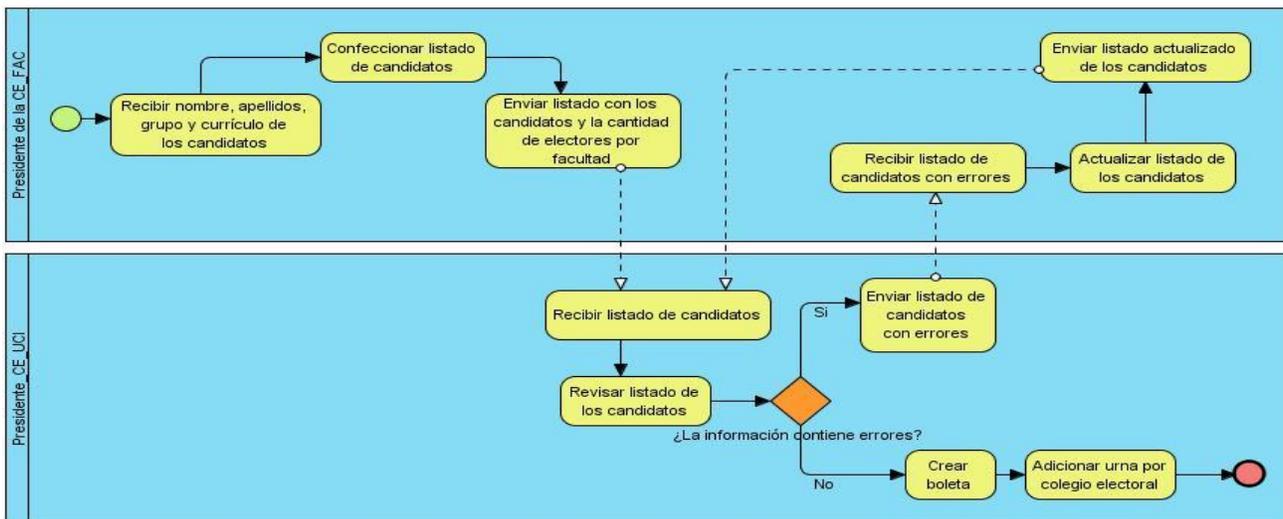


Ilustración 2: Diagrama del proceso de negocio Organizar Proceso Eleccionario

2.5.1.2. Proceso de negocio: Efectuar Voto

Este proceso se inicia cuando el Responsable del Colegio Electoral pide el carnet de identidad al elector. Éste se lo entrega para que verifique el nombre, los apellidos y el número del carnet de identidad del elector con los almacenados en el padrón electoral. Si coinciden los datos, el responsable del colegio electoral indica al elector para que firme el padrón electoral y procede a efectuar el voto, en caso de que los datos no sean correctos le notifica que no podrá efectuar el voto en ese colegio electoral.

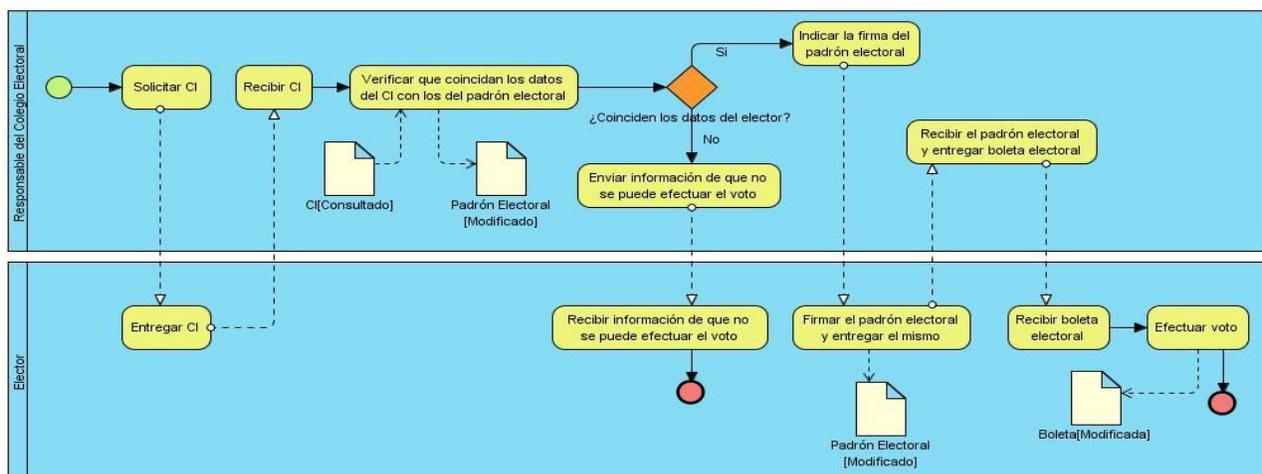


Ilustración 3: Diagrama del proceso de negocio Efectuar Voto

2.5.1.3. Proceso de negocio: Emitir Resultados

Este proceso comienza cuando el responsable del colegio electoral verifica que cada uno de los votos emitidos por los electores son válidos para su posterior escrutinio, en caso contrario se encarga de anular las boletas electorales que contengan errores. Luego el responsable del colegio electoral confecciona un informe con los resultados de las votaciones y se les comunica a los electores.

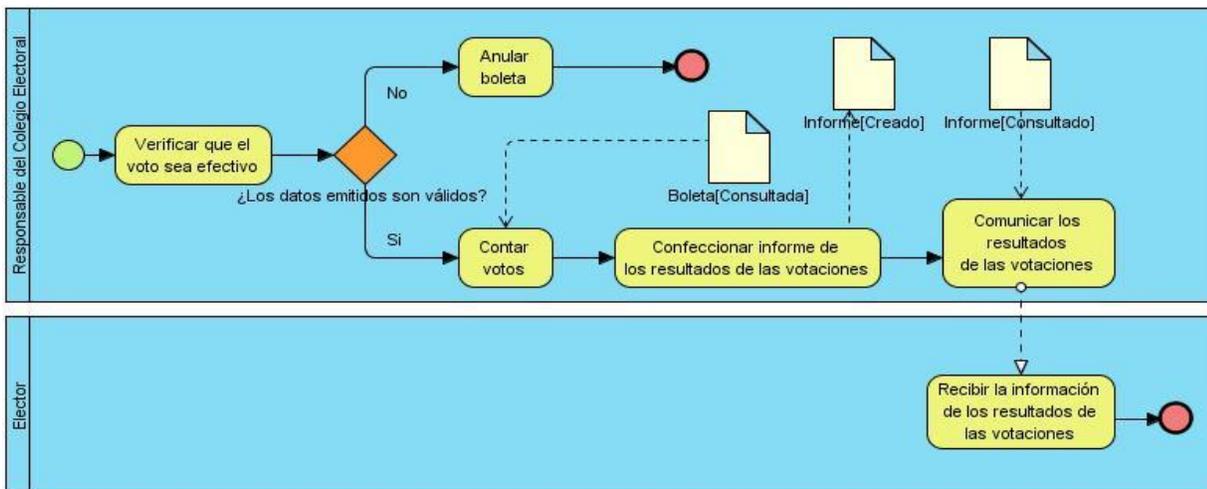


Ilustración 4: Diagrama del proceso de negocio Emitir Resultados

En el artefacto Modelo de Negocio se encuentra una descripción más detallada de los procesos de negocio.

2.6. Especificación de requisitos

Un **requerimiento** es una representación documentada de una condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Se pueden encontrar dos tipos de requerimientos: funcionales y no funcionales. (Presman, 2002)

2.6.1. Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir sin alterar la funcionalidad del producto, esto quiere decir que se mantienen invariables sin importar con qué propiedades o cualidades se relacionan. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema. (Presman, 2002)

A partir de los procesos de negocio estudiados y las actividades a automatizar identificadas en cada uno de los módulos, se definen 36 requerimientos funcionales para la realización del Sistema de Registro Electrónico Directo para las votaciones en la UCI.

RF1: Autenticar

RF2: Buscar persona

RF3: Registrar usuario

RF4: Listar usuario

RF5: Eliminar usuario

RF6: Adicionar roles

RF7: Listar roles

RF8: Modificar rol

RF9: Eliminar rol

RF10: Registrar funcionalidad

RF11: Listar funcionalidades 4

RF12: Modificar funcionalidad

RF13: Eliminar funcionalidad4

RF14: Registrar candidato

RF15: Listar candidato

RF16: Eliminar candidato

RF17: Mostrar currículum

RF18: Eliminar currículum

RF19: Registrar datos de colegio electoral

RF20: Listar colegio electoral

RF21: Actualizar datos de colegio electoral

RF22: Eliminar colegio electoral

RF23: Registrar urna

RF24: Listar urna

RF25: Actualizar urna

RF26: Eliminar urna

RF27: Listar tipos de votación

RF28: Seleccionar tipos de votación

RF29: Listar candidatos

RF30: Registrar nuevos candidatos

RF31: Registrar votos por miembros

RF32: Registrar votos por presidente

RF33: Mostrar resultados por candidatos

RF34: Mostrar comparecencia por colegio electoral

RF35: Mostrar resultados generales

RF36: Mostrar cantidad de votos por urna perteneciente a cada colegio electoral

2.6.2. Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Definiendo a las propiedades como características que hacen al producto atractivo, usable, rápido o confiable. (Presman, 2002)

A continuación se muestra un listado de los 11 requerimientos no funcionales identificados.

RNF1: Usabilidad

RNF2: Soporte

RNF3: Rendimiento

RNF4: Portabilidad

RNF5: Seguridad

RNF5.1: Confidencialidad

RNF5.2: Integridad

RNF5.3: Disponibilidad

RNF6: Apariencia o Interfaz Externa

RNF7: Apariencia o Interfaz Interna

RNF8: Ayuda y documentación en línea

RNF9: Software

RNF10: Hardware

RNF11: Restricciones en el diseño y la implementación

En el artefacto Especificación de requisitos se detallan los requerimientos funcionales y no funcionales identificados para el Sistema de Registro Electrónico Directo para las votaciones en la UCI.

2.6.3. Validación de requisitos

Métrica de calidad de especificidad de requisitos

Las métricas de calidad para la especificación de requisitos son una técnica que da como resultado una medida de lo ajustada que es una especificación de requisitos, donde se establecen magnitudes y valores. Para el presente trabajo se han realizado dos observaciones para adquirir requisitos fuertes y sin incoherencias, en las mismas se lograron los siguientes resultados a partir de los datos:

Observación 1: En la primera observación se encontraron 2 requisitos que realizaban la misma funcionalidad. Además los revisores encontraron dos requisitos que podían ser reestructurados en uno. Por lo que de 38 requisitos tuvieron ambigüedad 2 de ellos.

n_r : Número de requisitos.

n_f : Número de requisitos funcionales. (38)

n_{nf} : Número de requisitos no funcionales. (11)

$$n_r = n_f + n_{nf}$$

$$n_r = 38 + 11$$

$$n_r = 49$$

Q_1 : Consistencia de la interpretación por los revisores.

N_{u1} : Número de requisitos para la que todos los revisores tuvieron interpretaciones iguales.

$$Q_1 = \frac{N_{u1}}{n_r}$$

$$Q_1 = \frac{47}{49} = 0,96$$

El valor obtenido por Q_1 fue de 0.96, el cual es bastante cercano al valor 1 lo que comprueba que los requisitos contienen un alto grado de claridad ya que responden a una única interpretación siendo satisfactoria para el desarrollo de la aplicación.

Observación 2: En esta observación se solucionaron los problemas de ambigüedad y se reestructuraron los requisitos que fueron señalados. Después de los cambios realizados se cuenta con iguales interpretaciones por parte de los revisores para los 47 requisitos obtenidos.

n_f : Número de requisitos funcionales. (36)

n_{nf} : Número de requisitos no funcionales. (11)

$$n_r = n_f + n_{nf}$$

$$n_r = 36 + 11$$

$$n_r = 47$$

Q_1 : Consistencia de la interpretación por los revisores.

N_{u1} : Número de requisitos para la que todos los revisores tuvieron interpretaciones iguales.

$$Q_1 = \frac{N_{u1}}{n_r}$$

$$Q_1 = \frac{47}{47} = 1$$

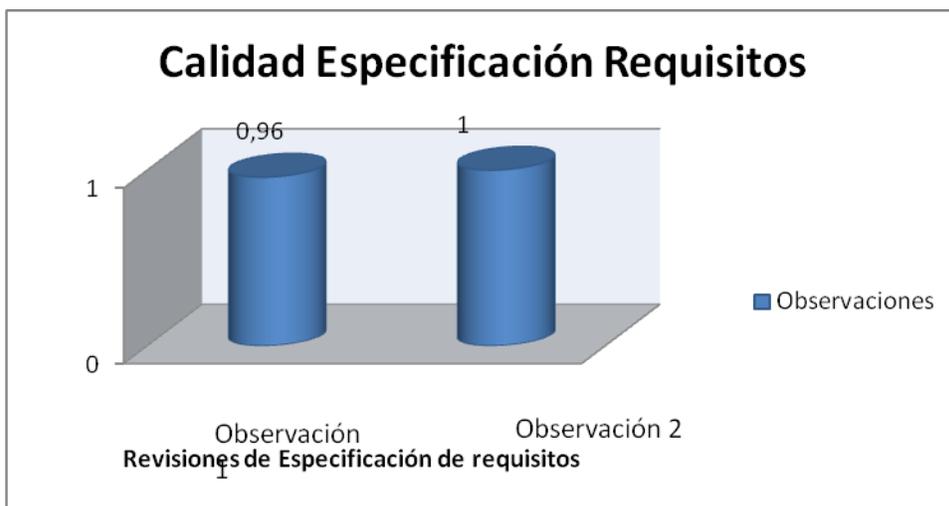


Ilustración 5: Calidad de la especificación de requisitos.

Resultado:

Después de haber aplicado esta métrica se demostró que las especificaciones de los requisitos presentan un alto grado de claridad, ya que el valor obtenido Q_1 finalmente es 1. Esto significa que los resultados son satisfactorios lo cual no atrasa el desarrollo de la aplicación.

2.7. Modelo de sistema

Como resultado del flujo de trabajo de requisitos se obtiene una vista externa del sistema. El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requerimientos. En esta vista interna se especifican mejor los casos de uso y se determinan las clases necesarias para llevar a cabo las funcionalidades en ellos contenidos, constituyendo una entrada para el análisis, el diseño y las pruebas. (Presman, 2002)

El modelo de casos de uso describe lo que hace el sistema para cada tipo de usuario. Cada uno de estos se representa mediante uno o más actores (personas).

Se han identificado 5 actores del sistema, seguidamente se muestra una figura que especifica las responsabilidades de cada uno de ellos.

Actor	Descripción
Usuario	Es el actor responsable de autenticarse para acceder a la aplicación y de ésta forma verificar su usuario y contraseña, y así definir su nivel de acceso al sistema.
Administrador	Es el actor que una vez autenticado en el sistema con el rol de administrador, tiene la responsabilidad de gestionar los roles y usuarios, además asigna a los usuarios su correspondiente rol.
Elector	Es el actor encargado de efectuar el voto y puede consultar el currículum de los candidatos, también tiene permisos para buscar y agregar otro candidato a la boleta.
Responsable del colegio electoral	Es el actor responsable de realizar el escrutinio de los votos, puede consultar los reportes de las elecciones y emitir los resultados.
Presidente de la CE_UCI	Es el actor encargado de confeccionar el listado general de los candidatos de todos los colegios, es decir, la boleta electoral, con sus respectivos datos (Nombre, Apellidos, Grupo, Currículum). Además asigna las urnas a los colegios correspondientes, gestiona los colegios electorales, elimina y modifica los currículos de los candidatos, y puede eliminar los candidatos que por alguna razón ya no van a estar en la boleta electoral.

Ilustración 6: Descripción de los actores del sistema

Las relaciones existentes entre los actores y los Casos de Usos del sistema se muestran en el diagrama de Casos de Uso del Sistema.

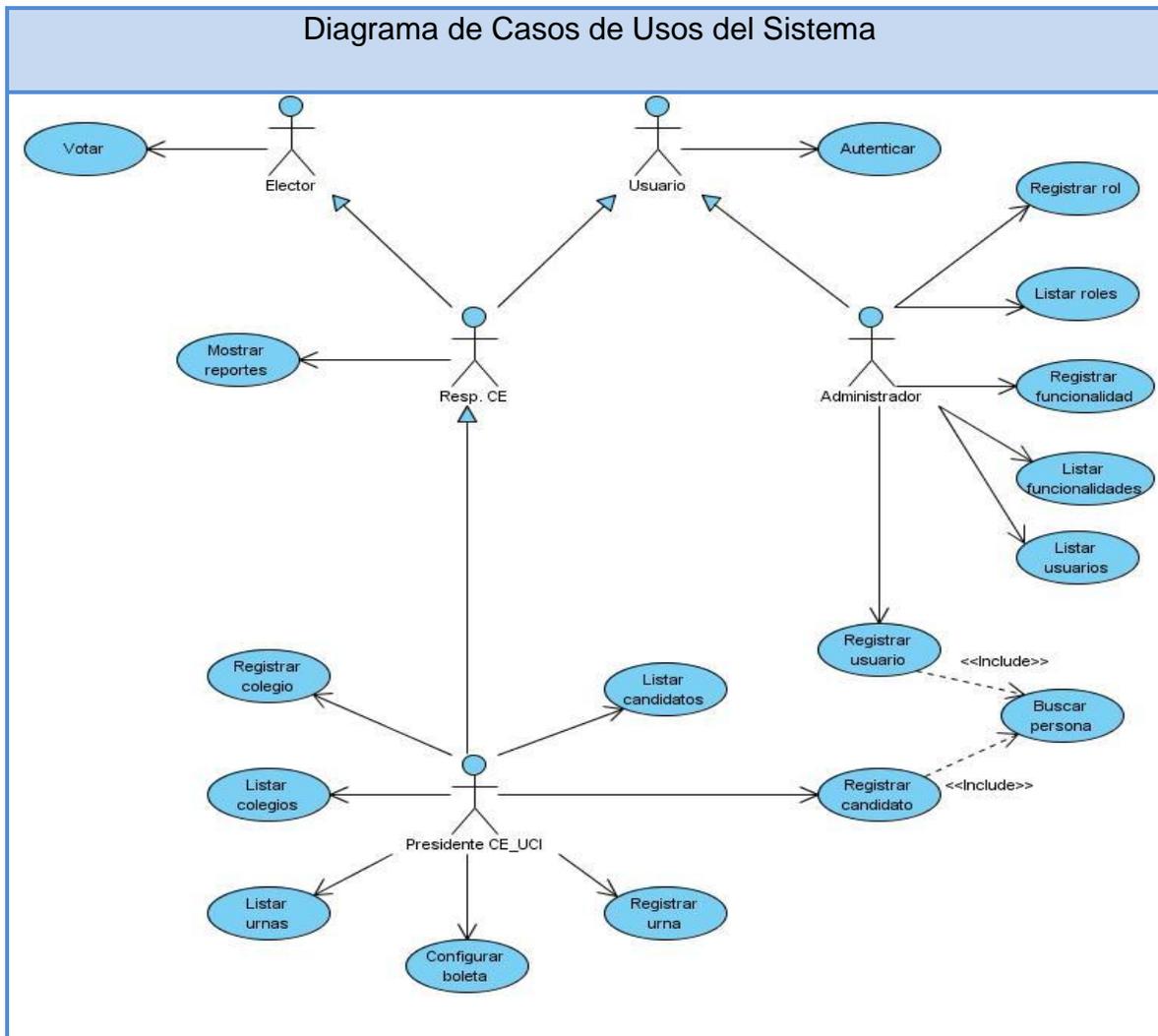


Ilustración 7: Diagrama de Casos de Usos del Sistema

En el artefacto Modelo de sistema se encuentra la descripción detallada de todos los Casos de Usos identificados para el Sistema de Registro Electrónico Directo para las votaciones en la UCI.

2.7.1. Patrones de Casos de Uso

- ✓ **Múltiples actores:** Rol común (Common Role):

Dos actores desempeñan el mismo papel en determinado caso de uso. Este rol es representado por otro actor, que contiene de forma hereditaria los actores que comparten este rol. Este patrón debe aplicarse cuando, desde el punto de vista de un caso de uso hay solo una entidad externa interactuando con cada instancia del caso de uso.

En el sistema se han identificado varios actores que interpretan un rol común: el rol usuario, surge por la necesidad de que todas las personas que necesitan acceder a los módulos de la aplicación que requieran autenticación puedan hacerlo, ya sea un Administrador, un Responsable de un Colegio Electoral o el Presidente de la Comisión Electoral a nivel UCI. También está el rol de Elector, éste no necesita autenticarse, pues solo accederá al módulo de Urna Digital para efectuar el voto, los actores Responsable de un Colegio Electoral y Presidente de la Comisión Electoral a nivel UCI se comportan en algún momento del proceso electoral como un Elector, pues éstos también están habilitados para votar.

✓ **CRUD:** Crear, Leer, Actualizar y Eliminar (Creating, Reading, Updating and Deleting):

Este patrón posee un caso de uso, llamado “Información CRUD” o “Gestionar Información”, que modela todas las operaciones que se pueden realizar sobre una parte de la información de un tipo determinado; ya sea crearla, leerla, actualizarla y eliminarla. Es importante destacar que este patrón se usa cuando todos los flujos contribuyen al mismo valor de negocio y son cortos y simples.

Este patrón es utilizado en el sistema para gestionar distintas informaciones, tales como: Gestionar usuario, Gestionar rol, Gestionar funcionalidad, Gestionar urna, Gestionar colegio y Gestionar candidato.

El artefacto Modelo de Casos de Uso del Sistema recoge la descripción de los actores del sistema, el diagrama de casos de usos y la descripción detallada de los 17 casos de uso identificados para el Sistema de Registro Electrónico Directo para las votaciones en la UCI.

2.8. Conclusiones parciales

Se propone un sistema para automatizar las votaciones en la UCI a partir del objeto de estudio, mostrándose el diagrama de casos de uso del negocio. A partir de la definición de los requerimientos funcionales y no funcionales se definen los casos de usos del sistema a tener en cuenta para el desarrollo de la aplicación. Con el desarrollo de este capítulo quedaron sentadas las bases que darán paso al diseño y construcción del Sistema de Registro Electrónico Directo para las votaciones en la UCI.

Capítulo 3

3.1. Introducción del capítulo

En la fase de Elaboración el flujo de mayor peso por la gran cantidad de actividades que se realizan es el de Análisis y Diseño. En el análisis se puede estructurar los requisitos de manera que facilite su comprensión, su preparación, su modificación y en general su mantenimiento, ya que el objetivo del análisis es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución.

Con el modelado del diseño, se adquiere una mayor comprensión de los aspectos relacionados con los requerimientos funcionales, no funcionales y de las restricciones necesarias para el sistema. Este Flujo de Trabajo contribuye a la definición de una arquitectura estable y sólida, creando un plano del modelo de implementación.

El modelo de análisis y el modelo de diseño cuentan con los mismos artefactos, clases y realizaciones de casos de uso, pero en el análisis los casos de usos están descritos en un lenguaje entendible para los clientes, por lo que sería el paso intermedio para llegar al diseño, debido a que el modelo de diseño es más detallado y está descrito en un lenguaje técnico completamente entendible para los programadores. El modelo de análisis es una abstracción al modelo del diseño.

Para el desarrollo del Sistema de Registro Electrónico Directo para las votaciones en la UCI se decide pasar directamente del levantamiento de requisitos al diseño del sistema, reconociendo las ventajas del modelo del análisis, no se hace necesario el desarrollo del mismo para esta aplicación, pues es un proyecto pequeño, los analistas, diseñadores e implementadores serán las mismas personas y no se va a integrar ningún miembro nuevo al equipo de desarrollo. Además, se tienen bien definidos y estables los requerimientos y la tecnología de programación.

El propósito del presente capítulo es definir la estructura y elementos del diseño, describir los casos de uso en términos de clases y sus objetos, representándolos gráficamente en Diagramas de Clases, así como fundamentar los patrones empleados y estructurar el Modelo de Datos.

3.2. Modelo del diseño

Un Modelo de Diseño es una abstracción del Modelo de Implementación y su código fuente, el cual fundamentalmente se emplea para representar y documentar su diseño. Es usado como entrada esencial en las actividades relacionadas a la implementación. Representa a los casos de uso en el dominio de la solución. Este Modelo puede contener: los diagramas, las clases, paquetes, subsistemas, relaciones, colaboraciones, atributos, las realizaciones de los casos de uso, entre otros que se puedan considerar para el sistema en desarrollo. (Presman, 2002)

3.2.1. Fundamentación del uso de Patrones

Un patrón es una solución recurrente para un problema en un contexto o la respuesta al problema dentro de un contexto que ayuda a resolver las dificultades. Los Patrones de Diseño son directrices y principios estructurados que describen un problema común y entregan una buena solución ya probada a la que se le da un nombre. (Larman, 2004)

La utilización de los mismos en la concepción del Sistema de Registro Electrónico Directo para las votaciones en la UCI, ayudará a diseñar correctamente en menos tiempo, a construir clases reutilizables, facilitará la documentación y conducirá a la definición de una arquitectura pequeña, simple y comprensible.

3.2.1.1. Patrones para asignación de responsabilidades

✓ Alta Cohesión

La cohesión es la medida de la fuerza que une a las responsabilidades de una clase. Una clase con baja cohesión es aquella que hace muchas cosas no afines o muchas tareas, lo que trae como consecuencias dificultades para entender, reutilizar y conservarla. Son delicadas y las afectan constantemente los cambios. Una clase con alta cohesión mejora la claridad y la facilidad de su uso, su mantenimiento se simplifica y es fácil de reutilizar. A menudo, se genera un bajo acoplamiento. (Larman, 2004)

El Sistema de Registro Electrónico Directo para las votaciones en la UCI se diseñó asignando responsabilidades de modo que la cohesión sea alta, por lo que los componentes del mismo

tienen el mínimo de tareas en el sistema, ejemplo de esto lo representan los métodos del negocio, ya que tienen bien delimitadas sus responsabilidades, no recargando en ningún caso sus funcionalidades, permitiendo mayor eficiencia y que el tiempo de respuesta y ejecución no exceda lo estimado, generando por ende bajo acoplamiento.

✓ **Bajo Acoplamiento**

El acoplamiento es la medida de la fuerza con que una clase está conectada a otras clases, a las que conoce y recurre a ellas. Una clase con bajo acoplamiento no depende de muchas otras. Una clase con alto acoplamiento recurre a muchas otras, lo que trae consigo que los cambios en una clase ocasionen cambios en otras, sean más difíciles de entender y de reutilizar porque se requiere la presencia de otras clases de las que dependen.

El diseño del Sistema de Registro Electrónico Directo para las votaciones en la UCI está regido por la creación de dependencias escasas, un ejemplo de esto lo constituye la definición de métodos del negocio poco dependientes de otros, lo que reduce el impacto del cambio y favorece la reutilización.

3.2.1.2. Patrones estructurales

✓ **Fachada (facade)**

El patrón fachada trata de simplificar la interfaz entre dos sistemas o componentes de software ocultando un sistema complejo detrás de una clase que hace de pantalla o fachada. Denominado por algunos como, patrón no converses con extraños.

Para el diseño del Sistema de Registro Electrónico Directo para las votaciones en la UCI se emplea una clase Fachada en la capa de lógica de negocio, denominada My_Controller, de la cual heredan todas las demás clases, ocultando la complejidad de sus acciones internas. Este patrón brinda, además, escalabilidad y portabilidad al sistema, reduciendo el impacto del cambio.

3.2.1.3. Patrones arquitectónicos

✓ **Modelo Vista Controlador**

La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si se realiza un diseño que mezcle los componentes de interfaz y de negocio, entonces los cambios que se realicen en las interfaces producirán modificaciones en los componentes de negocio, lo que proporcionará mayor trabajo y más riesgo de errores. (Larman, 2004)

En el diseño del Sistema de Registro Electrónico Directo para las votaciones en la UCI se separa la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma, las modificaciones en las vistas/interfaces impactarán en menor medida en la lógica de negocio o de datos, reduciendo el impacto del cambio.

Elementos del patrón:

- **Modelo:** datos y reglas de negocio
- **Vista:** muestra la información del modelo al usuario
- **Controlador:** gestiona las entradas del usuario

En la propuesta de solución:

- **Modelo:** Formado por la base de datos del Sistema de Registro Electrónico Directo para las votaciones en la UCI.
- **Vista:** Constituidas por los ficheros Java Script y clientes PHP de la capa de presentación.
- **Controlador:** Métodos del negocio asociados las votaciones como parte del proceso electoral.

3.2.2. Diagrama genérico de las clases del diseño

En el diagrama genérico de las clases del diseño que sigue a continuación se muestra la especificación para las clases de la aplicación, incluyendo la siguiente información:

- ✓ Clases, asociaciones y atributos.
- ✓ Métodos.

- ✓ Navegabilidad.
- ✓ Dependencias.

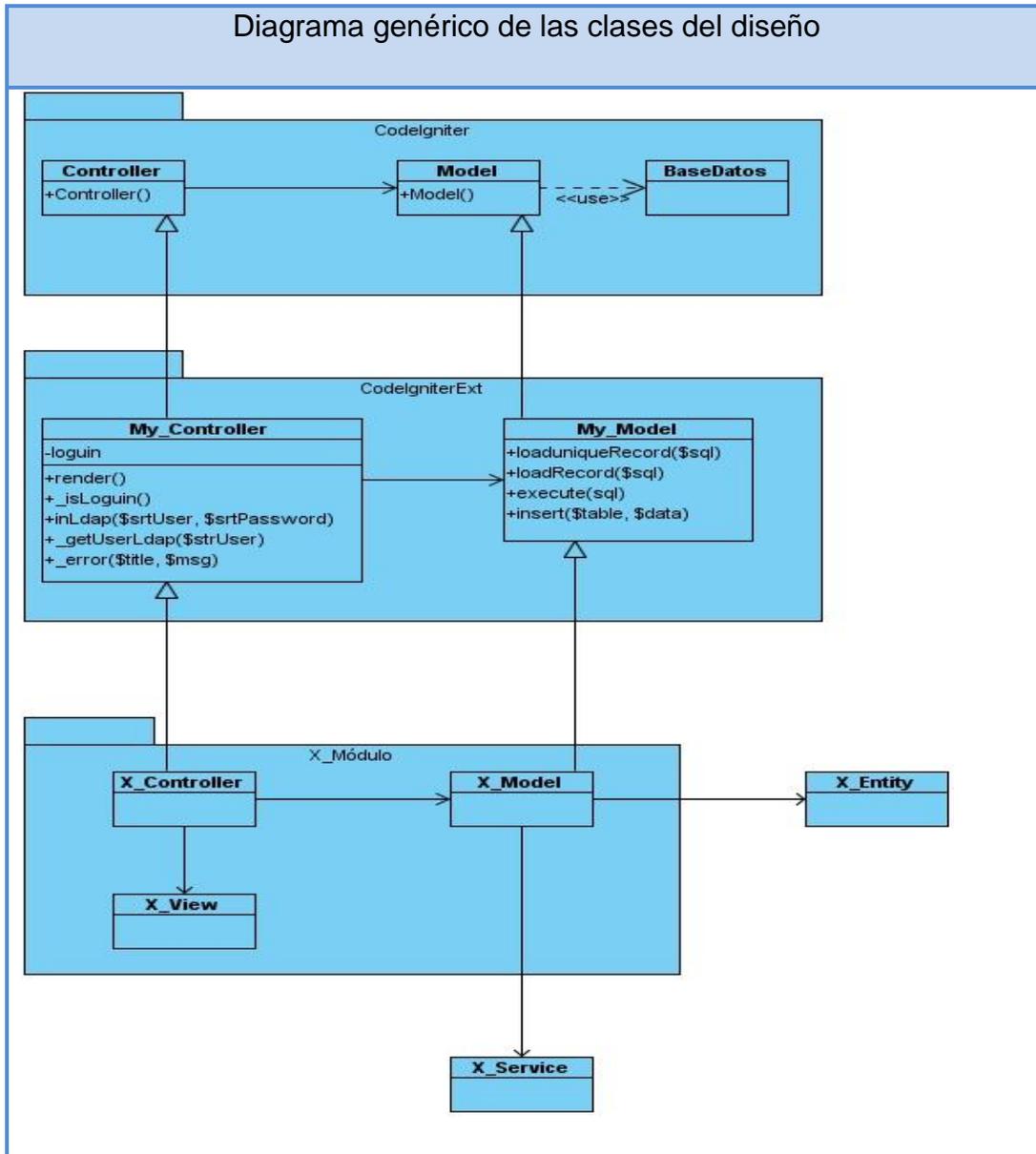


Ilustración 8: Diagrama genérico de las clases del diseño

3.2.3. Diagramas de interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. En el contexto de las clases describen la forma en que grupos de objetos colaboran para proveer un comportamiento. (PRESSMAN, 2002)

3.2.3.1. Diagramas de secuencia

El diagrama de secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase. Mientras que el diagrama de casos de uso permite el modelado de una vista de negocio del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes pasados entre los objetos. (Presman, 2002)

En el artefacto Modelo de Diseño se encuentran los siguientes diagramas: el diagrama genérico de las clases del diseño, un diagrama de diseño específico para cada Caso de Uso del sistema y un diagrama de secuencia por sección de cada Caso de Uso.

3.3. Diseño de la Base de Datos

Una Base de Datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización. (Marqués, 2001)

Un Sistema de Gestión de la Base de Datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la Base de Datos. Proporciona un acceso controlado a la misma, provee seguridad, integridad, concurrencia y controla la recuperación ante fallos. Además, presenta un mecanismo de vistas que permite mostrar a los usuarios sólo aquellos datos que les interesan. (Marqués, 2001)

El diseño de la Base de Datos de un sistema consta de tres fases: diseño conceptual, diseño lógico y diseño físico de la Base de Datos.

3.3.1. Modelo conceptual

El modelo conceptual consiste en construir un esquema de la información que se va a utilizar, independiente de cualquier consideración física. Este esquema conceptual permite identificar las entidades, atributos y relaciones, con el objetivo de comprender la perspectiva que cada usuario tiene de los datos.

El esquema conceptual se construye utilizando la información que se encuentra en la especificación de los requisitos del software y constituye una fuente de información para el diseño lógico de la base de datos. (Marqués, 2001)

3.3.2. Modelo lógico de datos

El modelo conceptual se refina en un modelo lógico, eliminando las construcciones que no se pueden representar en el modelo de la base de datos. Conforme se va desarrollando el esquema lógico, éste se va probando y validando con los requerimientos. (Marqués, 2001)

El esquema lógico desempeña un papel importante en la etapa de mantenimiento del sistema, ya que permite que los futuros cambios que se realicen en la aplicación o en los datos, se representen correctamente en la base de datos.

En el artefacto Modelo de Datos se encuentra especificado el modelo lógico de datos para el Sistema de Registro Electrónico Directo para las votaciones en la UCI, incluyendo el diagrama de clases persistentes.

3.3.2.1. Diagrama de clases persistentes

Para el diseño del modelo lógico de la base de datos del sistema, se parte de un diagrama de clases persistentes, definiendo la persistencia como la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Una vez conformado dicho diagrama se puede obtener el modelo físico de datos.

3.3.3. Modelo físico de datos

El esquema lógico se traduce en un esquema físico para el SGBD. La fase de diseño físico considera las estructuras de almacenamiento y los métodos de acceso necesarios para proporcionar un acceso eficiente a la base de datos. (Marqués, 2001)

El propósito del diseño físico es describir cómo se va a implementar el esquema lógico obtenido en la fase anterior. Concretamente consiste en:

- ✓ Obtener un conjunto de relaciones y las restricciones que se deben cumplir sobre ellas.
- ✓ Determinar las estructuras de almacenamiento y los métodos de acceso que se van a utilizar para conseguir prestaciones óptimas.

- ✓ Diseñar el modelo de seguridad del sistema.

En el artefacto Modelo de Datos se detalla el modelo físico de datos para el Sistema de Registro Electrónico Directo para las votaciones en la UCI, incluyendo el diagrama entidad-relación y la descripción de las tablas de la base de datos.

3.3.3.1. Diagrama entidad-relación

El modelo conceptual más utilizado es el modelo entidad-relación, que posee los siguientes conceptos: entidades, relaciones, atributos, dominios de atributos, identificadores y jerarquías de generalización.

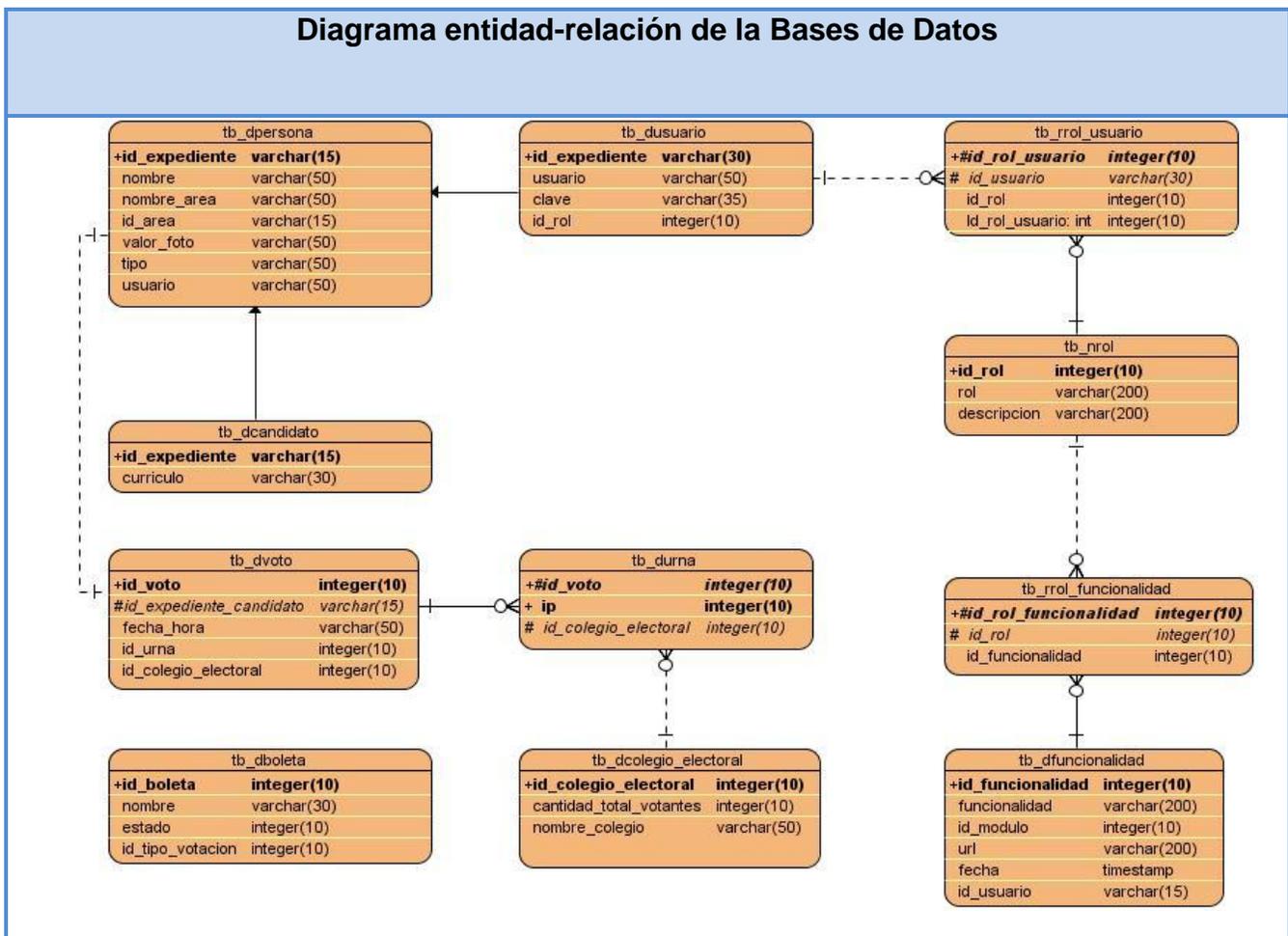


Ilustración 9: Modelo Entidad-Relación de la Base de Datos

3.4. Conclusiones parciales

En este capítulo se describió la estructura arquitectónica propuesta para la solución. Quedando fundamentados los patrones definidos para el diseño e implementación del sistema, contribuyendo a la reutilización y a una mayor comprensión del mismo. Se detallaron los elementos del diseño, describiéndose las clases y sus relaciones; así como el modelo de datos, describiéndose cada una de las tablas de la base de datos. Con el diseño propuesto, se materializan con precisión los requerimientos definidos para el sistema, proporcionando la idea de lo que se quiere construir, sirviendo como guía para la implementación.

Capítulo 4

4.1. Introducción

En este capítulo se da inicio a la fase de construcción presentando el modelo de implementación, en el mismo se exponen los elementos por los que se compone el modelo de diseño, los cuales se implementan en términos de componentes. En este modelo de implementación, se describe como se organizan los componentes disponibles en el entorno de implementación y en el lenguaje de programación a utilizar. Además, se presentan los casos de pruebas para cada caso de uso, siguiendo específicamente el método de pruebas de caja negra. En esta fase todas las características, componentes, y requerimientos deben ser integrados, implementados, y probados en su totalidad.

4.2. Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas que se establecen entre los componentes de hardware y de software en el sistema, es decir, indica la situación física de los componentes lógicos desarrollados, lo que significa que sitúa el software en el hardware que lo contiene.

Para la estructura de la posible distribución del sistema de votaciones se utilizó un nodo PC_Urna que representa las computadoras en la que los electores van a efectuar el voto y otro nodo para obtener los reportes de las votaciones. Estos nodos se conectarán mediante el protocolo de Transferencia de Hipertexto (HTTP por sus siglas en inglés, HyperText Transfer Protocol) al nodo Servidor Web de Aplicaciones y a este se conectará al Directorio Activo mediante el Protocolo Ligero de Acceso a Directorios (LDAP por sus siglas en inglés, Lightweight Directory Access Protocol). Desde este servidor web de Aplicaciones se puede acceder a los nodos Servidores de Akademos, Identidad y ASSETS mediante el Protocolo Simple de Acceso a Objetos (SOAP por sus siglas en inglés, Simple Object Access Protocol).

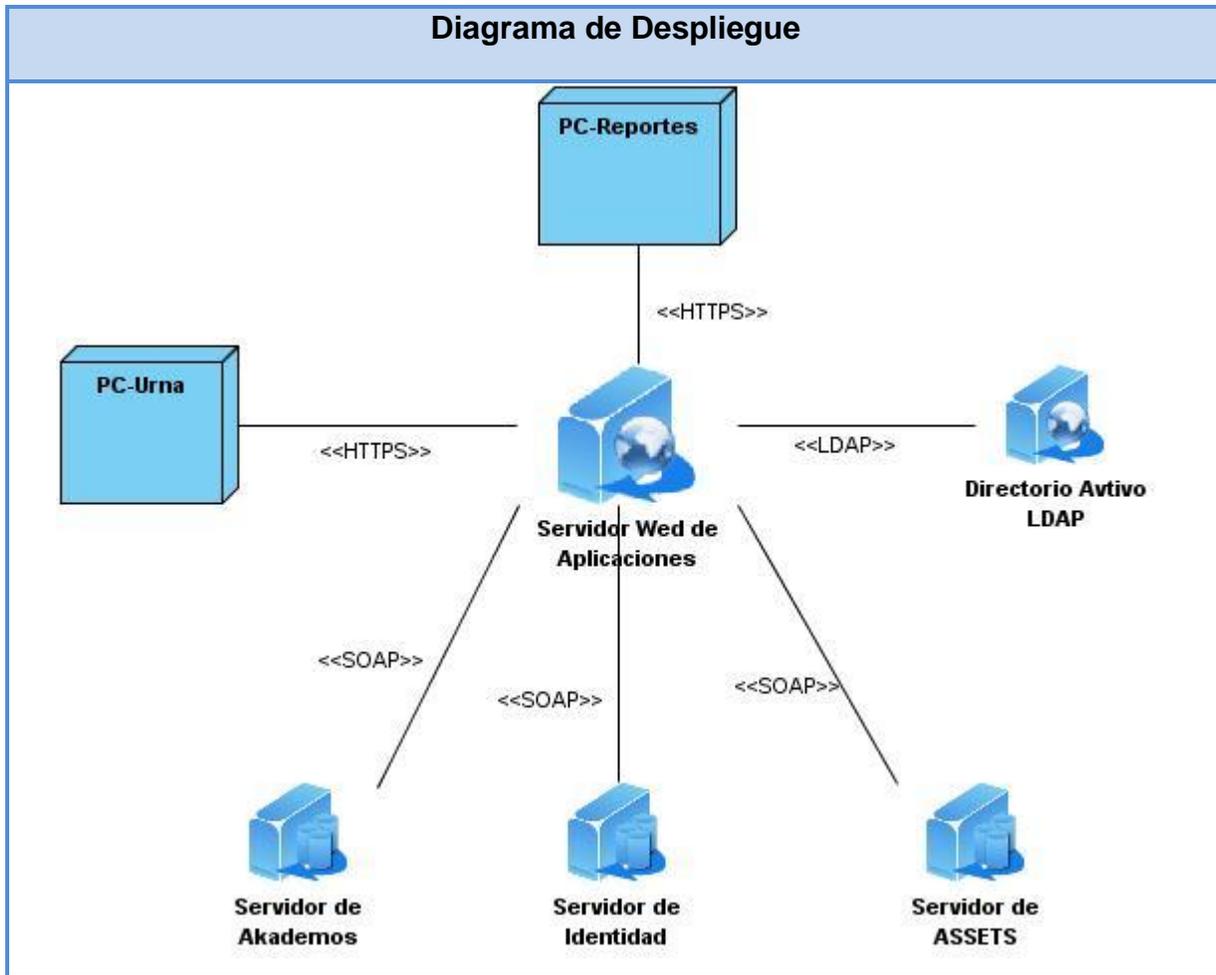


Ilustración 10: Diagrama de despliegue

4.3. Diagrama de componentes

Un diagrama de componentes describe los tipos de elementos de software que entran en la fabricación de la aplicación. Estos elementos pueden agruparse en paquetes según la lógica, con la intención de simplificar la implementación. Los diagramas de componentes se relacionan con los diagramas de clases. Estos diagramas se utilizan para modelar la vista estática de un sistema y muestra la organización y las dependencias lógicas entre un conjunto de componentes software, sean estos componentes de código fuente, binarios o ejecutables.

Los diagramas de componentes son la representación de los componentes unidos al funcionamiento del framework CodeIgniter y tienen como objetivo figurar la estructura general de la aplicación.

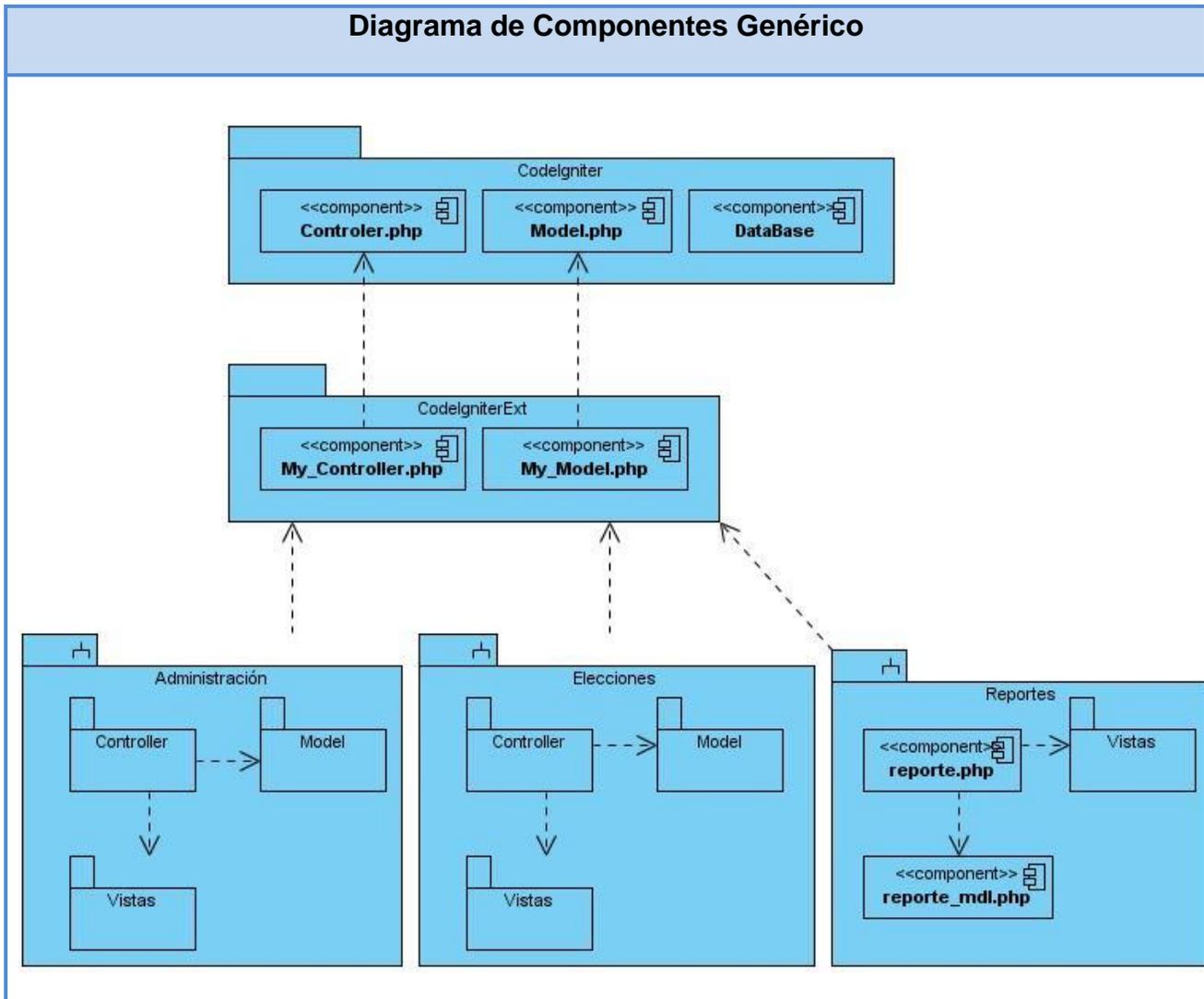


Ilustración 11: Diagrama de componentes genéricos

La descripción de los componentes de implementación de cada módulo se encuentra detallada en el artefacto Modelo de Implementación.

4.4. Métodos de prueba

Una de los últimos flujos del ciclo de vida antes de entregar un programa para su explotación, es el flujo de trabajo de pruebas. Este flujo es una actividad en la cual un sistema es ejecutado bajo requerimientos especificados, los resultados son observados y registrados, haciendo finalmente una evaluación del sistema. Dentro de este flujo se pueden desarrollar pruebas al software con el objetivo de garantizar la calidad del producto desarrollado lo que implica

- ✓ Verificar la interacción de componentes.
- ✓ Verificar la integración adecuada de los componentes.
- ✓ Verificar que todos los requisitos se han implementado correctamente.
- ✓ Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el software al cliente.
- ✓ Diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

4.4.1. Resultados de las pruebas de caja negra

Uno de los métodos de prueba aplicado en el sistema fue el de caja negra, el cual se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software.

En una primera iteración de los 17 Casos de Prueba realizados, 3 de ellos detectaron errores en la aplicación. La siguiente gráfica muestra los resultados obtenidos.



Ilustración 12: Resultados de la primera iteración de los CP

Después de resolver las no conformidades detectadas, se realizó una segunda iteración de los Casos de Prueba, obteniéndose estos resultados.



Ilustración 13: Resultados de la segunda iteración de los CP

La gráfica muestra el correcto funcionamiento del sistema, debido a que no se detectaron errores en la segunda iteración de los Casos de Prueba.

4.4.2. Resultados de las pruebas de caja blanca

Las pruebas de caja blanca aseguran que la operación interna se ajusta a las especificaciones, y que todos los componentes internos se han comprobado de forma adecuada. Entre las técnicas usadas se encuentran; la cobertura de caminos, pruebas sobre las expresiones lógico-aritméticas, pruebas de camino de datos, comprobación de bucles.

Tipos de prueba de caja blanca:

- ✓ Prueba de Condición: Es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa.
- ✓ Prueba de Flujo de Datos: Se seleccionan caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.
- ✓ Prueba de Bucles: Es una técnica de prueba de caja blanca que se centra exclusivamente en la validez de las construcciones de bucles.
- ✓ Prueba del Camino Básico: Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho

conjunto de caminos independientes se construye el grafo de flujo asociado y se calcula su complejidad ciclomática.

La técnica del camino básico fue aplicada al Sistema de Registro Electrónico Directo para las votaciones en la UCI, la misma permitió obtener una medida de la complejidad lógica del código para los métodos críticos del sistema, logrando que se ejecutaran todos los bucles en sus límites operacionales.

De los casos de prueba aplicados dos de ellos demostraron que en los métodos Buscar Candidato Registrar urna existían errores lógicos.

A continuación se muestra el resultado de la prueba aplica al método Buscar Candidato.



```
C:\WINDOWS\system32\cmd.exe
C:\wamp\php\includes>C:\wamp\php\php.exe phpunit.php ejemplo.php
PHPUnit 3.4.0 by Sebastian Bergmann.

F

Time: 7 seconds
There was 1 failure:

1) StackTest::testBuscarCandidato
Failed asserting that two arrays are equal.
--- Expected
+++ Actual
@@ @@
Array
(
    [idPersonal =>

C:\wamp\php\includes\ejemplo.php:18
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
C:\wamp\php\includes>
```

Ilustración 14: Resultado insatisfactorio del método Buscar Candidato

La prueba demuestra la existencia de errores, los que son corregidos como se muestra en la siguiente figura.



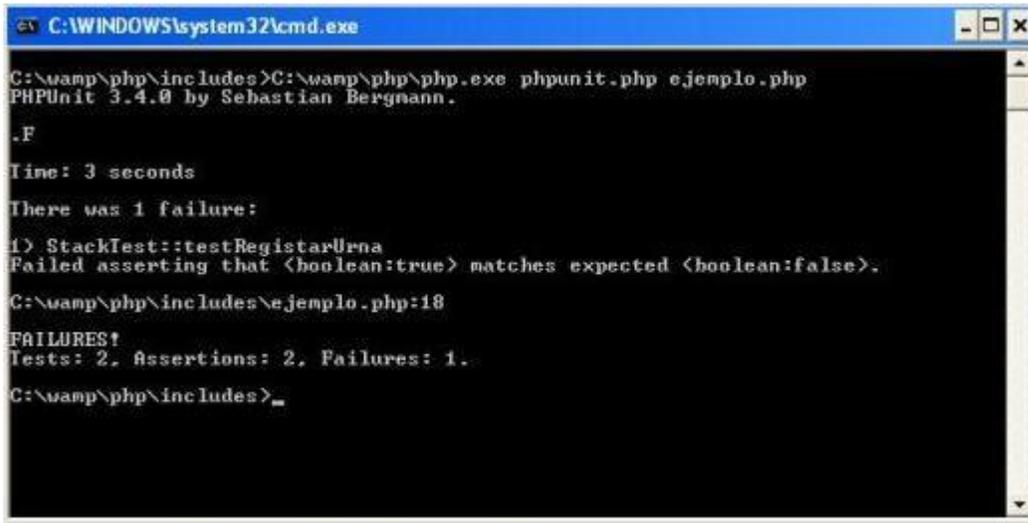
```
C:\WINDOWS\system32\cmd.exe
C:\wamp\php\includes>C:\wamp\php\php.exe phpunit.php ejemplo.php
PHPUnit 3.4.0 by Sebastian Bergmann.

.

Time: 10 seconds
OK (1 test, 1 assertion)
C:\wamp\php\includes>
```

Ilustración 15: Resultado satisfactorio del método Buscar Candidato

De igual forma sucede con el método Registrar Urna. Seguidamente se muestran los resultados.

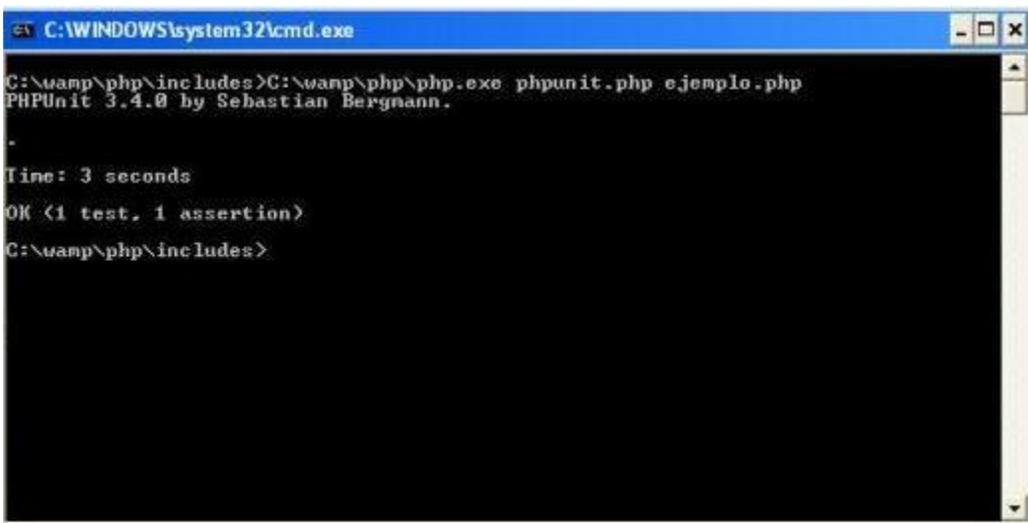


```
C:\WINDOWS\system32\cmd.exe
C:\uamp\php\includes>C:\uamp\php\php.exe phpunit.php ejemplo.php
PHPUnit 3.4.0 by Sebastian Bergmann.

.F
Time: 3 seconds
There was 1 failure:
1) StackTest::testRegistrarUrna
Failed asserting that <boolean:true> matches expected <boolean:false>.

C:\uamp\php\includes\ejemplo.php:18
FAILURES!
Tests: 2, Assertions: 2, Failures: 1.
C:\uamp\php\includes>_
```

Ilustración 16: Resultado insatisfactorio del método Registrar



```
C:\WINDOWS\system32\cmd.exe
C:\uamp\php\includes>C:\uamp\php\php.exe phpunit.php ejemplo.php
PHPUnit 3.4.0 by Sebastian Bergmann.

.
Time: 3 seconds
OK (1 test, 1 assertion)
C:\uamp\php\includes>
```

Ilustración 17: Resultado satisfactorio del método Registrar Urna

Una descripción más detallada de los casos de pruebas se encuentra en el artefacto de Casos de Pruebas.

4.5. Conclusiones parciales

Con la realización de este capítulo se concluye que han sido representados los elementos necesarios para la implementación del sistema. Se modelaron los diagramas de despliegue y de componentes, quedando conformado el modelo de implementación para la fabricación del sistema. Se realizaron también las pruebas de caja negra a los casos de usos con lo cual quedan probadas que las funcionalidades implementadas se desarrollaron correctamente.

Conclusiones Generales

Con este trabajo se propone un Sistema de Registro Electrónico Directo para las votaciones en la Universidad de las Ciencias Informáticas. Con el cumplimiento de los objetivos y las tareas trazadas, se obtuvieron las siguientes conclusiones:

- Se realizó un análisis de los principales conceptos asociados a los Sistemas Electorales y se profundizó en el estudio de los Sistema de Registro Electrónico Directo como tipo de Sistema de Votación Digital más adecuado a utilizar para las votaciones en la UCI.
- Se analizó y seleccionó la metodología, tecnología y herramienta para el desarrollo de la aplicación web.
- Se documentaron los flujos de trabajo: Modelamiento de Negocio, Requerimientos, Diseño, Implementación y Prueba.
- Se diseñó una aplicación que cumple con los requerimientos trazados.
- Se implementó una aplicación web para las votaciones en la UCI que brinda la posibilidad de selección inmediata de los candidatos.
- La aplicación permite conocer la cantidad de votos en tiempo real, lo que acelera la etapa de escrutinio y consecuentemente la entrega de los resultados, optimizando la utilización de los recursos materiales y humanos.
- Se realizaron las actividades de prueba de liberación del producto implementado.

Recomendaciones

Una vez vencidos los objetivos de esta investigación, y teniendo en cuenta las experiencias obtenidas a lo largo de su desarrollo se recomienda:

- ✓ Profundizar en el estudio del desarrollo de los procesos electorarios del Poder Popular, con el objetivo de continuar el desarrollo de este sistema, para emplearlo en este tipo de votaciones.
- ✓ Desplegar la aplicación en las próximas elecciones que se realicen en la Universidad de las Ciencias Informáticas.
- ✓ Implementar una versión superior de la aplicación que permita imprimir los reportes que se obtienen en tiempo real de las votaciones.

Referencias Bibliográficas

- Alberto Ruiz. 2009.** Entorno de desarrollo NetBeans. [En línea] 13 de 7 de 2009.
<http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=720>.
- Alvarez, Miguel Angel. 2003.** Zend Studio. [En línea] 4 de 06 de 2003.
<http://www.desarrolloweb.com/articulos/1178.php>.
- Aston. 2007.** Rational Rose Enterprise. [En línea] 29 de 08 de 2007. [Citado el: 20 de 02 de 2010.]
<http://www.taringa.net/posts/downloads/883942/Rational-Rose-Enterprise-Edition-2002.html>.
- Barriento, Manuel Sánchez. 2008.** Introducción a BPMN. [En línea] 2 de 11 de 2008. [Citado el: 15 de 02 de 2010.] <http://www.aprendergratis.com/introduccion-a-bpmn.html>.
- Berzal, Fernando y Cortijo, Francisco J. & Cubero, Juan Carlos. 2007.** *Desarrollo profesional de aplicaciones web con ASP.NET*. 2007.
- Bunge, Tomás Ignacio. 2004.** *El voto electrónico, esnobismo de la era cibernética*. Argentina : Ciudad Política, 2004.
- Busaniche, Beatriz, Heinz, Federico y Rezinovsky, Alfredo. 2008.** *VOTO ELECTRÓNICO: LOS RIESGOS DE UNA ILUSIÓN*. Córdoba : Fundación Vía Libre, 2008.
- Diagramas de Interacción. [En línea] <http://www-gris.det.uvigo.es/~avilas/UML/node41.html>.
- Electoral, Dirección Nacional. 2005.** *Nuevas Tecnologías y Procesos Electorales*. Argentina : Ministerio del Interior de Argentina, 2005.
- García, Miguel Ángel. 2002.** *Tutorial de Java Server Pages*. 2002.
- Garrett, Jesse James. 2005.** AJAX un nuevo acercamiento a Aplicaciones Web. [En línea] 18 de 02 de 2005. [Citado el: 14 de 03 de 2010.] <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>.
- Garza, Luis M. Gonzáles de la. 2008.** *Voto electrónico por internet, Constitución y riesgos para la democracia* . Madrid : Edisofer , 2008.
- Gassmann, Alejandro. 2002.** Manual de introducción al JavaScript. [En línea] 18 de 06 de 2002.
- Larman, Craig. 2004.** *“UML y patrones”*. s.l. : Tomo I Capítulos 18, Páginas 185-215., 2004.
- Leopoldo, Carlos. 2007.** Zend Framework, una introducción. [En línea] 27 de 11 de 2007. [Citado el: 14 de 03 de 2010.] <http://techtastico.com/post/zend-framework-una-introduccion/>.
- Marqués, Mercedes. 2001.** *Apuntes de Ficheros y Bases de Datos*. España : s.n., 2001.

Mateu, Carles. 2004. *Desarrollo de aplicaciones Web*. Barcelona : Universidad Obrera de Catalunya, 2004.

Pérez, Javier Eguíluz. 2009. *CSS avanzado*. 2009.

Potencier, Fabien. 2008. *Symfony la guía definitiva*. 2008.

Presman, R. S. 2002. *Ingeniería del Software, un enfoque práctico*. s.l. : Félix Varela, 2002.

Rial, Juan. 2001. *“Modernización del Proceso Electoral: el Voto Electrónico en América Latina”*. s.l. : Programa de Reforma Política del PNUD, 2001.

Rico, Gerardo Ruiz. 1997. *Formas de Gobierno y Sistemas Electorales*. España : Tirant Lo Blanch, 1997.

Rivera, Volkan. 2007. Codelgniter un PHP Framework fácil y ligero. [En línea] 2 de 08 de 2007. [Citado el: 20 de 02 de 2010.] <http://www.volkanrivera.com/esp/?p=24>.

Rodríguez González, Pilar. 2008. *Estudio de la Aplicación de Metodologías Ágiles para la Evolución de Productos Software*. Organización y Estructura de la Información, E.U. de Informática (UPM). 2008. Tesis (Master).

Rojas, Iván Guzmán de. 2000. *“Automatización de los procesos electorales”*. Instituto Interamericano de Derechos Humanos : San José, 2000.

SANCHEZ, M. A. M. 2004. *Metodologías De Desarrollo De Software*. 2004.

Tula, Maria Ines. 2007. *Voto electrónico*. España : ARIEL, 2007.

Web. 2010. Aplicaciones Web vs. Escritorio (Desktop). [En línea] 2010. [Citado el: 10 de 2 de 2010.] <http://alxplus.blogspot.com/2006/08/aplicaciones-web-vs.html>.

Zubyc. 2009. POO en PHP 5. [En línea] 10 de 10 de 2009. [Citado el: 20 de 02 de 2010.] <http://www.php-hispano.net/novedades/1252616380-ayuda-para-haitiacute.html>.

Anexos

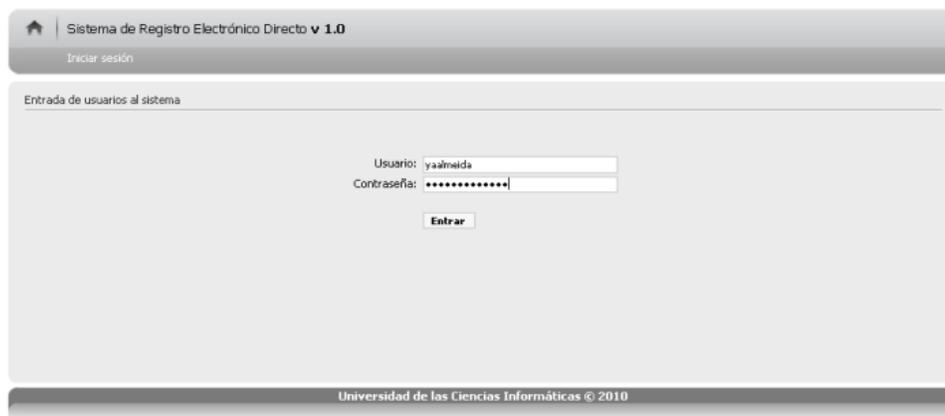


Ilustración 18: Autenticarse en el sistema

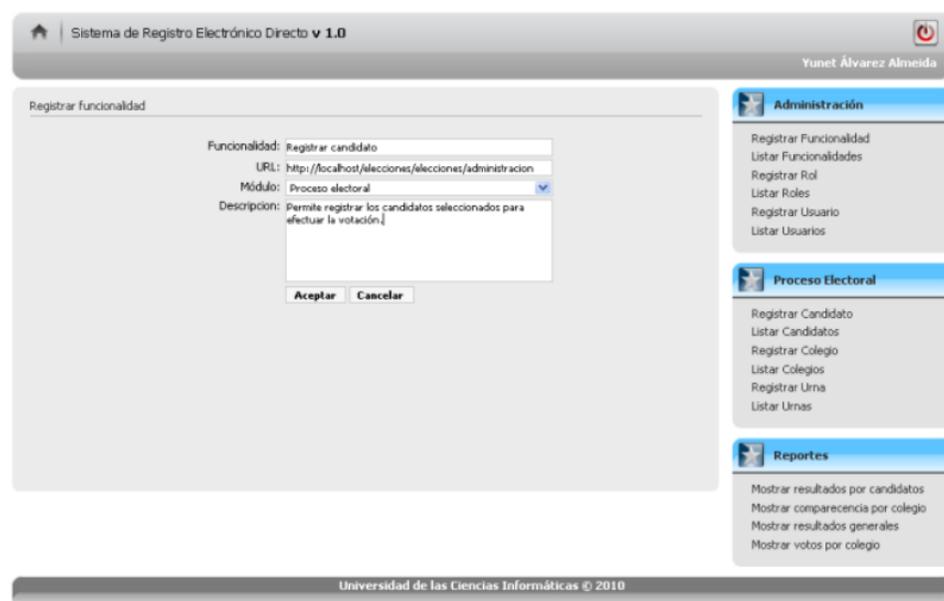


Ilustración 19: Funcionalidad Registrar Candidato