

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



FACULTAD DE CIENCIAS Y TECNOLOGÍAS COMPUTACIONALES

*Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas*

Título:

SILECTOS: Sistema para la Gestión de Electos v2.0

Autor:

Lizsandra Díaz Grecesqui

Milagros García Morán

Tutores:

Ing. René Leandro Cruz Laguna

Ing. Aldis Joan Abreu Medina

Consultante:

Félix González Martínez

La Habana, junio de 2017

“Año 59 de la Revolución”



*“No se vive celebrando derrotas sino
superando victorias”*

le

Declaración de Autoría

Declaramos ser autores del presente trabajo de diploma “SILECTOS: Sistema para la Gestión de Electos v2.0”, y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Autores:

Lizsandra Díaz Grecesqui

Milagros García Morán

Tutores:

Ing. René Leandro Cruz Laguna

Ing. Aldis Joan Abreu Medina

Datos de Contacto

Autores:

Nombre: Lizardsandra Díaz Grecesqui

Correo electrónico: lgrecesqui@estudiantes.uci.cu

Nombre: Milagros García Morán

Correo electrónico: mmoran@estudiatas.uci.cu

Tutores:

Nombre: Ing. René Leandro Cruz Laguna

Especialidad de graduación: Ingeniero en Ciencias Informáticas

Años de experiencia en el tema: 4

Años de graduado: 4

Email: rlacruz@uci.cu

Nombre: Ing. Aldis Joan Abreu Medina

Especialidad de graduación: Ingeniero en Ciencias Informáticas

Categoría docente: Instructor

Años de experiencia en el tema: 7

Años de graduado: 7

Email: ajabreu@uci.cu

Consultante:

Nombre: Ing. Felix González Martínez

Especialidad de graduación: Ingeniero en Ciencias Informáticas

Años de experiencia en el tema: 4

Años de graduado: 4

Email: felix@uci.cu

Agradecimientos

Hoy finaliza mi etapa de estudiante y comienza una aún más difícil pero necesaria: ser profesional. No fueron solo 5 años de universidad sino 22 largos años de aprendizaje y superación. Gracias Mí, Pí, Puduchi, por ser la mejor familia. Por ustedes he llago a ser quien soy y les demostraré que puedo ser mucho más. Quiero que sepan que, si pudiese elegir a mi familia, no escogería a nadie que no fuesen ustedes porque, sobre todas las cosas, nunca se nos ha olvidado que somos una familia y esa es la base para superar cualquier obstáculo en la vida.

Mimita querida, ¿qué te puedo decir a ti? Tu siempre estuviste ahí para todo lo que necesité. En cada circunstancia, buena o mala, quiero que sepas que nunca me fallaste y, sobre todo, que siempre me estuviste dando lecciones para la vida. Ahora me vienen a la mente todas esas veces que te pregunté ¿Mí, tú me quieres? Quiero que sepas que nunca dudé ni por un segundo tu respuesta a pesar de las veces que no me la decías, pero la verdad es que me gusta escucharte cuando me dices "Claro que si te quiero" o "¿Cómo no te voy a querer?" Te adoro con la vida y espero que Diosito me de muchos años a tu lado para poder seguir escuchándote decir esas palabras mágicas, que para mí lo son todo.

Desde niña escuché una frase que me guió durante toda la vida: "Tú estudia lo que quieras, pero lo que comiences, lo terminas". Mi Pí, mi papito lindo, no eres capaz de imaginarte cuántas veces me aferré a esa frase para no desistir en estos 5 años de carrera. Gracias por darme esa gran lección, ser una ingeniera en estos momentos, la mayor parte se lo debo a tus sabias palabras. Sobre todo, gracias por demostrarme que siempre puedo contar contigo, que lo más importante para mi vida personal es que, no importa lo que digan la gente, si yo me lo propongo entonces yo lo logro. Te quiero mucho papá.

Gracias a mi nana, mi tata, mi hermana, mi Puduchi, Tay. Quisiera que estuviese aquí para poder decirle todas las cosas que me he guardado por más de cuatro años y, a pesar de saber que no vendrás, no pierdo las esperanzas de ver que entres por la puerta y me digas "Nana, vine, aquí estoy". Yo sé que si no está aquí conmigo no es porque no quiera y también sé cuánto le debe doler perderse este momento. Pero tata, lo bueno es que ya tienes una hermana ingeniera, que todos tus esfuerzos no fueron en vano y que puedes estar orgullosa de mí. Sé que hemos compartido muy pocos momentos juntas, y que la mayor parte del tiempo peleábamos, pero no dudes ni por un momento que te amo mucho y que tú eres mi ídolo, mi ejemplo a seguir. Mientras más metas alcances, más propósitos me trazaré y te prometo, por ti y por mí, que los voy a cumplir.

A Kiki, mi cuñi, gracias por apoyarme todo este tiempo aunque no nos conociéramos en persona. Pero, sobre todo, te agradezco que no dejes sola a mi hermana, por ser su apoyo ahora que está lejos de nosotros. Dale muchos motivos para reír y haz por ella lo que la distancia impide que yo haga.

A mis abuelos Arnoldo y Norberto, así como mis tíos Tania y Enrique que, aunque no esté para escucharme decir "Ya me gradué, ya soy ingeniera", sé que desde el cielo me están mirando y apoyando.

A mi hermana de corazón, la que yo escogí y ella me demostró que no me equivoqué: Yenisey. Gracias por compartir estos largos 9 años y permitirme conocer una nueva familia. Por soportar mis cambios de humor, mi mal genio y por correr en las noches hasta mi apartamento cada vez que me deprimía.

A mis tíos y primos, por todos esos momentos en familia, por siempre estar al tanto del progreso de mi tesis, por sus exigencias y sus ánimos.

A los profesores de la universidad por contribuir en mi formación integral ya sea en el aula, en el escenario o en las canchas.

A mi familia de la UCI: Lenia, Midalis, Suinny, Yanelis, Claudia, Saydi, Marlon, Rolyen, Ramiro, Osmin, Rubert, Yuniór, Lázaro, Teudis, Leonel, Roger, Leodán, Pedro, Javier y Adolfo. Sin ustedes no me imagino como hubiesen sido estos años. Gracias por ser los mejores.

Al piquete de los festivales y juegos. Gracias por esos momentos felices que nunca olvidaré.

A mi dúo de tesis por las noches de sueños compartidas durante la tesis y por todo el esfuerzo realizado para llevar a cabo la tesis a pesar de estar lejos de su bebé.

A mis tutores René, Félix y Aldis por estar siempre al pendiente de la tesis, por las tantas noches de desvelo y, sobre todo, por los conocimientos brindados, por dedicarme su tiempo y poder contar con ellos en todo momento. También a todos los trabajadores del proyecto Elecciones por recibirnos como un miembro más del laboratorio durante todo el curso y por la ayuda que nos brindaron todo el tiempo.

A los miembros del tribunal por sus sabias críticas constructivas.

En fin, gracias a todas esas personas y familiares que de una forma u otra me ayudaron y formaron parte de mi vida en estos 5 años de la carrera.

Lizsandra Díaz Grecesqui

Agradezco a mis padres por todo su amor y dedicación, por estar siempre a mi lado en los momentos difíciles dando apoyo y fuerzas para seguir adelante por mostrarme el camino correcto para ser mejor persona cada día, por sus días de desvelo y preocupación pendientes de mis estudios, todas estas palabras no son suficientes para describir y agradecer la infinidad de cosas que han hecho por mí. A mis tíos que también son mis padres y durante toda mi vida no han dejado de guiarme, cuidarme y sacarme de apuros cada vez que los he necesitado sin poner reparo por muy difícil que estuviera la situación, muchas gracias por contribuir a realizar este sueño. A mis hermanos (Antonio, Yenier y Adriel) que son mis ejemplos y mis paños de lágrimas, que han sabido complacer mis caprichos y aconsejarme cada vez que los he necesitado, sin quitar sus regaños para formar en mis mejores valores. A mis abuelas por darme tanto amor y cariño por llevarme a la escuela, aunque llegáramos tarde, por sus juegos y por transmitirme esa ternura que las caracteriza. Y muy especialmente quiero agradecerle a una persona que llegó a mi vida para cambiarla totalmente, hacer de ella una revolución y hacer que experimentara lo más bello que puede existir en el mundo ser madre. A mi hija Ainhoa Isabella que la amo con todo mi corazón y cada vez que he estado cansada de batallar me abraza con esas manecitas tan pequeñas dándome un beso y diciéndome te quiero mamá. Haciendo que para mí la vida vuelva a tener el mayor sentido. A mis sobrinas que las quiero muchos. Agradecer de forma general a toda mi familia por su apoyo incondicional y su cariño. A mi tutor por toda su ayuda y por sus memes que no pueden faltar, a mi oponente por su ayuda y al resto del tribunal por todas sus críticas constructivas que ayudaron a la elaboración de esta tesis.

Milagros García Morán

Dedicatoria

Dedicado a:

A mis abuelos Arnoldo y Norberto, porque a pesar de no estar físicamente para verme convertida en ingeniera, sé que desde el cielo me guían y me protegen.

A mi hermana por ser mi ídolo, por demostrarme que puedo conseguir todo lo que me proponga y por siempre estar orgullosa de mí.

A mis padres por ser las personas más importantes en mi vida y por enseñarme a nunca rendirme.

Lizandra Díaz Grecesqui

Dedico esta tesis a mi hija Ainhoa Isabella, a mis padres y a mis hermanos. Los amo.

Milagros García Morán

Resumen

El Centro de Tecnologías de Gestión de Datos de la Universidad de las Ciencias Informáticas cuenta con el Proyecto para la Informatización de los Procesos Electorales Cubanos, encargado de desarrollar los sistemas informáticos de apoyo al trabajo realizado en las comisiones electorales de Cuba. El software encargado de la gestión de la información de los electos posterior a los procesos eleccionarios Sistema para la Gestión de Electos v1.0 presenta un grupo de deficiencias en la realización de sus tareas. Debido a esto, el objetivo de la presente investigación es desarrollar un nuevo sistema que contribuya a la gestión de dicha información. Para guiar el proceso de desarrollo del sistema se empleó la metodología AUP_UCI, el lenguaje de modelado UML v2.5, las herramientas Visual Paradigm para UML v8.0, NetBeans IDE v8.0, ExtJS v4.2, JavaScript, Symfony v2.7, PHP v5.5, YAML, PostgreSQL v9.4, PgAdmin III, JasperReports v6.1, Tomcat v7.0 y Apache Web Server v2.2. Además, se realizó el análisis, diseño e implementación del software obteniéndose como resultado el Sistema para la Gestión de Electos v2.0. Para comprobar el correcto funcionamiento del sistema se realizaron pruebas funcionales, de seguridad y de rendimiento

Palabras clave

Delegado, Diputado, Elecciones, Electos. Procesos.

Abstract

The Center for Data Management Technologies of the University of Information Sciences has the Project for the Computerization of Cuban Electoral Processes, in charge of developing the computer systems to support the work carried out in the electoral commissions of Cuba. The software in charge of the management of the information of the elect after the electoral processes System for the Management of Elect v1.0 presents a group of deficiencies in the accomplishment of its tasks. Due to this, the objective of the present investigation is to develop a new system that contributes to the management of this information. To guide the development process of the system we used the AUP_UCI methodology, the UML v2.5 modeling language, the Visual Paradigm tools for UML v8.0, NetBeans IDE v8.0, ExtJS v4.2, JavaScript, Symfony v2.7, PHP v5.5, YAML, PostgreSQL v9.4, PgAdmin III, JasperReports v6.1, Tomcat v7.0 and Apache Web Server v2.2. In addition, we performed the analysis, design and implementation of the software resulting in the System for the Management of Elect v2.0. In order to verify the correct functioning of the system, functional, safety and performance tests were carried out

Keywords

Delegate, Deputy, Elections, Elected. Processes

Índice de Contenido

Introducción	1
Capítulo 1: Fundamentación teórica del Sistema para la Gestión de Electos v2.0	5
1.1. Conceptos relacionados con el dominio de investigación	5
1.2. Análisis de los sistemas informáticos para los procesos eleccionarios cubanos	7
1.3. Herramientas y tecnologías para la construcción de SILECTOS v2.0	8
AUP_UCI	8
UML v2.5	9
Visual Paradigm para UML v8.0	9
NetBeans IDE v8.0	10
ExtJS v4.2	10
JavaScript	10
Symfony v2.7	10
PHP v5.5	11
YAML	11
PostgreSQL v9.4	11
PgAdmin III	11
JasperReports v6.1	12
Tomcat v7.0	12
Apache Web Server v2.2	12
1.4. Conclusiones del capítulo	13
Capítulo 2: Análisis y diseño del sistema SILECTOS v2.0	14
2.1. Modelo del dominio	14
Diagrama conceptual del Modelo de Dominio	15
Especificación de los requisitos del sistema	16
Modelo de Caso de Uso del Sistema	19

2.2. Modelo de diseño	30
Patrón arquitectónico Modelo-Vista-Controlador	30
Diagrama de paquetes	30
Diagrama de clases del diseño.....	33
Diagrama de secuencia.....	37
Diagrama entidad-relación.....	38
Diagrama de despliegue.....	39
2.3. Conclusiones del capítulo	40
Capítulo 3: Implementación y pruebas del sistema SILECTOS v2.0	41
3.1. Modelo de implementación	41
Diagrama de componentes.....	41
Código fuente	43
3.2. Pruebas del software	45
Niveles de pruebas.....	45
Tipos de pruebas.....	46
Métodos de prueba.....	46
Pruebas funcionales aplicando el método Caja Negra.....	46
Pruebas funcionales aplicando el método Caja Blanca	49
Pruebas de seguridad	50
Pruebas de rendimiento (Carga).....	51
Pruebas de aceptación.....	52
3.3. Conclusiones del capítulo	52
Conclusiones generales.....	53
Referencias Bibliográficas.....	54

Índice de Tablas

Tabla 1: Comparación de soluciones existentes	7
Tabla 2: Definición de las clases del modelo de dominio	15
Tabla 3: Descripción de los actores del sistema SILECTOS v2.0	21
Tabla 4: Descripción textual del caso de uso Gestionar Delegados a la AMPP (M).	23
Tabla 5: Descripción de las variables para el caso de prueba Gestionar delegado a la AMPP (M)	47
Tabla 6: Caso de Prueba. Escenario Buscar Delegado a la AMPP (M).....	47
Tabla 7: Caminos básicos.....	50
Tabla 8: : Resultados de las pruebas de seguridad.....	50
Tabla 9: Resultado de las pruebas de Rendimiento (Carga)	51

Índice de Figuras

Fig. 1: Modelo de dominio para SILECTOS v2.0.....	15
Fig. 2: Diagrama de Casos de Uso del Sistema.....	20
Fig. 3: Patrón Múltiples Actores - Rol Común del diagrama de CUS.....	22
Fig. 4: Patrón Extensión Concreta por Extensión del diagrama de CUS.....	22
Fig. 5: Patrón Inclusión Concreta por Inclusión del diagrama de CUS.....	22
Fig. 6: Diagrama de paquetes del sistema SILECTOS v2.0.....	31
Fig. 7: Diagrama de paquetes del sistema SILECTOS v2.0. Estructura del marco de trabajo ExtJS v4.2 dentro del directorio js.....	32
Fig. 8: Diagrama de Clase de Diseño del CU Gestionar Delegado AMPP (M).....	33
Fig. 9: Patrón Controlador del Diagrama de Clase de Diseño del CU Gestionar Delegado AMPP (M).....	35
Fig. 10: Patrón Bajo Acoplamiento del Diagrama de Clase de Diseño del CU Gestionar Delegado AMPP (M).....	35
Fig. 11: Patrón Alta Cohesión Diagrama de Clase de Diseño del CU Gestionar Delegado AMPP (M).....	36
Fig. 12: Patrón GoF Observer Pattern.....	36
Fig. 13: Patrón GoF Mediator Pattern.....	37
Fig. 14: Diagrama de secuencia. Sección Modificar Delegado AMPP (M).....	37
Fig. 15: Diagrama entidad-relación del esquema data del sistema SILECTOS v2.0.....	38
Fig. 16: Diagrama de despliegue de SILECTOS v2.0.....	39
Fig. 17: Diagrama de componentes del CU Gestionar delegados a la AMPP (M).....	42
Fig. 18: Ejemplo de código fuente evidenciando los estándares de codificación definidos por el marco de trabajo ExtJS v4.2.....	44
Fig. 19: Ejemplo de código fuente evidenciando los estándares de codificación definidos por el marco de trabajo Symfony v2.7.....	45
Fig. 20: Iteraciones de las pruebas funcionales.....	49
Fig. 21: Grafo de flujo asociado a la funcionalidad.....	49

Introducción

Uno de los procesos políticos más importante de cualquier país son las elecciones, método democrático para designar a los representantes del pueblo; por lo cual, se puede afirmar que constituyen la base de la democracia. Esta se considera un elemento clave para la nominación de los gobernantes, la participación política de los ciudadanos, así como la promoción de una sucesión del poder de manera pacífica y ordenada. (...) *Su importancia para la política radica fundamentalmente, en que es la fuente de legitimación tanto del sistema político como de sus líderes o dirigentes, además de ser el medio a través del cual se verifica la participación política de las masas populares (...)* (Fernández Baeza, y otros).

El Sistema Electoral Cubano está regulado por la Ley Electoral, aprobada por la Asamblea Nacional del Poder Popular (ANPP) desde el 29 de octubre de 1992. En esta se establecen dos tipos de procesos electorarios: Elecciones Parciales y Elecciones Generales. En estos se eligen los Delegados a las Asambleas Municipales del Poder Popular (AMPP), su Presidente y Vicepresidentes; además de los Delegados a las Asambleas Provinciales del Poder Popular (APPP) y los Diputados a la ANPP con sus respectivos dirigentes. También se eligen el Presidente, Primer Vicepresidente, Vicepresidentes, Secretario y demás miembros del Consejo de Estado (ONE, 2011).

La Comisión Electoral Nacional (CEN) en coordinación con Universidad de las Ciencias Informáticas (UCI) se encuentran inmersas en el desarrollo de los sistemas informáticos que favorecen la gestión de la información de los procesos electorales. El Proyecto para la Informatización de los Procesos Eleccionarios del Centro de Tecnologías de Gestión de Datos (DATEC), es el encargado del desarrollo de dichos softwares desde el año 2010.

Actualmente las comisiones electorales utilizan los Sistemas de Gestión e Información del Proceso Electoral: SIGEL Parciales v3.0 y SIGEL Generales v2.0 para el registro de la información de todos los pasos electorales, desde la gestión de las autoridades hasta la constitución de las Asambleas del Poder Popular. Estos facilitan el flujo de revisión de la información entre los niveles municipal, provincial y nacional; fueron utilizados por primera vez durante el proceso de Elecciones Generales 2012-2013.

Una vez concluido el proceso electoral, toda la información referida a los electos resultantes almacenada en los sistemas antes mencionados, es transferida al Sistema para la Gestión de Electos SILECTOS v1.0. Este software es el encargado de mantener actualizado el listado de los electos entre procesos electorarios y genera reportes estadísticos que permiten el apoyo al proceso de toma de decisiones. En caso de ser necesario realizar la sustitución de un electo, ya sea de los Delegados a la AMPP, APPP o Diputados a la

ANPP debido a una revocación o liberación de cargo, el proceso debe ser notificado al nivel inmediato superior para que sea aprobado o rechazado mediante dicha aplicación. Este flujo de información se realiza mediante el uso de dicho sistema.

Sin embargo, el sistema posee insuficiencias en la actualización de la información referida a una persona sustituta, ya que permite que la misma sea modificada por más de un usuario con diferentes roles aún cuando la información debería ser cambiada solo por el usuario correspondiente al nivel que se está modificando en dicho momento. Unido a esto, el sistema permite asignar dos cargos a una misma persona, así como asignarle un municipio por el cual no fue electo. Otro problema detectado fue que el sistema permite asignar a un mismo usuario diferentes niveles dentro de la aplicación, lo que constituye una violación de seguridad en el mismo. Por otra parte, cuando se necesita activar una dirección IP o un usuario que ha sido desactivado anteriormente, es necesario realizarlo a nivel de base de datos debido a que el administrador del sistema no cuenta con una funcionalidad implementada para este fin, lo que genera una demora innecesaria en la solución de dicho problema.

Por todo lo antes expuesto se plantea como **problema de investigación** ¿Cómo contribuir a la gestión de la información de los electos en el sistema SILECTOS v2.0 para el proceso de apoyo a la toma de decisiones en la sustitución de los electos? Definiéndose como **objeto de estudio**: los sistemas informáticos de apoyo a los procesos electorarios cubanos, enmarcado en el **campo de acción**: la gestión de la información de los electos en SILECTOS v2.0.

Para darle solución al problema determinado, se define como **objetivo general**: desarrollar la versión 2.0 de SILECTOS que contribuya a la gestión de la información de los electos en las diferentes comisiones electorales de Cuba. Del mismo, se desglosan los siguientes **objetivos específicos**:

1. Caracterizar los referentes teóricos para la construcción del Sistema para la Gestión de Electos SILECTOS v2.0.
2. Realizar el análisis del Sistema para la Gestión de Electos SILECTOS v2.0
3. Diseñar el Sistema para la Gestión de Electos SILECTOS v2.0.
4. Implementar el Sistema para la Gestión de Electos SILECTOS v2.0.
5. Verificar el correcto funcionamiento del Sistema para la Gestión de Electos SILECTOS v2.0.

Con el propósito de dar cumplimiento a los objetivos específicos definidos, se plantean las siguientes **preguntas de investigación:**

- ¿Cuáles son las herramientas y la metodología en las que se basa la construcción del Sistema para la Gestión de Electos SILECTOS v2.0?
- ¿Cuáles son las funcionalidades y características que debe tener el Sistema para la Gestión de Electos SILECTOS v2.0?
- ¿Cómo estructurar la implementación del sistema SILECTOS v2.0 para que permita una mejor organización del código?
- ¿Cómo verificar el correcto funcionamiento del sistema SILECTOS v2.0?

Para dar respuesta a las preguntas de investigación anteriormente definidas, se proponen las siguientes **tareas de investigación:**

- Sistematización de las concepciones teóricas referidas al proceso posterior a las elecciones para definir las características que debe poseer el Sistema para la Gestión de Electos SILECTOS v2.0.
- Fundamentación de las metodologías, herramientas y tecnologías a utilizar para el desarrollo del Sistema para la Gestión de Electos SILECTOS v2.0.
- Análisis del Sistema para la Gestión de Electos SILECTOS v2.0 que permita determinar los requisitos funcionales que debe poseer el sistema a implementar.
- Diseño del Sistema para la Gestión de Electos SILECTOS v2.0 para proporcionar los detalles acerca de la arquitectura, las interfaces y los componentes del software, necesarios para implementar el sistema.
- Implementación del Sistema para la Gestión de Electos SILECTOS v2.0 para la construcción del software a partir de los resultados obtenidos en las fases Análisis y Diseño.
- Elaboración los casos de prueba que serán aplicados al Sistema para la Gestión de Electos SILECTOS v2.0 que permitan guiar la ejecución del proceso de prueba.
- Aplicación de las pruebas funcionales, de seguridad y de rendimiento al Sistema para la Gestión de Electos SILECTOS v2.0 para verificar el correcto funcionamiento del sistema implementado.

Para el cumplimiento del objetivo general trazado, se utilizarán los siguientes métodos de investigación:

Métodos teóricos:

- Histórico-Lógico: Permitirá el análisis histórico de los softwares desarrollados para apoyar los procesos electorarios en Cuba de forma lógica y secuencial. De esta forma se podrán determinar las características de cada uno de los sistemas.

- Analítico-Sintético: Será empleado en el análisis de los diferentes sistemas informáticos de apoyo a los procesos eleccionarios existentes en Cuba para determinar las características y las limitaciones de cada uno de ellos.

Métodos empíricos:

- Entrevistas no estructuradas: Serán realizadas a los especialistas del Proyecto para la Informatización de los Procesos Eleccionarios Cubanos, con el objetivo de obtener información sobre el funcionamiento del sistema SILECTOS v1.0.

Estructura del trabajo de diploma

El trabajo de diploma presenta la siguiente estructura: introducción, tres capítulos, conclusiones generales, recomendaciones, referencias bibliográficas y anexos.

Capítulo 1: Fundamentación teórica del Sistema para la Gestión de Electos v2.0

En el capítulo se definieron los conceptos relacionados con el dominio de investigación. De igual forma, se realizó un estudio sobre los sistemas informáticos de apoyo a los procesos eleccionarios existentes en Cuba con el fin de identificar posibles soluciones para el problema de investigación definido. También se describió la metodología y las herramientas a emplear en la construcción del Sistema para la Gestión de Electos SILECTOS v2.0.

Capítulo 2: Análisis y diseño del sistema SILECTOS v2.0

En este capítulo se describió la propuesta de solución para el problema de investigación. Se tomó como punto de partida para la construcción del sistema el modelo de dominio, además se especificaron los requisitos funcionales y no funcionales que debe cumplir el software. Se realizaron los diagramas de Casos de Uso, de paquetes, de clases del diseño, de secuencia y entidad-relación para guiar la implementación de SILECTOS v2.0. El diagrama de despliegue se realizó para describir el entorno en que será publicado el sistema.

Capítulo 3: Implementación y prueba del sistema SILECTOS v2.0

En el capítulo se realizó el modelo de implementación de SILECTOS v2.0 para la descripción de la estructura física del sistema y la relación entre sus elementos, además se definieron los estándares de codificación para una mejor comprensión del código por los desarrolladores. Se describieron las pruebas realizadas con el objetivo de comprobar las funcionalidades del software en los diferentes escenarios, para verificar que los resultados sean los esperados.

Capítulo 1: Fundamentación teórica del Sistema para la Gestión de Electos v2.0

Capítulo 1: Fundamentación teórica del Sistema para la Gestión de Electos v2.0

En este capítulo se definen los conceptos relacionados con el dominio de investigación. De igual forma, se realiza un estudio sobre los sistemas informáticos de apoyo a los procesos electorarios existentes en Cuba con el fin de identificar posibles soluciones para el problema de investigación definido. También describen las herramientas a emplear en la construcción del Sistema para la Gestión de Electos SILECTOS v2.0.

1.1. Conceptos relacionados con el dominio de investigación

En este epígrafe se muestran los conceptos relacionados con el objeto de estudio:

Electos:

En la Ley Electoral se define como electos a (...) *aquellas personas nombradas Delegado o Diputado una vez finalizados los procesos electorarios. Las mismas ocuparán cargos en las diferentes Asambleas del Poder Popular y en el Consejo de Estado (...)* (ONE, 2011).

Sistema de información:

Manuel de Peralta Carrasco, en la Revista de derecho Extremadura, define sistema de información como (...) *un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio, teniendo en cuenta el equipo computacional necesario para que el sistema pueda operar y el recurso humano que interactuará con este. Es encargado de realizar cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información (...)* (Revista de derecho Extremadura, 2011).

Según Freddy Robalino, un sistema de información es (...) *un conjunto de elementos que interactúan entre sí con el fin de apoyar una empresa o negocio (...)* (Robalino, 2015).

Enrique Arenas Mendoza conceptualiza sistema de información como (...) *el sistema encargado de brindar información con atributos importantes para ayudar al proceso de toma de decisiones (...)* (Arenas, 2015).

En la investigación se asume como sistema de información el definido por Manuel de Peralta Carrasco en la Revista de derecho Extremadura debido a que es el que más se ajusta a la investigación a desarrollar.

Capítulo 1: Fundamentación teórica del Sistema para la Gestión de Efectos v2.0

Gestión de información:

En el *Harrod's Librarian's Glossary*, se conceptualiza gestión de información como (...) *un término impreciso que sirve para designar un conjunto de actividades orientadas a la generación, coordinación, almacenamiento o conservación, búsqueda y recuperación de la información tanto interna como externa contenida en cualquier soporte* (...) (Prytherch, 2000).

Chun Choo plantea que la gestión de información, (...) *tiene como objetivo optimizar la utilidad y contribución de los recursos de información para alcanzar los fines de la organización, por lo que es necesario la creación de canales y medios para transmitir o acceder a la información* (...) (Choo, 2002).

Michael W. Hill define que en la gestión de información (...) *se debe garantizar que la información esté disponible para cada persona de la organización en el momento requerido* (...) (Hill, 2005).

En la investigación se asume como gestión de información el definido por Ray Prytherch en *Harrod's Librarian's Glossary* debido a que es el que más se ajusta a la investigación a desarrollar.

Sistema de Gestión de Información:

Davis y Olson conceptualizan sistema de gestión de información como (...) *un sistema integrado y automatizado para proveer la información que sostenga las funciones de operatividad, gestión y toma de decisiones en una organización* (...) (Davis, y otros, 1984).

Moreiro González define sistema de gestión de información como (...) *el conjunto de políticas y normas relacionadas entre sí que se establecen para el acceso y tratamiento de los recursos de información, incluye los registros administrativos y los archivos, el soporte tecnológico y el público a que se destina. En su evolución el sistema puede manejar la función de inteligencia corporativa y generar productos de inteligencia* (...) (Moreiro, 1998).

En la investigación se asume como gestión de información el definido por Gordon Davis y Margretheen Olson el libro *Management information systems: conceptual foundations, structure, and development. 2nd ed*, debido a que es el que más se ajusta a la investigación a desarrollar.

Capítulo 1: Fundamentación teórica del Sistema para la Gestión de Electos v2.0

1.2. Análisis de los sistemas informáticos para los procesos electorarios cubanos

Hasta el año 2005, la Comisión Electoral Nacional de Cuba no contaba con un sistema informatizado que permitiera la gestión de la información de los procesos electorarios en el país, retrasando la realización de los análisis estadísticos. Por eso los especialistas de la empresa Desoft desarrollaron los sistemas Elena I y II, los cuales permitieron gestionar los datos de los procesos para la nominación-elección de los Delegados a la AMPP, APPP y los Diputados a la ANPP. Estos sistemas gestionaban esta información hasta el nivel municipal y excluían todos los datos referentes a los electos posterior a los procesos electorarios.

Ante esta dificultad los especialistas del Proyecto para la Informatización de los Procesos Electorarios Cubanos, del centro DATEC de la UCI, desarrollaron los Sistemas de Gestión e Información del Proceso Electoral (SIGEL Parciales y SIGEL Generales). Estos abarcan el registro de la información de los pasos electorales, desde la gestión de las autoridades hasta la constitución de las Asambleas del Poder Popular, y permiten gestionar la información hasta el nivel de circunscripción. Quedando sin gestionar la información de los electos una vez finalizados los procesos electorarios.

Para resolver el inconveniente existente se desarrolló el Sistema para la Gestión de Electos SILECTOS v1.0. Este software posibilita la gestión, la sustitución y el cubrimiento de plazas vacantes de Delegados y Diputados en el tiempo concebido entre cada proceso electorario, además de la generación de estadísticas y reportes necesarios para establecer comparaciones que serán de ayuda para la toma de decisiones.

La Tabla 1 muestra la comparación de los sistemas informáticos de apoyo a los procesos electorarios existentes en Cuba hasta la actualidad (Ver Tabla 1).

Tabla 1: Comparación de soluciones existentes

	Elena I y II	SIGEL (Parciales y Generales) v3.0	SILECTOS v1.0
Soporte	no	si	no
Formato de exportación de reportes	no	PDF, Excel	PDF, Excel
Entorno de desarrollo	Escritorio	WEB	WEB
Software libre	no	si	si
Análisis de electos posterior a los procesos electorarios	no	no	si

Capítulo 1: Fundamentación teórica del Sistema para la Gestión de Electos v2.0

El estudio de estos sistemas arrojó como conclusión que SILECTOS v1.0 es el único que realiza la gestión de los electos posterior a los procesos electorarios. A pesar de esto, en el mismo no se lleva a cabo la gestión de la información como es debido, presentando las siguientes deficiencias:

- Reemplaza un electo sustituto, aunque este no haya sido aprobado.
- Nomencladores como municipio y provincia electa se pueden modificar en los Delegados a la AMPP y a la APPP.
- La información sobre los electos puede ser modificada por diferentes usuarios.
- La activación-desactivación de los usuarios y direcciones IP se deben realizar directamente desde la base de datos.
- No cumple con la política de mínimos privilegios para la gestión de los permisos.
- No permite llevar el seguimiento y monitorización de los errores producidos desde el servidor.

Debido a estos elementos se decidió realizar la versión 2.0 de SILECTOS, aceptando los elementos de estructura y diseño de interfaz de usuario que poseen SIGEL (Parciales y Generales), con las que el cliente ya está familiarizado. Para el desarrollo de la nueva versión se tomarán las buenas prácticas de SILECTOS v1.0 como es el funcionamiento del negocio estructurado por módulos, los pasos para la realización del proceso de sustitución de los electos y la información que se obtiene en la generación de los reportes.

1.3. Herramientas y tecnologías para la construcción de SILECTOS v2.0

Para la implementación de SILECTOS v2.0 se tomarán como herramientas base, las establecidas por el Proyecto para la Informatización de los Procesos Eleccionarios Cubanos en el desarrollo de los sistemas informáticos para la CEN. Las mismas son: el lenguaje de modelado UML v2.5, la herramienta CASE Visual Paradigm para UML v8.0, los marcos de trabajo Symfony v2.7 y ExtJS v4.2, los lenguajes de programación PHP v5.5 y JavaScript, el sistema gestor de base de datos PostgreSQL v9.4, la herramienta externa JasperReports v6.1, el contenedor web con soporte de servlets Tomcat v7.0 y el servidor web Apache v2.2. Estas herramientas están definidas en el documento 0120_Arquitectura de Software Proyecto para la Informatización de los Procesos Eleccionarios Cubanos, Sección 2: Decisiones y fundamentación de la selección de los componentes de producto. También se hará uso de la metodología AUP en su variante AUP_UCI.

AUP_UCI

Es una variante metodológica para la actividad productiva de la UCI, basada en la metodología Proceso Unificado Ágil (AUP, del inglés *Agile Unified Process*) en unión con el modelo CMMI-DEV v1.3. Aplica

Capítulo 1: Fundamentación teórica del Sistema para la Gestión de Efectos v2.0

técnicas ágiles entre las que se incluye el modelado ágil, el desarrollo dirigido por pruebas y la gestión de cambios ágil (Rodríguez, 2014). Se escoge esta metodología debido a los siguientes elementos que se ajustan a la presente investigación:

- Fue desarrollada para proyectos productivos de la universidad.
- Es apropiado para proyectos pequeños y de bajos recursos, lo que permite incrementar las probabilidades de éxito.
- Tiene un enfoque centrado al cliente con iteraciones cortas.
- Evita la elaboración de documentos, diagramas e iteraciones innecesarias requeridas en otras metodologías.
- Permite detectar errores tempranos a través de los ciclos iterativos.

Para el desarrollo del sistema SILECTOS v2.0 serán adoptadas las disciplinas: Modelado de negocio en la variante 3-Modelo Conceptual (MC), Requisitos en el Escenario No. 2, Análisis y diseño, Implementación y Pruebas internas.

UML v2.5

El Lenguaje de Modelado Unificado v2.5 (UML, del inglés *Unified Modeling Language*), es un lenguaje estándar para el modelado de sistemas de software. Se centra en: la representación gráfica de un sistema orientado a objetos, la especificación de las características del sistema antes de su construcción, la implementación de los sistemas diseñados a partir de los modelos especificados y la documentación de los elementos gráficos del sistema desarrollado que puedan servir para su futura revisión (OMG, 2015). El empleo de este lenguaje se debe a las ventajas que posee para dirigir las tareas y orientar el orden de ejecución de las actividades a desarrollar y se escoge esta versión por ser la más actualizada

Visual Paradigm para UML v8.0

Es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software. Es multiplataforma, utiliza UML como lenguaje de modelado, cuenta con una versión libre para la comunidad y apoya la construcción de aplicaciones con mayor calidad. Permite diseñar varios tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Esta herramienta también proporciona una interfaz gráfica de usuario y soporta la revisión ortográfica, ofreciendo sugerencias para diversos idiomas (Paradigm, 2013). Se escoge esta herramienta porque facilita representación gráfica y documentación del sistema; además, especifica las características y constituye una guía para el proceso de desarrollo del software en todas las fases de construcción.

Capítulo 1: Fundamentación teórica del Sistema para la Gestión de Electos v2.0

NetBeans IDE v8.0

Es un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés *Integrated Development Environment*) o entorno de programación que ha sido empaquetado como un programa de aplicación. Está escrito en *Java*, es de código abierto y gratuito para desarrolladores de software. Además, es multiplataforma y fácil de instalar (Jackson, 2014). Se escoge esta herramienta ya que ofrece las características necesarias para la creación de aplicaciones web en los lenguajes de programación PHP o *JavaScript*.

ExtJS v4.2

Es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas que flexibiliza la estructura jerárquica de objetos del navegador web como el DOM (del inglés *Document Object Model*), las peticiones AJAX (del inglés *Asynchronous JavaScript And XML*) y el diseño dinámico de las mismas (DHTML, del inglés *Dinamic HyperText Markup Language*). Tiene la gran funcionalidad de crear interfaces de usuario bastante funcionales y la capacidad de desarrollar aplicaciones web con estándares, patrones y arquitecturas como es el caso de la Programación Orientada a Objetos (POO) y el patrón arquitectónico Modelo-Vista-Controlador (ExtJS, 2013).

JavaScript

Es un lenguaje de programación multiplataforma y orientado a objetos. Contiene una biblioteca estándar de objetos, tales como *Array*, *Date*, y *Math*, y un conjunto central de elementos del lenguaje tales como operadores, estructuras de control, y sentencias (MDN, 2015). Se escoge este lenguaje de programación debido a que ExtJS v4.2 lo emplea para la construcción de las clases del lado del cliente.

Symfony v2.7

Es un marco de trabajo diseñado para optimizar el desarrollo de aplicaciones web. Emplea el patrón arquitectónico Modelo-Vista-Controlador que le permite separar la lógica del negocio, la lógica del servidor y la presentación de la información. Es desarrollado completamente con PHP utilizando las versiones 5.3 o superiores y es compatible con la mayoría de los sistemas gestores de base de datos (Symfony, 2014). Esta versión incorpora características como la caché HTTP que mejora la carga automática de páginas del sistema. Además, el manejo de rutas es más fácil y más rápida ya que agrega los *namespace* que permiten la creación de diferentes clases con igual nombre en diferentes directorios, por lo que ruta de acceso de archivos de clases es única. También posee un contenedor de inyecciones de dependencias para el manejo de los plugins, un motor de plantillas Twig que permite tener separada la lógica del negocio de los datos que serán mostrados al usuario.

Capítulo 1: Fundamentación teórica del Sistema para la Gestión de Electos v2.0

PHP v5.5

Es un lenguaje de propósito general y de código abierto creado para el desarrollo web, que puede ser embebido en páginas HTML. El objetivo principal de este lenguaje es permitir a los desarrolladores web escribir de forma rápida y dinámica páginas web (Kaneiwa, y otros, 2015). Ha sido publicado siempre como software libre, con una licencia llamada Licencia PHP. Es utilizado como módulo de servidores web como es el caso de *Apache*, lo que lo hace extremadamente veloz. Es completamente expandible, está compuesto por un conjunto de módulos y una variedad de extensiones de código (PHP, 2011). Su uso se debe principalmente, por ser el lenguaje de programación empleado en la implementación del marco de trabajo *Symfony v2.7*.

YAML

Es un formato de serialización de datos legible por humanos inspirado en lenguajes como XML, C, Python. Fue creado bajo la creencia de que todos los datos pueden ser representados adecuadamente como combinaciones de listas, mapeos y datos simples. La sintaxis es relativamente sencilla y fue diseñada teniendo en cuenta que fuera muy legible pero que, a la vez, fuese fácilmente mapeable a los tipos de datos más comunes en la mayoría de los lenguajes de alto nivel (YAML, 2011). Se escoge este formato para la definición de los archivos de configuración.

PostgreSQL v9.4

Es un Sistema Gestor de Base de Datos (SGBD) que posee gran escalabilidad permitiendo ajustarse al número de computadoras y a la cantidad de memoria que posee el sistema de forma óptima, soportando una mayor cantidad de peticiones simultáneas de manera correcta. Emplea un modelo cliente-servidor y utiliza multiprocesos en lugar de multihilos para garantizar la estabilidad del sistema, permitiendo que el fallo de uno de los procesos no afecte al resto y que el sistema continúe funcionando (PostgreSQL, 2014).

PgAdmin III

Es un cliente de administración para base de datos diseñado para responder a las necesidades de los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. Es una herramienta de código abierto para la administración de bases de datos *PostgreSQL* y derivados (*EnterpriseDB*, *PostgresPlus*, *AdvancedServer* y *GreenplumDatabase*). Está disponible en más de una docena de lenguajes y para varios sistemas operativos como son *Microsoft Windows*, *Linux*, *FreeBSD*, *Mac OSX* y *Solaris* (Michaud, 2015). Se escoge esta herramienta porque su interfaz gráfica soporta todas las características de *PostgreSQL* y hace simple su administración.

Capítulo 1: Fundamentación teórica del Sistema para la Gestión de *Electos v2.0*

JasperReports v6.1

Es una librería de código abierto para el desarrollo de informes con Java sobre diferentes tipos de fuentes de datos. Emplea el motor de reportes JasperReport y el diseñador de plantillas IReport, y posee un mayor número de elementos para el diseño, de operaciones sobre las variables incluyendo el cálculo con funciones de Java, y permite trabajar con varios elementos al mismo tiempo. Otras características significativas que provee son (Jaspersoft, 2014):

- El software es libre pues es distribuido mundialmente bajo los términos de la Licencia Pública para Bibliotecas GNU (del inglés *Library Public License*) y está respaldado por una gran comunidad internacional de desarrollo.
- Los reportes son capaces de presentar los datos de manera textual o a través de gráficos. Además de mostrar los datos deseados, pueden generar o calcular nuevos datos de forma dinámica y mostrarlos.
- Permite un diseño flexible de los reportes: Los reportes se pueden dividir en secciones opcionales que son: título del reporte, el encabezado de página, una sección para los detalles del reporte, el pie de página y una sección de resumen que aparece al final del reporte.

Tomcat v7.0

Es la implementación de referencia oficial para las especificaciones *Servlet* y *Java Server Pages* (JSP). Es un producto muy robusto, altamente eficiente y uno de los más potentes contenedores de *Servlets* existentes. Su punto débil reside en la complejidad de su configuración, dado el gran número de opciones existente (Mateu, 2004). Se escoge esta herramienta ya que puede utilizarse como un servidor de aplicaciones web con HTML, *Servlets* y JSPs, o como complemento al servidor Apache.

Apache Web Server v2.2

Es un servidor web desarrollado por la Fundación Apache Software (ASF, del inglés *Apache Software Foundation*), de código abierto y con licencia pública general. Sigue una arquitectura cliente-servidor (Astelehena, 2008). Se escoge esta versión porque tiene las siguientes características:

- Estable: impide caídas o cambios inesperados en el servidor.
- Flexible: permite trabajar con la mayor parte de las extensiones web que existen en la actualidad.
- Multiplataforma: está disponible para diferentes plataformas como GNU/Linux, Windows, MacOS.

Capítulo 1: Fundamentación teórica del Sistema para la Gestión de Electos v2.0

1.4. Conclusiones del capítulo

Luego de finalizado este capítulo, se concluye que fueron abordados los principales conceptos relacionados con los sistemas informáticos, y se analizaron los softwares existentes para el apoyo a los procesos eleccionarios en Cuba. Esto permitió deducir la necesidad de crear un sistema para gestionar la información de los electos posteriores a los procesos eleccionarios. La metodología seleccionada: AUP-UCI desarrollada para proyectos productivos de la universidad, comprende todas las fases de la construcción de un sistema, estableciendo disciplinas y escenarios para su desarrollo. El Visual Paradigm v8.0 como herramienta CASE permitirá elaborar los diagramas necesarios de la solución, empleando como lenguaje de modelado UML v2.5. Las herramientas NetBeansIDE v8.0, Symfony v2.7, ExtJS v4.2 y los lenguajes de programación PHP v5.5 y JavaScript, posibilitarán la implementación de la aplicación web. El uso de PostgreSQL v9.4 como SGBD y el PgAdmin III como herramienta de administración de los datos, permiten la administración y el soporte de las estructuras físicas para brindar un correcto almacenamiento de la información. El servidor de aplicaciones web Apache v2.2 responderá a peticiones del navegador vía https.

Capítulo 2: Análisis y diseño del sistema SILECTOS v2.0

En el presente capítulo se describe la propuesta de solución para el problema de investigación mediante el modelo conceptual como representación visual de las clases conceptuales del dominio de interés, así como la descripción de estas. Además, se especifican los requisitos funcionales y no funcionales como características imprescindibles que debe cumplir el software, y su diagramación a través de Casos de Uso del sistema contribuyendo al cumplimiento del objetivo general de la investigación, empleando para esto los patrones de Casos de Uso. También se plasma el diagrama de paquetes y el de clases del diseño que siguen el patrón arquitectónico Modelo-Vista-Controlador, al igual que el diagrama de secuencias para ejemplificar un flujo de información y el diagrama entidad-relación con las tablas que contiene la base de datos. Por último, se visualiza el diagrama de despliegue donde se describe el ambiente dentro del cual el sistema será desplegado.

2.1. Modelo del dominio

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés, por lo que se considera el artefacto más importante que se crea durante el análisis orientado a objetos. Su objetivo es describir clases u objetos de software con responsabilidades. Puede ser tomado como punto de partida para el diseño de los sistemas ya que supone el funcionamiento interno donde el software imita la realidad en alguna medida (Revista Ingenierías Universidad de Medellín, 2010). Es por esto que el modelo conceptual constituya una primera versión del sistema A continuación, se muestra el modelo de dominio para SILECTOS v2.0 (Ver Fig. 1) y la descripción de las clases conceptuales (Ver Tabla 2).

Diagrama conceptual del Modelo de Dominio

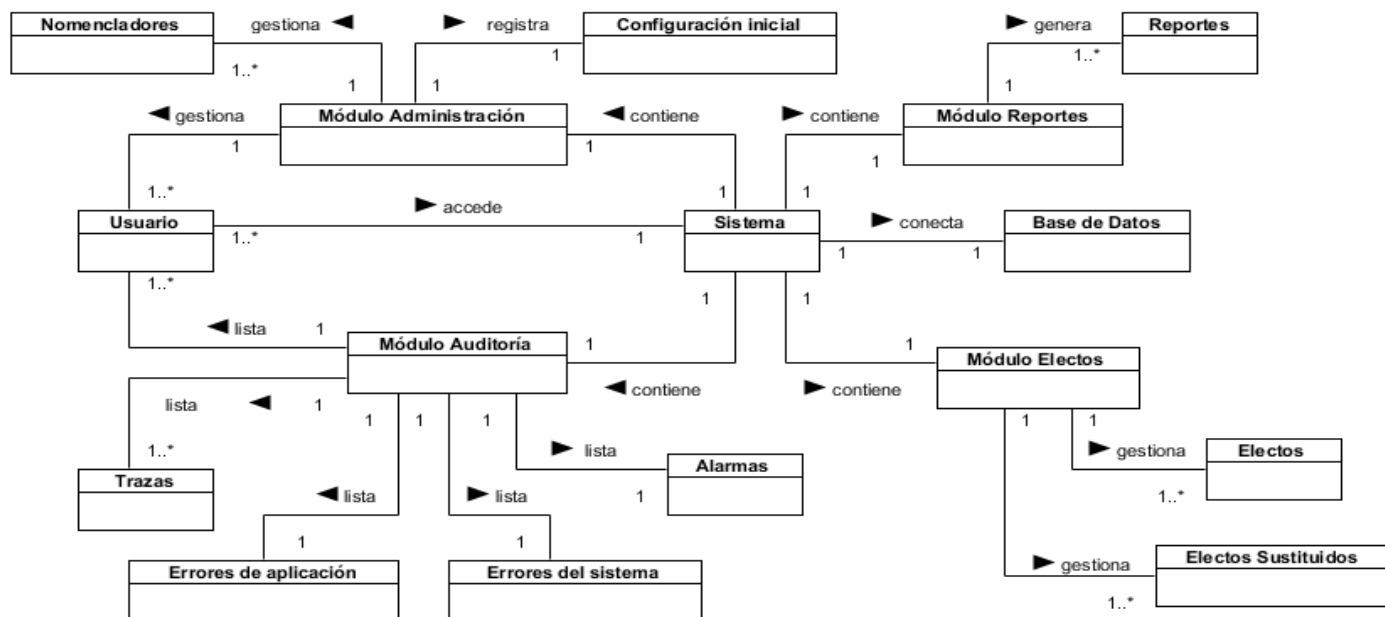


Fig. 1: Modelo de dominio para SILECTOS v2.0

Definición de clases del modelo de dominio

Tabla 2: Definición de las clases del modelo de dominio

Clases	Descripción
Usuario	Persona que accede al sistema para realizar alguna actividad en el mismo. Estos son: administrador, auditor, usuario municipal, usuario provincial o usuario nacional.
Sistema	Software desarrollado para llevar a cabo la gestión de los electos. Está compuesto por los módulos Administración, Auditoría, Electos y Reportes. Se encuentra conectado a una base de datos para el almacenamiento de la información.
Módulo Administración	Módulo del sistema encargado de registrar los parámetros configurables, así como de gestionar los usuarios y los nomencladores. Es accedido por el usuario con rol administrador.
Nomencladores	Lista de atributos que caracterizan a cada usuario del sistema. Estos son: noticia, nivel de escolaridad, organización política o de masas, ocupación laboral, color de piel, país, provincia, municipio, bloquear/desbloquear IP, Sesiones activas.
Configuración inicial	Inserción de los datos necesarios para registrar la configuración del sistema. Estos datos son: legislatura, mandato, cantidad máxima de IP por usuario, cantidad máxima de intentos de conexión fallidos, tiempo máximo para cambiar la contraseña (días) y tiempo máximo para cerrar la sesión (minutos).

Capítulo 2: Análisis y diseño del sistema SILECTOS v2.0

Módulo Reportes	Módulo del sistema encargado de generar informes sobre los electos y los electos sustituidos, en los niveles municipal, provincial y nacional. Es accedido por los usuarios que tienen asignado el rol visor de reportes.
Reporte	Informes del sistema sobre los Delegados a las AMPP, APPP y Diputados a la ANPP, así como de las sustituciones, necesarios para el proceso de toma de decisiones.
Módulo Electos	Módulo del sistema encargado de llevar a cabo la gestión de los electos y de los electos sustituidos. Es accedido por los usuarios que tienen asignado el rol municipal, provincial o nacional.
Electo	Delegados y Diputados que fueron electos en las AMPP, APPP, ANPP y se encuentran registrados en el sistema.
Electo Sustituido	Delegados de las AMPP, APPP y Diputados de las ANPP registrados en el sistema, que han sido sustituidos por revocaciones o liberaciones.
Módulo Trazas	Módulo del sistema encargado de visualizar las trazas, las alarmas, los errores del sistema, los errores de la aplicación y los usuarios registrados en el mismo. Es accedido por aquel usuario que tenga asignado el rol auditor.
Alarmas	Cada alarma muestra la siguiente información: alerta, usuario, IP cliente, fecha, acción y recurso.
Errores del sistema	Cada error muestra la siguiente información: tipo de error, IP cliente, mensaje, IP servidor, fecha y hora.
Errores de la aplicación	Cada error muestra la siguiente información: IP cliente, usuario, fecha, hora, tipo y mensaje.
Traza	Secuencia de acciones que han sido ejecutadas por un usuario en el sistema. Cada traza muestra la siguiente información: nombre y apellidos, usuario, rol, IP cliente, provincia, municipio, fecha, hora, acción, tipo de traza, recurso, agente y observaciones.
Base de datos	Espacio de memoria en forma de tabla, destinada a almacenar toda la información manejada por el sistema. Es desarrollada a partir del empleo de un SGBD.

Especificación de los requisitos del sistema

La especificación de requisitos de software es uno de los flujos de trabajo más importantes, ya que en él se establece qué debe hacer exactamente el sistema a desarrollar. Los requisitos son como un contrato donde se debe cumplir todas las condiciones que se definan, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que se especifiquen (Wesley, 1999). Tienen como principal objetivo guiar el desarrollo hacia el sistema deseado y este propósito es conseguido por medio de requisitos funcionales y no funcionales.

Requisitos funcionales

Los requisitos funcionales (RF) son declaraciones de los servicios que debe proporcionar un sistema de manera en que este reaccione a determinadas entradas y su comportamiento sea el esperado (Sommerville, 2005). Es decir, representan las funcionalidades del sistema.

SILECTOS v2.0 cuenta con un total de 255 requisitos funcionales que se describen en el documento Especificación de Requisitos de Software SILECTOS v2.0, perteneciente al expediente de proyecto de dicho sistema. En el mismo se listan y describen las funcionalidades que debe cumplir el sistema. A continuación, se detallan algunos de estos:

RF1. Activar usuario

Descripción: El sistema debe permitirle al usuario con rol administrador la posibilidad de activar un grupo, subconjunto o todos los usuarios.

Salida: Usuario o listado de usuarios activos.

RF2. Desactivar usuario

Descripción: El sistema debe permitirle al usuario con rol administrador la posibilidad de desactivar un grupo, subconjunto o todos los usuarios.

Salida: Usuario o listado de usuarios desactivados.

RF3. Activar dirección IP

Descripción: El sistema debe permitirle al usuario con rol administrador la posibilidad de activar un grupo, subconjunto o todas las direcciones IP.

Salida: Dirección o listado de direcciones IP activos.

RF4. Desactivar dirección IP

Descripción: El sistema debe permitirle al usuario con rol administrador la posibilidad de desactivar un grupo, subconjunto o todas las direcciones IP.

Salida: Dirección o listado de direcciones IP desactivados.

RF5. Listar errores del servidor.

Descripción: El sistema debe permitir al usuario con rol auditor listar detalles de los errores del servidor, mostrándose los campos: Tipo de error, IP Cliente, Mensaje, IP Servidor, Fecha, Hora.

Salida: Listado de los errores del servidor.

RF6. Visualizar detalles de los errores del servidor.

Descripción: El sistema debe permitir al usuario con rol auditor visualizar detalles de un error previamente seleccionado, mostrándose los campos: Tipo de error, IP Cliente, Mensaje, IP Servidor, Fecha, Hora.

Salida: Detalles del error del servidor previamente seleccionado, mostrándose los datos Tipo de error, IP Cliente, Mensaje, IP Servidor, Fecha, Hora.

RF7. Buscar errores del servidor.

Descripción: El sistema debe permitir al usuario con rol auditor realizar una búsqueda avanzada (filtra la búsqueda a partir de, al menos, uno de los siguientes campos: Tipo de error, IP Cliente, Mensaje, IP Servidor, Fecha, Hora).

Entrada: Datos por los que se desea filtrar la búsqueda en el listado de errores del servidor.

Salida: Listado de errores del servidor que coinciden con los datos especificados

Requisitos no funcionales

Los requerimientos no funcionales (RNF) son aquellos que no se refieren directamente a las funciones específicas que proporciona la herramienta, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y capacidad de almacenamiento (Sommerville, 2005). Es decir, representan aquellos atributos no funcionales que debe poseer la herramienta.

Usabilidad

RNF 1. La herramienta debe ser WEB, pero con características muy similares a las aplicaciones de escritorio en cuanto al diseño de las interfaces visuales.

RNF 2. Los formularios del sistema deben permitir que al presionarse la tecla ENTER, ejecute la funcionalidad de guardar los datos una vez introducidos.

Eficiencia

RNF 3. El sistema debe tener ofuscada la capa de presentación con el objetivo de garantizar un mejor rendimiento.

RNF 4. El sistema debe tener un mecanismo de carga de baja demanda con el objetivo de reducir la cantidad de archivos con extensiones *.js que se descargan para el cliente.

RNF 5. El sistema debe dar respuesta a las peticiones de los usuarios en un tiempo menor a los 2 segundos.

Restricción de diseño

Para la implementación del sistema se requiere del uso de las siguientes herramientas:

RNF 6. Lenguaje y marco de trabajo para el desarrollo del sistema del lado del servidor:

- Lenguaje de programación PHP v5.5.
- Marco de trabajo Symfony v2.7.

RNF 7. Lenguaje y marco de trabajo para el desarrollo del sistema del lado del cliente:

- Lenguaje de programación JavaScript.
- Marco de trabajo ExtJS v4.2

Componentes adquiridos

RNF 8. El componente adquirido para integrar sus servicios con el sistema es *JasperReport v6.1*. Debe estar instalado como una restricción para su uso: Java v7, Tomcat v7.

Interfaz de usuario

RNF 9. Las interfaces de usuario serán diseñadas a modo de aplicaciones RIA (del inglés, *Rich Internet Application*) lo que permite a los usuarios contar con aplicaciones web con una experiencia de usuario similar a la de las aplicaciones de escritorios. Para lograr este fin se empleará el lenguaje *JavaScript*.

Interfaz de hardware

RNF 10. El servidor donde se va a desplegar la aplicación tiene las siguientes características:

- Memoria RAM: 4 GB.
- Procesador: Cuatro procesadores de 2.13GHz.
- HDD: Un disco de 20 GB.

Interfaz de software

RNF 11. La PC Servidor donde se instale SILECTOS v2.0 debe contener los siguientes elementos:

- Servidor de aplicaciones Apache v2.2.
- Lenguaje de programación PHP v5.5 y las bibliotecas *php5-fpm*, *php5-gd*, *mcrypt*, *php-apc*, *php5-pgsql*, *php5-curl*, *php5-mcrypt*, *php5-xsl*.
- Sistema Gestor de Base de Datos PostgreSQL v9.4.

RNF 12. En las computadoras clientes se utilizará el navegador web Mozilla Firefox en las versiones comprendidas entre la 46 y 53.

Estándares aplicables

RNF 13. YAML: El sistema hará uso de YAML como formato estándar para ficheros de configuración.

Modelo de Caso de Uso del Sistema

El modelo de Casos de Uso describe la funcionalidad propuesta del nuevo sistema, compuesto por actores, Casos de Uso y la relación que existe entre estos. Un caso de uso representa una unidad discreta de

interacción entre un usuario y el sistema, es una unidad simple de trabajo significativo (Sparx, 2007).

Diagrama de Casos de Uso del Sistema

Un Diagrama de Casos de Uso del Sistema (CUS) es un dibujo geométrico que se utiliza para representar de manera gráfica el comportamiento de un sistema desde el punto de vista del usuario. Es una forma de agrupamiento de los requisitos funcionales del software para representar las actividades que el sistema puede ejecutar. A continuación, se muestra el diagrama de Casos de Uso del Sistema correspondiente a SILECTOS v2.0 (Ver Fig. 2) y la descripción de los actores (Ver Tabla 3).

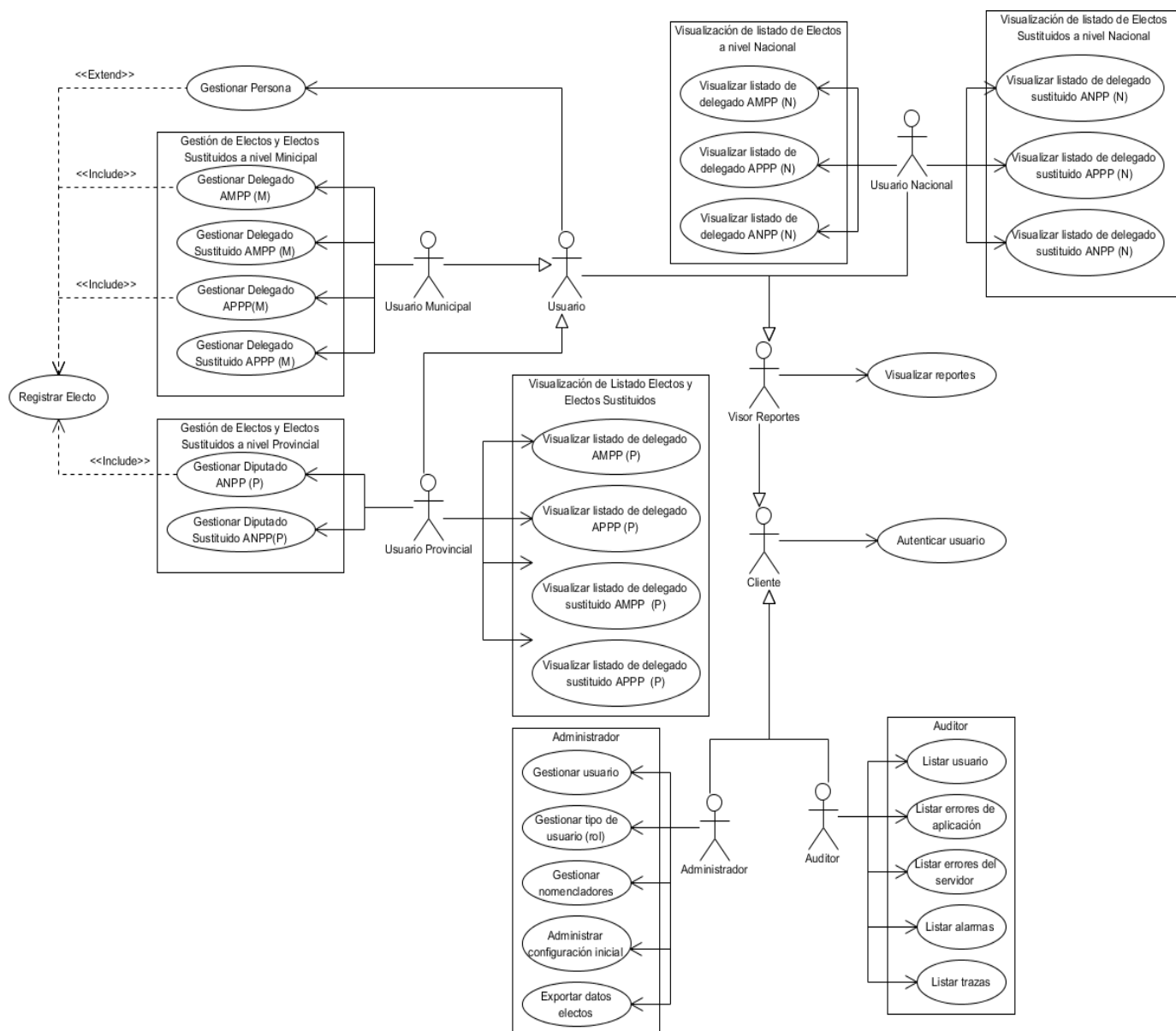


Fig. 2: Diagrama de Casos de Uso del Sistema

Descripción de los actores del sistema

Tabla 3: Descripción de los actores del sistema SILECTOS v2.0

Actor	Descripción
Usuario Municipal	Inicia las funcionalidades relacionadas con la gestión de los Delegados y de los Delegados Sustituídos en la AMPP y la APPP a nivel municipal.
Usuario Provincial	Inicia las funcionalidades relacionadas a la Visualización de listados de los Delegados y los Delegados Sustituídos en la AMPP y la APPP, además de la gestión de los Diputados y los Diputados Sustituídos en la ANPP a nivel provincial.
Usuario	Actor genérico que inicia las funcionalidades de gestionar personas, común para los actores Usuario Municipal y Usuario Provincial.
Usuario Nacional	Inicia las funcionalidades relacionadas a la visualización de los listados de electos y la visualización de los listados de electos sustituidos en las AMPP, las APPP y la ANPP a nivel nacional.
Visor Reportes	Inicia las funcionalidades de visualización de reportes, común para los actores: Usuario Municipal y Usuario Nacional. (Se visualizan solo los reportes del nivel indicado o inferior, si es Usuario Municipal, reportes municipales; si es Usuario Provincial, reportes municipales y provinciales; si es Usuario Nacional, reportes municipales, provinciales y nacionales).
Administrador	Administra el sistema gestionando usuarios, tipo de usuarios (roles) y nomencladores.
Auditor	Lleva el control de las trazas del sistema.
Cliente	Actor genérico que inicia la funcionalidad de Autenticar usuario, común para todos los actores.

Patrones de Casos de Uso utilizados

Los patrones de Casos de Uso son comportamientos que deben existir en el sistema ayudando a describir lo que el sistema debe hacer, es decir, describen el uso del sistema y cómo este interactúa con los usuarios. Estos patrones son utilizados generalmente como plantillas que describen cómo deberían ser estructurados y organizados los Casos de Uso. Son patrones que capturan mejores prácticas para modelar Casos de Uso (Sommerville, 2005). A continuación, se describen los patrones de Casos de Uso utilizados:

El patrón **Múltiples Actores - Rol Común** se utiliza cuando dos actores (Usuario Municipal y Usuario Provincial) juegan el mismo rol sobre un caso de uso (CU Gestionar persona). Este rol es representado por otro actor (*Usuario*), heredado por los actores que comparten este rol (Ver Fig. 3).^o

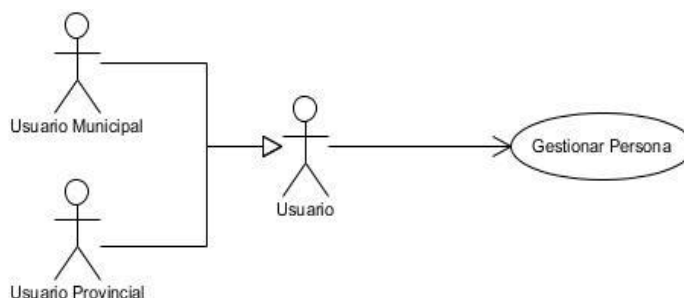


Fig. 3: Patrón Múltiples Actores - Rol Común del diagrama de CUS

El patrón **Extensión Concreta por Extensión** consiste en una relación de un caso de uso de extensión (*CU Gestionar Persona*) a un caso de uso base (*Registrar Electo*), que especifica cómo el comportamiento definido por el caso de uso de extensión (listar, modificar, eliminar y visualizar electos) puede insertarse dentro del comportamiento definido por el caso de uso base (adicionar electo) sin expresar cambios en este último (Ver Fig. 4).

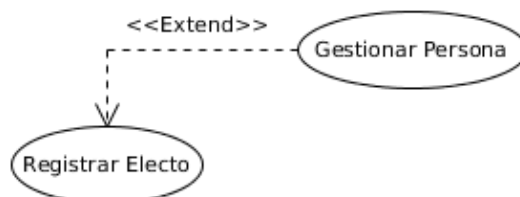


Fig. 4: Patrón Extensión Concreta por Extensión del diagrama de CUS

El patrón de **Inclusión Concreta por Inclusión** es muy similar al patrón *Extensión Abstracta*, diferenciándose en que sí provoca cambios en el caso de uso original (Jacobson, 2004). Dicho patrón se evidencia en el caso de uso base Gestionar Diputado a la ANPP que incluye el CU Registrar Electo para el proceso de sustitución de los Diputados cuando la persona sustituta no se encuentra en el listado. Esto implica que cualquier cambio en el caso de uso incluido provoca cambios en el caso de uso base (Ver Fig. 5).

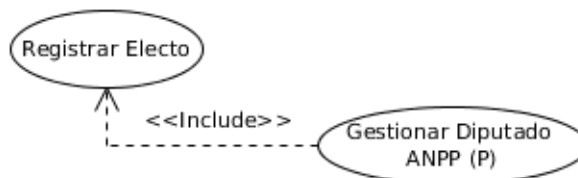


Fig. 5: Patrón Inclusión Concreta por Inclusión del diagrama de CUS

Descripción textual de los Casos de Uso del sistema

Un caso de uso también se define como el conjunto de escenarios que tienen una meta de usuario en común, permitiendo definir los límites del sistema y las relaciones usuario-sistema. Los escenarios no son más que la secuencia de acciones e interacciones entre los usuarios y el sistema (Gutierrez, 2011). La Tabla

Capítulo 2: Análisis y diseño del sistema SILECTOS v2.0

4 muestra la descripción textual del caso de uso Gestionar Delegados a la AMPP (M) (Ver Tabla 4).

Tabla 4: Descripción textual del caso de uso Gestionar Delegados a la AMPP (M).

Objetivo	Listar, Modificar, Sustituir, Visualizar Detalles y Buscar Delegados a la AMPP (M).	
Actores	Usuario municipal (Inicia)	
Resumen	El caso de uso inicia cuando el usuario municipal selecciona la opción Gestionar Delegados a la AMPP (M) ubicada en el menú izquierdo dentro de la pestaña Electos, y finaliza cuando se modifica, sustituye, visualizan los detalles o se buscan Delegados a la AMPP a nivel municipal.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	El usuario municipal ha sido autenticado con los permisos necesarios.	
Postcondiciones	El usuario municipal ha sustituido, modificado, visualizado los detalles o buscado a los Delegados a la AMPP a nivel municipal.	
Flujo de eventos		
Flujo básico: “Gestionar Delegados AMPP (M)”		
	Actor	Sistema
1.	El usuario municipal modifica, sustituye, visualiza los detalles o realiza búsqueda avanzada sobre los Delegados a la AMPP del municipio al que pertenece.	
2.		<p>Puede realizar varias acciones con los Delegados a la AMPP del municipio al que pertenece el usuario municipal:</p> <p>Modificar un Delegado a la AMPP en el sistema. Ver Sección 1: “Modificar Delegados a la AMPP (M)”.</p> <p>Sustituir un Delegado a la AMPP en el sistema. Ver Sección 2: “Sustituir Delegados a la AMPP (M)”.</p> <p>Visualizar detalles de un Delegado a la AMPP en el sistema. Ver Sección 3: “Detalles de Delegado a la AMPP (M)”.</p> <p>Buscar Delegados a la AMPP en el sistema. Ver Sección 4: “Buscar Delegados a la AMPP (M)”.</p>
3.		Finaliza el Caso de Uso.

Capítulo 2: Análisis y diseño del sistema SILECTOS v2.0

Sección 1: “Modificar Delegado a la AMPP (M)”		
Flujo básico: “Modificar Delegado a la AMPP (M)”		
	Actor	Sistema
1.	El usuario municipal accede a través del menú a la opción <i>Gestionar Delegados a la AMPP (M)</i> .	
2.		El sistema muestra una ventana con el listado de los Delegados a la AMPP del municipio al que pertenece el usuario municipal.
3.	El usuario municipal selecciona del listado de Delegados aquel que se necesita modificar y se encuentre en el estado <i>En Elaboración</i> .	
4.	El usuario municipal presiona el botón <i>Modificar</i> .	
5.		El sistema muestra la interfaz <i>Modificar Delegado a la AMPP</i> .
6.	El usuario municipal llena los campos que desea modificar.	
7.	El usuario municipal presiona el botón <i>Aceptar</i> .	
8.		El sistema verifica que no se encuentren campos vacíos.
9.		El sistema valida los datos introducidos.
10.		El sistema modifica los datos necesarios, excepto el <i>Número de identidad</i> que aparecerá deshabilitado.
11.		El sistema muestra un mensaje de confirmación.
12.		Finaliza el Caso de Uso.
Flujos alternos		
Nº 8: “Campos vacíos”		
	Actor	Sistema
8.		El sistema bordea el campo vacío de color rojo y muestra un mensaje de alerta indicando que existen campos en blanco.
Nº 9: “Datos no válidos”		
	Actor	Sistema
9.		El sistema bordea el campo con Datos no válidos de color rojo y muestra un mensaje de alerta indicando que existen campos no válidos.

Prototipo elemental de interfaz gráfica de usuario

Modificar delegado a la AMPP

Primer nombre: <input style="width: 90%;" type="text"/> Segundo nombre: <input style="width: 90%;" type="text"/> Primer apellido: <input style="width: 90%;" type="text"/> Segundo apellido: <input style="width: 90%;" type="text"/> Entidad laboral o centro de estudio: <input style="width: 90%;" type="text"/> Sexo: <input style="width: 90%;" type="text"/>	Nivel de escolaridad: <input style="width: 90%;" type="text"/> Ocupación: <input style="width: 90%;" type="text"/> Color de la piel: <input style="width: 90%;" type="text"/> Número de identidad: <input style="width: 90%;" type="text"/>	Organizaciones políticas y de masas: <input type="checkbox"/> CDR <input type="checkbox"/> ANAP <input type="checkbox"/> UJC <input type="checkbox"/> FEU <input type="checkbox"/> FEEM <input type="checkbox"/> PCC <input type="checkbox"/> FMC <input type="checkbox"/> CTC <input type="checkbox"/> ACRC <input type="checkbox"/> No necesario
<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>		

Sección 2: “Sustituir Delegado a la AMPP”

Flujo básico: “Sustituir Delegado a la AMPP”

1.	El usuario municipal accede a través del menú a la opción <i>Gestionar Delegados a la AMPP</i> .	
2.		El sistema muestra una ventana con el listado de los Delegados a la AMPP del municipio al que pertenece el usuario municipal.
3.	El usuario municipal selecciona del listado de Delegados aquel que se necesita sustituir y se encuentre en estado <i>Aprobado</i> .	
4.	El usuario municipal presiona el botón <i>Sustituir</i> .	
5.		El sistema muestra la interfaz <i>Sustituir Delegado a la AMPP</i> .
6.	El usuario municipal registra los campos y marca el campo plaza vacante si no hay ningún sustituto para esa persona.	
7.		El sistema muestra el campo <i>Fecha de plaza vacante</i> con la fecha actual, sin dejar que el usuario lo pueda modificar.
8.	El usuario municipal presiona el botón <i>Aceptar</i> .	
9.		El sistema verifica que no se encuentren campos vacíos.
10.		El sistema valida los datos introducidos.
11.		El sistema muestra un mensaje de confirmación.
12.		El sistema muestra el listado de electos actualizado

Capítulo 2: Análisis y diseño del sistema SILECTOS v2.0

		siendo el estado de proceso Plaza vacante.
13.		Finaliza el Caso de Uso.
Flujos alternos		
Nº 6.1: “Plaza no vacante”		
	Actor	Sistema
6.	El usuario municipal registra los campos y no marca el campo plaza vacante si hay un sustituto para esa persona.	
7.		El sistema verifica que no se encuentren campos vacíos.
8.		El sistema valida los datos introducidos.
9.		El sistema muestra una interfaz <i>Seleccionar nueva persona</i> con el listado de electores del municipio al cual pertenece el usuario municipal.
10.	El usuario municipal selecciona el nuevo Delegado a la AMPP.	
11.		El sistema indica si la persona seleccionada fue Delegado Municipal, Delegado Provincial o Diputado en el proceso anterior.
12.	El usuario municipal presiona el botón <i>Aceptar</i> .	
14.		El sistema muestra un mensaje de confirmación.
15.		El sistema muestra el listado de electos actualizado, siendo el estado de proceso del electo sustituido En Elaboración.
Nº 6.2: “Estado de proceso Plaza vacante”		
	Actor	Sistema
1.	El usuario municipal selecciona del listado de Delegados, aquel que se necesita sustituir y se encuentre en estado <i>Plaza vacante</i> .	
2.		El sistema muestra una interfaz <i>Seleccionar nueva persona</i> con el listado de electores del municipio al cual pertenece el usuario municipal.

Capítulo 2: Análisis y diseño del sistema SILECTOS v2.0

3.	El usuario municipal selecciona la propuesta del nuevo Delegado a la AMPP.	
4.		El sistema indica si la persona seleccionada fue Delegado Municipal, Delegado Provincial o Diputado en el proceso anterior.
5.	El usuario municipal presiona el botón <i>Aceptar</i> .	
6.		El sistema muestra un mensaje de confirmación.
7.		El sistema muestra el listado de electos actualizado, siendo el estado de proceso del electo sustituido En Elaboración.
8.		Finaliza el Caso de Uso.
Nº 6.3: "Insertar persona sustituta"		
	Actor	Sistema
6.	El usuario municipal registra los campos y no marca el campo plaza vacante si hay un sustituto para esa persona.	
7.		El sistema verifica que no se encuentren campos vacíos.
8.		El sistema valida los datos introducidos.
9.		El sistema muestra una interfaz <i>Seleccionar nueva persona</i> con el listado de electores del municipio al cual pertenece el usuario municipal.
10.	El usuario municipal presiona el botón <i>Adicionar</i> .	
11.		El sistema muestra la ventana <i>Adicionar persona</i> .
12.	El usuario inserta los datos del nuevo electo.	
13.	El usuario municipal presiona el botón <i>Aceptar</i> .	
14.		El sistema verifica que no se encuentren campos vacíos.
15.		El sistema valida los datos introducidos.
16.		El sistema muestra un mensaje de confirmación.
17.		El sistema muestra el listado de electos actualizado,

		siendo el estado de proceso del electo sustituido En Elaboración.
18.		Finaliza el Caso de Uso.

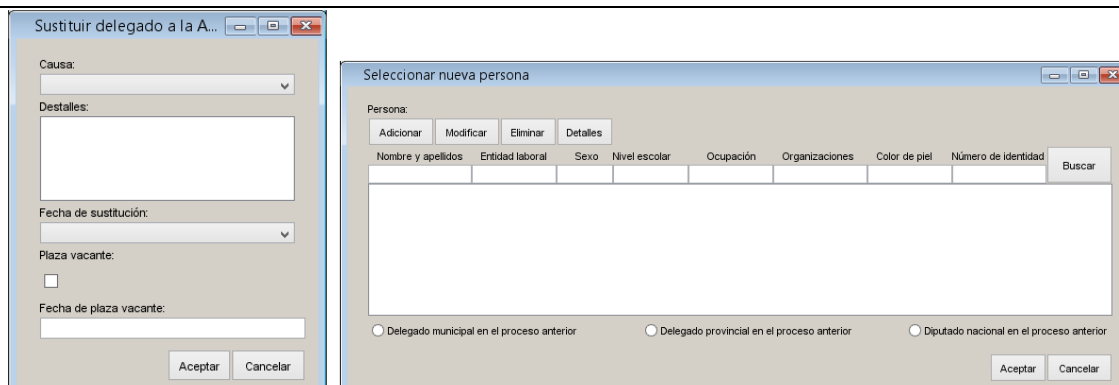
Nº 7: “Campos vacíos”

	Actor	Sistema
10.		El sistema bordea el campo vacío de color rojo y muestra un mensaje de alerta indicando que existen campos en blanco.

Nº 8: “Datos no válidos”

	Actor	Sistema
11.		El sistema bordea el campo con Datos no válidos de color rojo y muestra un mensaje de alerta indicando que existen campos no válidos.

Prototipo elemental de interfaz gráfica de usuario



Sección 3: “Detalles de Delegado a la AMPP (M)”

Flujo básico: “Detalles de Delegado a la AMPP (M)”

	Actor	Sistema
1.	El usuario municipal accede a través del menú a la opción <i>Gestionar Delegados a la AMPP</i> .	
2.		El sistema muestra una ventana con el listado de los Delegados a la AMPP del municipio al que pertenece el usuario.
3.	El usuario municipal selecciona del listado de Delegados aquel que desea visualizar los detalles.	
4.	El usuario municipal presiona el botón <i>Detalles</i> o da doble clic sobre el Delegado.	

Capítulo 2: Análisis y diseño del sistema SILECTOS v2.0

5.		El sistema muestra la interfaz <i>Detalles de Delegado a la AMPP</i> .
6.	El usuario municipal presiona el botón Cerrar.	
7.		Finaliza el Caso de Uso.

Prototipo elemental de interfaz gráfica de usuario

Sección 4: “Búsqueda avanzada”

Sección 4: “Búsqueda avanzada”

	Actor	Sistema
1.	El usuario municipal accede a través del menú izquierdo a la opción <i>Gestionar Delegados a la AMPP</i> .	
2.		El sistema muestra una ventana con el listado de los Delegados a la AMPP del municipio al que pertenece el usuario municipal.
3.	El usuario municipal da clic en la lupa de color azul que se encuentra en la barra de búsqueda.	
		El sistema muestra los campos por los que se puede efectuar la búsqueda.
	El usuario inserta los datos por los que desea efectuar la búsqueda.	
		El sistema muestra un listado según lo especificado.
		El sistema brinda la posibilidad de desactivar el filtrado.
		Finaliza el Caso de Uso.

Prototipo elemental de interfaz gráfica de usuario

2.2. Modelo de diseño

El modelo de diseño describe la realización física de los procesos, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de programación, tienen impacto en el sistema a considerar, siendo una entrada fundamental para las actividades de implementación. Este modelo se puede utilizar para visualizar la implementación y para soportar las técnicas de programación gráfica de la aplicación (Bruegge, y otros, 2008).

Patrón arquitectónico Modelo-Vista-Controlador

El patrón de diseño es una solución estándar para un problema común de programación, un proyecto o estructura de implementación que logra una finalidad determinada. También se considera como una manera más práctica de describir ciertos aspectos de la organización de un programa representadas por conexiones entre componentes de programas (Larman, 2010).

Symfony 2.7 basa su funcionamiento interno en el patrón arquitectónico Modelo-Vista-Controlador (MVC), el cual se encarga de separar los datos, la interfaz de usuario, y la lógica de control de una aplicación en tres componentes distintos. El mismo se aplicará para el diseño de los componentes debido a que propone tres capas fundamentales como lo indica su nombre, las cuales se describen a continuación (Revista Telem@tica, 2012) (Pavón, 2009):

- **El Modelo** es la capa encargada de representar los datos del programa, manejando y controlando todas sus transformaciones, además de ser el responsable de la información almacenada en la base de datos.
- **La Vista** es la capa encargada de la presentación visual de los datos representados por el Modelo, además de ser responsable de las páginas HTML.
- **El Controlador** es la capa encargada de proporcionar respuestas a las peticiones del usuario a partir de las operaciones sobre los datos representados en el Modelo.

Diagrama de paquetes

Un diagrama de paquetes muestra como un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, suministrando una descomposición de la jerarquía lógica de un sistema. Entre sus características se destaca: su empleo en un modelo de desarrollo para agrupar elementos relacionados, y la posibilidad de compartir el trabajo entre los miembros de un equipo asignando individualmente los paquetes (Rumbaugh, y otros, 2009). La organización física de los elementos que conforman el sistema se realizará teniendo en cuenta los principios establecidos por *Symfony v2.7*, que estructuran el proyecto en paquetes o *bundles*. En

la **¡Error! No se encuentra el origen de la referencia.** se muestra detalladamente la estructura del sistema SILECTOS v2.0 (Ver Fig. 6).

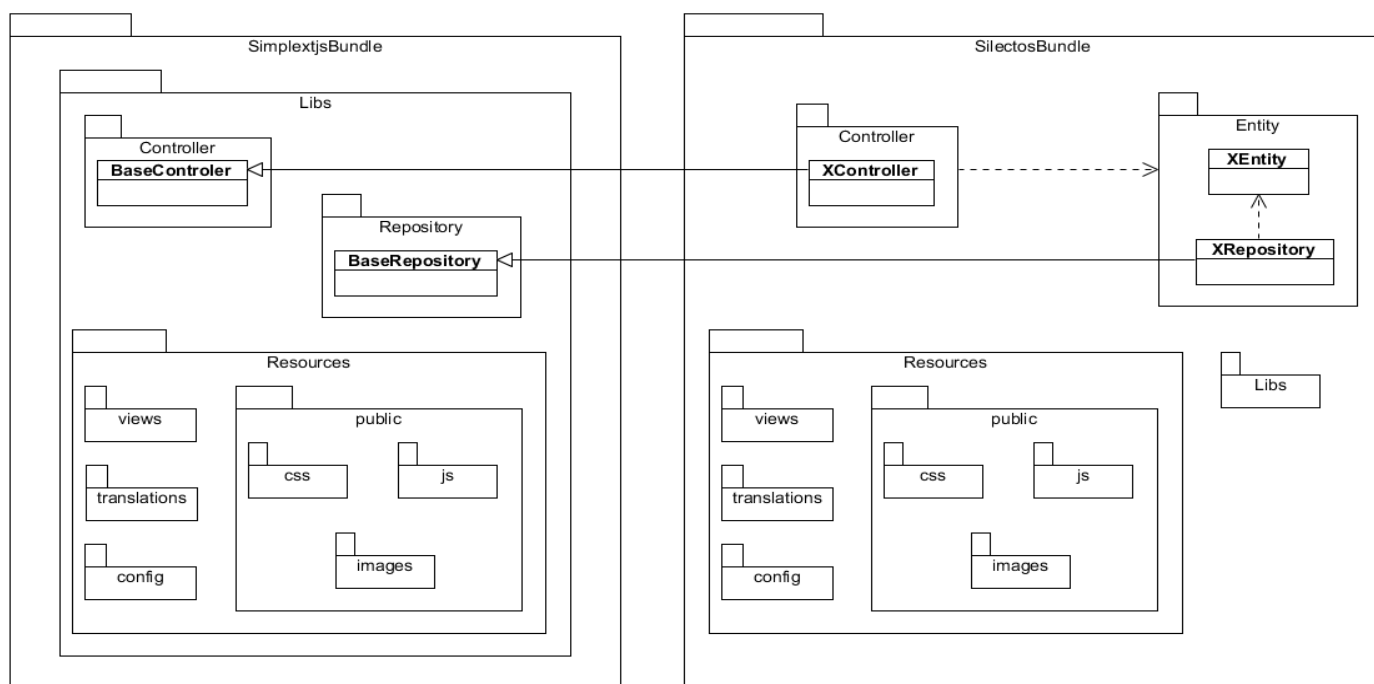


Fig. 6: Diagrama de paquetes del sistema SILECTOS v2.0

Descripción del diagrama de paquetes

El sistema está compuesto por dos *bundles*: *SimplextjsBundle* y *SilectosBundle*. El primero corresponde a una capa intermedia, desarrollada en el proyecto Elecciones para el desarrollo de sus sistemas, que opera entre el marco de trabajo Symfony v2.7 y el segundo *bundle* mencionado. El *SilectosBundle* corresponde a la versión 2.0 de SILECTOS el cual contiene los paquetes *Controller*, *Entity*, *Resources* y *Libs*.

En el paquete *Controller* se encuentran todos los controladores correspondientes a la aplicación, los cuales heredan de *BaseController* del *bundle SimplextjsBundle*. *Entity* es el directorio encargado de almacenar todas las entidades y los repositorios del sistema, heredando estos últimos de *BaseRepository* en la capa intermedia. La carpeta *Libs* será la encargada de agrupar los componentes propios del *bundle* como es el caso de los nomencladores, los normalizadores, los validadores y los componentes de conexión con los servicios de reportes.

Otra de las carpetas importantes dentro de ambos *bundles* es *Resource*, compuesta a su vez por otros paquetes como es el caso de *view*, contenedor de las plantillas *Twig* encargadas de renderizar los diferentes componentes del sistema. Las configuraciones de los diferentes servicios utilizados están contenidas en el

paquete *config*, mientras que *translation* encierra los archivos encargados del manejo de la traducción de las diferentes excepciones. En *public* se almacenan diferentes recursos como hojas de estilos, imágenes y archivos *JavaScript*, cada uno en su carpeta correspondiente. Es importante destacar que independientemente de la riqueza de Symfony v2.7 para la elaboración de la presentación, estas posibilidades no se utilizan dado que el sistema será implementado utilizando ExtJS v4.2, por lo que se hace necesario definir una organización para los componentes de presentación JavaScript. Por lo mismo, dentro de dicho directorio, se ubica otro nombrado *App* con todos los componentes correspondientes a la vista (**¡Error! No se encuentra el origen de la referencia.** Fig. 7).

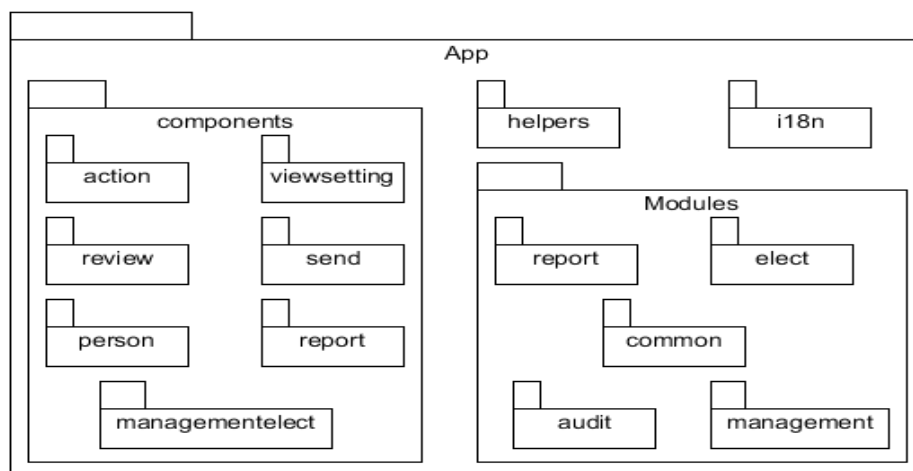


Fig. 7: Diagrama de paquetes del sistema SILECTOS v2.0. Estructura del marco de trabajo ExtJS v4.2 dentro del directorio *js*

El directorio *i18n* es el responsable de almacenar todos los elementos de internacionalización del sistema, entre estos los nombres de los *label*, los mensajes que se le muestran al usuario, títulos de ventanas y campos de formulario. *Helpers* es el paquete encargado de la agrupación de las clases ayudantes para la construcción de interfaz de usuarios, principalmente la construcción de los componentes de selección. En *components* se encuentran los componentes reutilizables de la aplicación, entre los que se encuentran: *action* con las acciones del sistema, *review* para la revisión de la información en los niveles provincia y nación, *person* para modificarlos los datos de los electos, *viewsettings* es contenedor de las clases para la configuración de las interfaces de usuarios, *send* organiza el envío de información a niveles superiores, *report* muestra al usuario la interfaz con los reportes generados desde *JasperReport*, y *management* es el componente base para la gestión de los electos en los diferentes niveles con que cuenta el sistema. En el directorio *modules* serán agrupados los diferentes módulos con que cuenta la aplicación: *report*, *elect*, *audit* que solo es accedido por el usuario con dicho rol, *managemnt* para la gestión de usuarios, roles y permisos, y *comun* donde serán agrupadas las funcionalidades comunes entre todos los usuarios, incluido el listado de noticias que serán mostradas.

Diagrama de clases del diseño

Los diagramas de clases del diseño (DCD) se utilizan para obtener una representación gráfica de las clases de un software y la relación que existe entre estas desde diferentes perspectivas. Con el objetivo de obtener clases más independientes, reutilizables y fáciles de mantener se distribuyen en capas siguiendo el diseño que imponen los estilos arquitectónicos (Rumbaugh, y otros, 2009). La siguiente imagen muestra un ejemplo de diagrama de clase del diseño correspondiente al caso de uso *Gestionar Delegado a la AMPP* (Ver Fig. 8):

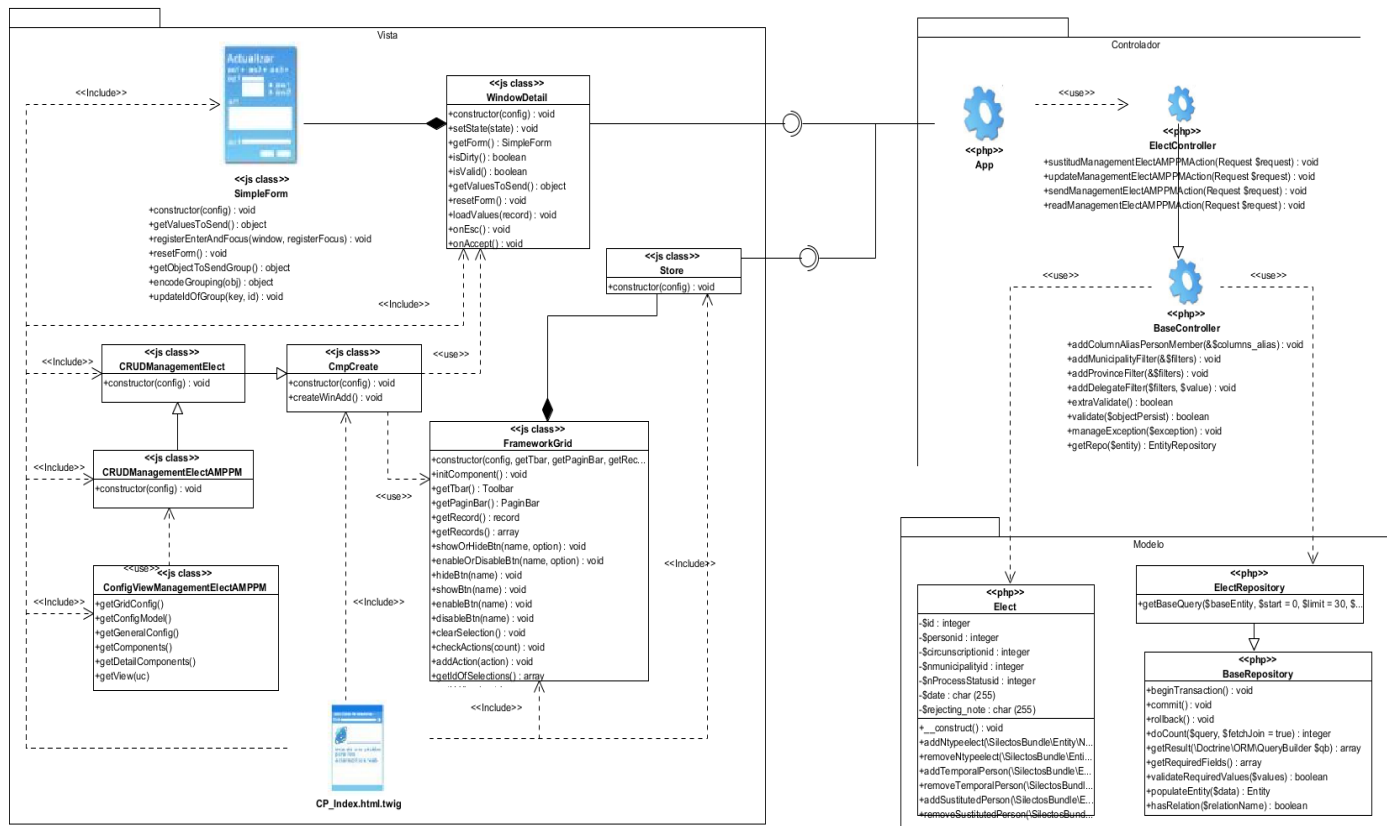


Fig. 8: Diagrama de Clase de Diseño del CU Gestionar Delegado AMPP (M)

Descripción del Diagrama de Clases del Diseño

Los DCD correspondientes al sistema SILECTOS v2.0, representan todas las clases con los atributos y los métodos correspondientes a cada caso de uso, a partir del empleo del patrón MVC. En el controlador, las clases que construyen las respuestas del lado del servidor serán implementadas en lenguaje PHP con la dependencia del marco de trabajo *Symfony v2.7*; mientras que, en la vista, las clases serán implementados en lenguaje *JavaScript* con la dependencia del marco de trabajo *ExtJS v4.2*. Por otra parte, las clases del modelo son generadas por el ORM *Doctrine 2* y serán las encargadas de la conexión con la base de datos. A continuación, se describen las clases del caso de uso Gestionar Delegado AMPP.

Clases del modelo:

- La clase **Elect** contiene la información referente a los electos.
- La clase **ElectRepository** posee las funcionalidades necesarias para realizar búsquedas y garantizar la persistencia de los datos de los electos.

Clases de la vista:

- La clase **CRUDManagementElect** construye la interfaz de usuario para sustituir, modificar, filtrar y ver detalles de un electo.
- La clase **CRUDManagementElectAMPP (M)** construye la interfaz de usuario para sustituir, modificar, filtrar y ver detalles de un electo a nivel municipal.
- La clase **ConfigViewManagementElectAMPP (M)** suministra la configuración necesaria para la construcción de la interfaz de usuario apoyándose en la clase **CRUDManagementElectAMPP (M)**.

Elementos del controlador:

- La clase **ElectController** es la encargada de dar respuesta a las peticiones generadas por el usuario y procesar la información correspondiente a la sustitución de un electo.
- La clase **App** es la encargada de recibir las peticiones generadas por el usuario desde la vista para que el controlador pueda procesar la información correspondiente y luego enviar la respuesta a estas (es la encargada de comunicar la vista con el controlador).

Además, quedaron representadas:

- La clase **BaseRepository** correspondiente a la capa intermedia (SimpleExtjs), encargada del manejo de los datos y la comunicación con los distintos gestores de bases de datos por parte del modelo.
- Las clases **CmpCreate**, **FrameworkGrid**, **Store**, **WindowDetail**, y **SimpleForm** correspondientes al marco de trabajo ExtJS y a la capa intermedia (SimpleExtjs), encargadas de atender a las peticiones de lado del cliente.
- La clase **BaseController** correspondiente a la capa intermedia (SimpleExtjs), encargada de atender las peticiones del lado del servidor.

Patrones GRASP

Los patrones de diseño de software permiten describir fragmentos y reutilizar ideas de diseño basado en la experiencia de otros. Entre los patrones de diseño más conocidos están los Patrones Generales de Asignación de Responsabilidades de Software (GRASP, del inglés *General Responsibility Assignment Software Patterns*), encargados de describir los principios fundamentales de la asignación de

responsabilidades a objetos, expresados en forma de patrones (Larman, 2010). Para lograr un diseño eficaz se emplearán algunos patrones GRASP que se muestran a continuación.

Patrón Controlador: Se emplea en la clase responsable de recibir o manejar un evento del sistema, definiendo para ello el método de operación. Su uso se pone de manifiesto en las clases *ElectController* y *BaseController*, encargadas de interactuar con el modelo y enviar una respuesta a la vista. Se evidencia en la **¡Error! No se encuentra el origen de la referencia.** (Ver Fig. 9).



Fig. 9: Patrón Controlador del Diagrama de Clase de Diseño del CU Gestionar Delegado AMPP (M)

Patrón Bajo Acoplamiento: Este patrón se enfoca en el número de dependencias entre las clases de manera que, en caso de producirse una modificación en alguna de las clases, se tenga la mínima repercusión posible en el resto. Se evidencia en la clase *CRUDManagementElectAMPP (M)* que, heredada de *CRUDManagementElect*, la cual adapta las funcionalidades de la segunda a la gestión de electos a nivel municipal sin incluir cambios en esta última, alcanzando así un bajo acoplamiento (Ver Fig. 10).

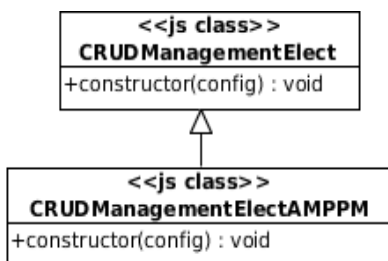


Fig. 10: Patrón Bajo Acoplamiento del Diagrama de Clase de Diseño del CU Gestionar Delegado AMPP (M)

Patrón Alta Cohesión: Este patrón expresa que cada elemento del diseño debe realizar una labor única dentro del sistema. Se evidencia en la clase *CmpCreate* que interactúa con las clases *FrameworkGrid* y *WindowDetail* para atender a las peticiones del cliente que serán heredadas por la clase *CRUDManagementElectAMPP (M)* (Ver Fig. 11).

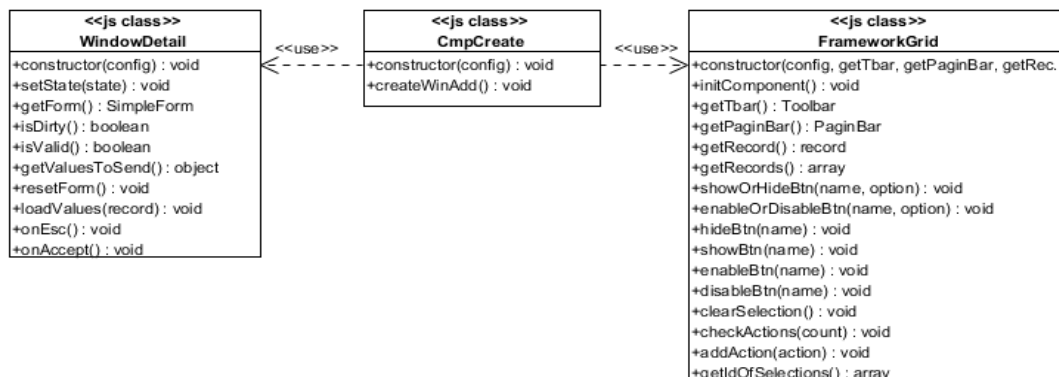


Fig. 11: Patrón Alta Cohesión Diagrama de Clase de Diseño del CU Gestionar Delegado AMPP (M)

Patrones GoF

Los Patrones Pandilla de Cuatro (GoF, del inglés *Gang of Four*) se basan principalmente en dos principios del diseño orientado a objetos: favorecer la composición del objeto sobre la herencia y programar una interfaz, no una implementación. Tienen dos usos principales en el desarrollo de software: plataforma común para desarrolladores y mejores prácticas. Fueron diseñados para proporcionar las mejores soluciones a ciertos problemas que se enfrentan durante el desarrollo del software (Gamma, y otros, 1994). A continuación, se listan los patrones GoF que se emplearán en la implementación del sistema.

El **Patrón Observador (Observer Pattern)** será aplicado a la clase *SessionLoginListener.php*, que estará pendiente del acceso al sistema por parte de los usuarios para la validación de la dirección IP que este tiene asignada y verificar si es permitida. En caso de la dirección no ser permitida, debe registra una traza de dirección inválida y enviar una alarma al auditor. Además, impide el acceso al sistema por parte del usuario (Ver Fig. 12).

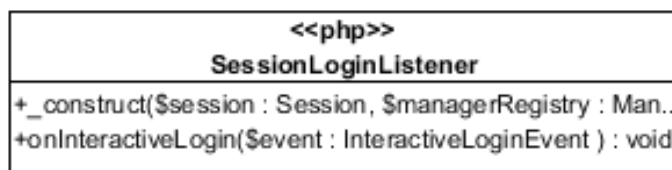


Fig. 12: Patrón GoF Observer Pattern

El **Patrón Mediador (Mediator Pattern)** será evidenciado en la clase *BaseController.php*, contenedora de métodos para operar entre las clases controladoras y el modelo del sistema (Ver Fig. 13).

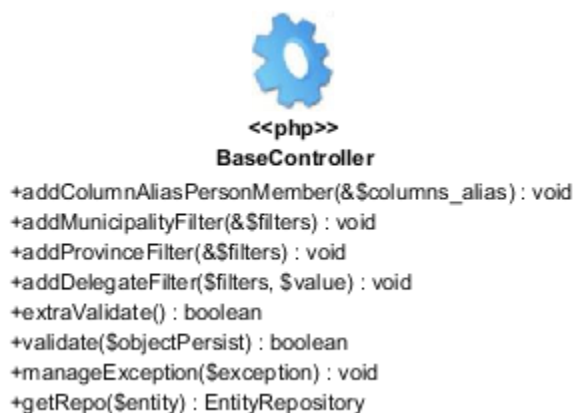


Fig. 13: Patrón GoF Mediator Pattern

Diagrama de secuencia

Los diagramas de secuencias muestran la forma en que un grupo de objetos se comunican (interactúan) entre sí a lo largo del tiempo. Un diagrama de secuencia consta de objetos, mensajes entre estos objetos y una línea de vida del objeto representada por una línea vertical (Pressman, 2011). A continuación, se muestra el diagrama de secuencias correspondiente a la sección Modificar Delegado a la AMPP correspondiente al caso de uso Gestionar Delegado a la AMPP (M) (Ver Fig. 14).

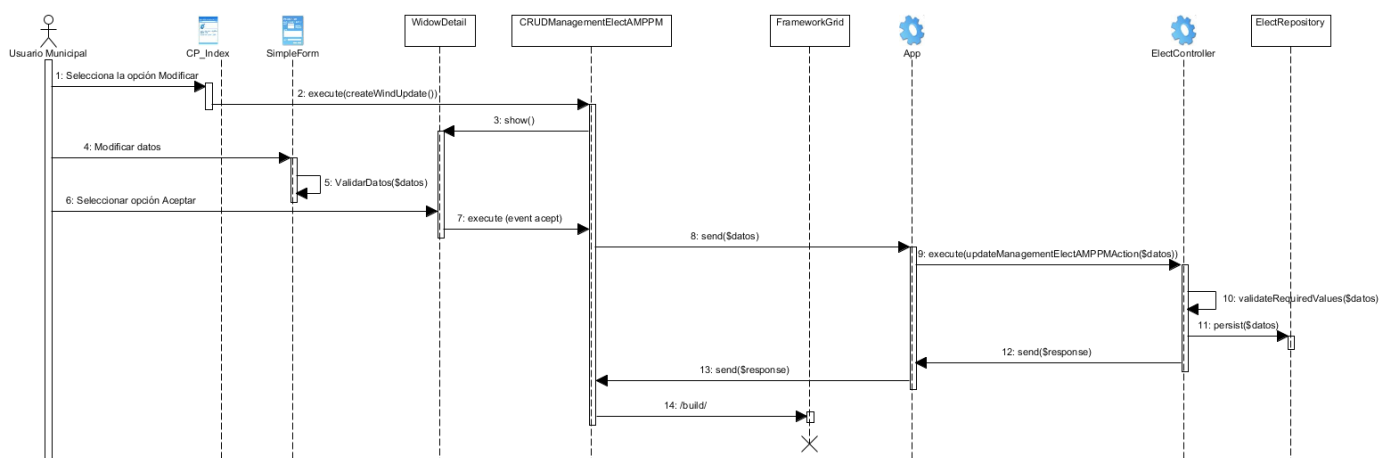


Fig. 14: Diagrama de secuencia. Sección Modificar Delegado AMPP (M)

Descripción del diagrama de secuencia

El Usuario Municipal selecciona uno de los electos del listado que muestra el sistema y selecciona la opción *Modificar* en la página cliente *CP_Index*. Dicha página es la encargada de ejecutar el método *createWindUpdate()* de la clase *CRUDManagementElectAMPP (M)*, la que mostrará la ventana *WindowDetail* con el formulario *SimpleForm*. Es en este formulario donde el Usuario Municipal modificará los datos

deseados, mientras que el mismo, valida que estos sean insertados correctamente. Posteriormente, el Usuario Municipal selecciona la opción *Aceptar* en la ventana *WidowDetail*. La vista lanza el evento *accept()* que es capturado por la clase *CRUDManagementElectAMPP (M)* y esta se encarga de enviar los datos al controlador frontal *App* encargado de ejecutar los métodos *updateManagementElectAMPP (M)* y *Action()* del controlador *ElectController*. Luego de realizada esta acción, la clase *ElectController* ejecuta los métodos *validateRequiredValues()* para validar los datos en la parte del servidor y *persist()* para actualizar los datos en la base de datos. Una vez finalizadas estas acciones, la clase *ElectController* envía un mensaje de respuesta al controlador frontal *App* y este a la clase *CRUDManagementElectAMPP (M)* para que esta última construya el *FrameworkGrid*, encargado de mostrar el listado de electos actualizado.

Diagrama entidad-relación

Un diagrama o modelo entidad–relación es una herramienta para el modelado de los datos que permite representar gráficamente las entidades relevantes de un sistema de información, así como sus interrelaciones y propiedades (Jacobson, y otros, 2004). A continuación, se muestra el diagrama entidad-relación con la información de los electos que debe ser almacenada en la base de datos de SILECTOS v2.0 (Diagrama entidad-relación del esquema data del sistema SILECTOS v2.0.

Fig. 15).

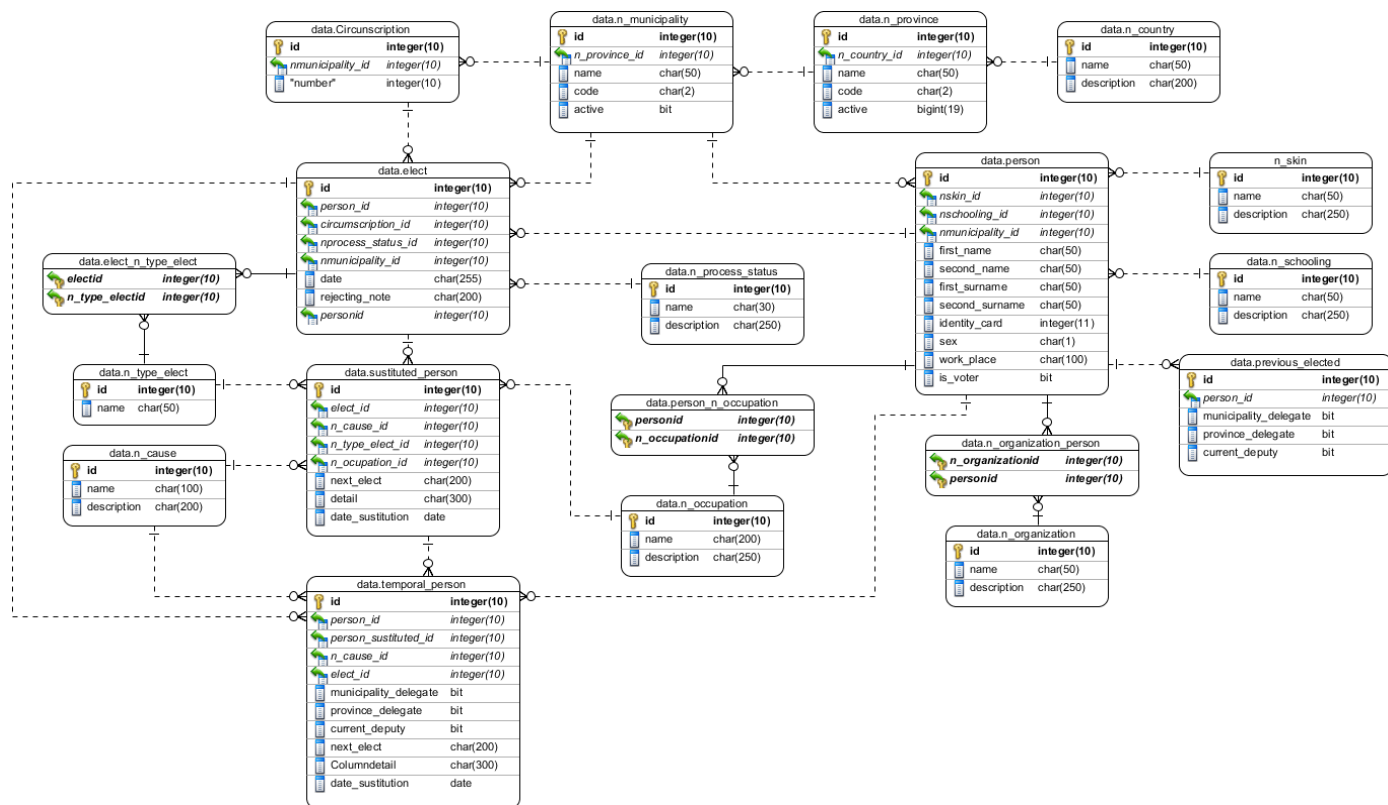


Fig. 15: Diagrama entidad-relación del esquema data del sistema SILECTOS v2.0.

Descripción de las entidades del diagrama entidad-relación

En la base de datos, cada entidad debe poseer un identificador que será incrementado automáticamente a medida que se inserten los datos. La entidad *circunscription* tiene asociado el identificador del municipio al que pertenece; y este, *n_municipality*, el de la provincia que le concierne. Por su parte, *n_province* se identifica además por el país. La tabla *person* contiene para cada individuo los identificadores del nivel escolar y color de la piel, además de mantener relación de mucho a mucho con las tablas *n_organiation* y *n_occupation*.

La entidad *elect* posee los identificadores de cada persona que resultó electa en el proceso electoral, junto con el de *circunscription*, *n_municipality* y *n_status_process* correspondiente. La misma establece una relación de mucho a mucho con *n_type_elect*. Para el proceso de sustitución, la tabla *sustituted_person* almacena el identificador del electo que será sustituido y de la causa. La información de la persona que ocupará el cargo estará provisionalmente en *temporal_person* hasta que sea aprobada o rechazada. Asimismo, en *previous_elected* se especifica si una persona fue Delegado o Diputado en el proceso electoral anterior a partir de su identificador.

Diagrama de despliegue

Describe el ambiente dentro del cual el sistema será instalado. Establece una correspondencia entre la arquitectura de software y la arquitectura de hardware del sistema, por lo que el rol encargado de realizarlo es el Arquitecto de Software. Muestran a los nodos procesadores la distribución de los procesos y de los componentes (Larman, 2010). A continuación, se muestra el diagrama de despliegue para el sistema SILECTOS v2.0 (Ver Fig. 16).

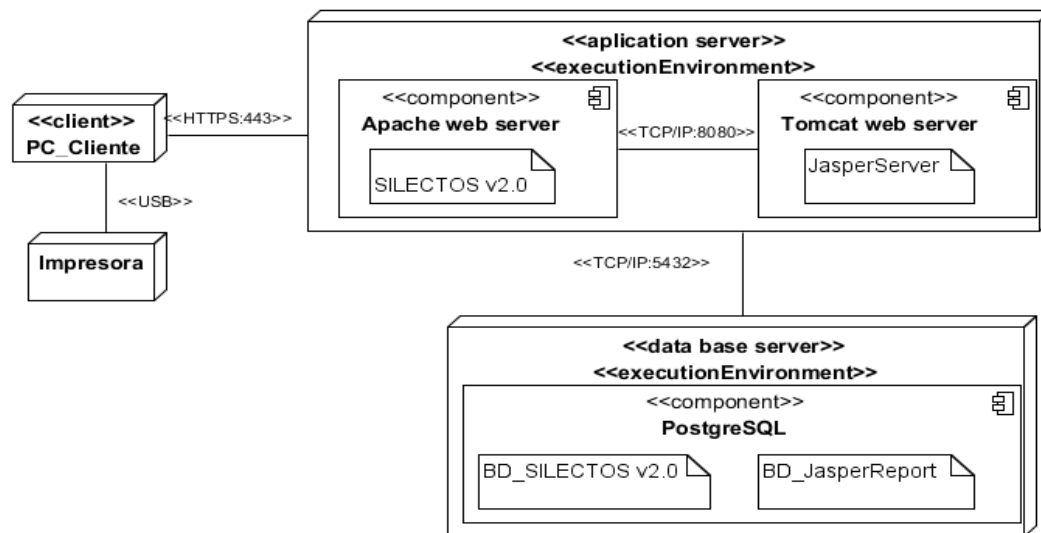


Fig. 16: Diagrama de despliegue de SILECTOS v2.0

Descripción de los nodos del diagrama de despliegue

- **PC Cliente:** se refiere a las estaciones de trabajo que el usuario utilizará para acceder a la aplicación Web.
- **Impresora:** se refiere al dispositivo conectado a las estaciones de trabajo, por donde se imprimirán los reportes.
- **Servidor de Aplicación:** servidor de aplicación utilizado para la publicación de la aplicación; la conexión del sistema con la PC Cliente se utiliza HTTPS (del inglés, *Hypertext Transfer Protocol Secure*). Es la herramienta principal para ejecutar la lógica de negocio en el lado del servidor. Es el responsable de ejecutar el código de las páginas servidor.
- **Servidor de Base de Datos:** se refiere a un servidor que radica un nodo determinado donde serán guardados los datos. En el mismo estarán almacenados todos los datos recopilados por todos los nodos. El servidor de bases de datos elegido es *PostgreSQL*, el cual está disponible para Linux y Windows.

- **Conexión HTTPS:** es el protocolo seguro de transferencia de hipertexto utilizado entre el navegador de los clientes y el servidor WEB por el puerto 443. Este elemento de la arquitectura representa un tipo de comunicación no orientado a la conexión entre clientes y servidor.
- **Conexión TCP/IP:** es la base del Internet que sirve para enlazar computadoras. El protocolo TCP/IP es utilizado para establecer la conexión entre el servidor de aplicación y el servidor de base de datos por el puerto 5432, y entre los componentes Apache Web Server y Tomcat Web Server por el puerto 8080.

2.3. Conclusiones del capítulo

Después de realizar los procesos de análisis y diseño del SILECTOS v2.0 se concluye que se tomó como punto de partida al modelo de dominio como para el diseño del sistema, quedando definidos 13 requisitos no funcionales y 255 funcionales, estos últimos se agruparon en 29 Casos de Uso mediante el empleo de los patrones correspondientes. La definición del patrón arquitectónico Modelo-Vista-Controlador, permitió identificar los componentes involucrados en el desarrollo de la solución. También se diseñó el diagrama de paquetes que suministró una descomposición de la jerarquía lógica del sistema. Se creó el diagrama de clases de diseño que proporcionó la representación gráfica de las clases del software y la relación que existe entre estas. Se expresó la forma en que un grupo de objetos se comunican (interactúan) entre sí a lo largo del tiempo en el diagrama de secuencia. Se elaboró el diagrama entidad-relación que representó gráficamente el modelado de los datos y las entidades del sistema. Además, se representó el ambiente de instalación de la aplicación en el diagrama de despliegue.

Capítulo 3: Implementación y pruebas del sistema SILECTOS v2.0

En el presente capítulo se muestra el modelo de implementación como resultado del diseño anteriormente desarrollado, ejemplificado por el diagrama de componentes del caso de uso Gestionar delegados a la AMPP (M) y los estándares de codificación empleados en el código. Se describen las pruebas realizadas, con el objetivo de comprobar las funcionalidades del software en los diferentes escenarios para, de esta forma, verificar en todos los casos que los resultados de las pruebas sean los esperados y, por tanto, el correcto funcionamiento del sistema.

3.1. Modelo de implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos (Pressman, 2011).

Diagrama de componentes

Un componente es una parte modular, desplegable y reemplazable de un sistema que encapsula implementación, puede exponer un conjunto de interfaces y proporciona la realización de los mismos. Son las piezas reutilizables de alto nivel a partir de las cuales se pueden construir los sistemas. Es por esto que, los diagramas de componentes son empleados para describir la estructura física del modelo de implementación del sistema y mostrar las relaciones entre sus elementos, reflejando de este modo, una estructura de alto nivel. Representan las dependencias entre componentes de software incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. Se rigen por la filosofía de la alta cohesión interna y el bajo acoplamiento externo (Rumbaugh, y otros, 2009). A continuación, se muestra el diagrama de componentes del CU Gestionar delegados a la AMPP (M) (Ver Fig. 17).

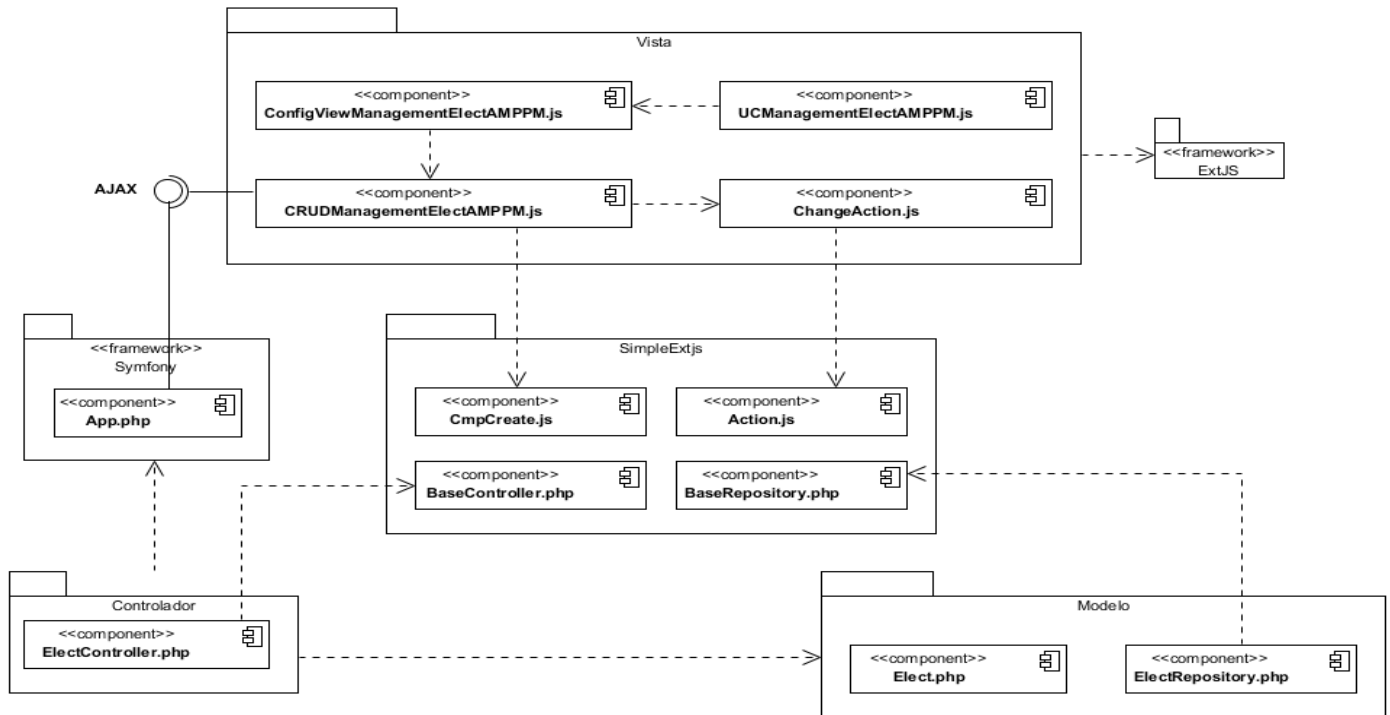


Fig. 17: Diagrama de componentes del CU Gestionar delegados a la AMPP (M)

Descripción del diagrama de componentes

El diagrama de componentes de la **¡Error! No se encuentra el origen de la referencia.**, corresponde al caso de uso Gestionar delegados a la AMPP (M), donde los componentes son agrupados en paquetes siguiendo el patrón arquitectónico MVC, por el que se rige el marco de trabajo Symfony v2.7. En la misma, se encuentran almacenados todos los componentes del lado del cliente en el paquete Vista, los cuales hacen uso de los elementos correspondientes al subsistema *ExtJS* y *SimpleExtJS*. Este paquete contiene a los componentes encargados de manejar el flujo de las interfaces, siendo la clase *CRUDManagementElectAMPP (M).js* la responsable de controlar las principales acciones de llamada al componente App del subsistema *Symfony*, utilizando la tecnología AJAX y el formato de intercambio JSON. El componente *ElectController.php*, del paquete Controlador, será el encargado de construir y devolver las respuestas del servidor al componente App, apoyándose en el elemento *BaseController.php* del subsistema *SimpleExtJS* y en las clases del paquete Modelo. Por su parte, *Elect.php* y *ElectRepository.php* son los componentes pertenecientes al Modelo encargados de contener la información de los electos, haciendo uso del componente *BaseRepository.php* en el subsistema *SimpleExtJS*.

Código fuente

El código fuente es un conjunto de líneas de texto e instrucciones que debe seguir la computadora para ejecutar dicho programa, por lo que, en él está escrito por completo su funcionamiento. Su contenido está escrito en un lenguaje de programación específico, para una mejor comprensión por parte de los programadores. Debe traducirse a lenguaje máquina para que pueda ser ejecutado por la computadora o a *bytecode* para que pueda ser ejecutado por un intérprete. Este proceso se denomina compilación (Rumbaugh, y otros, 2009).

La forma de escribir el código fuente es propia de cada programador y completamente diferente a la de cualquier otro. De la forma empleada depende la facilidad para ser comprendido, reutilizable o mejorado por otro desarrollador. Por lo mismo se hace necesario el empleo de determinados estándares de codificación para guiar el proceso de implementación. Los empleados en el SILECTOS v2.0 se describen a continuación.

Estándares de codificación

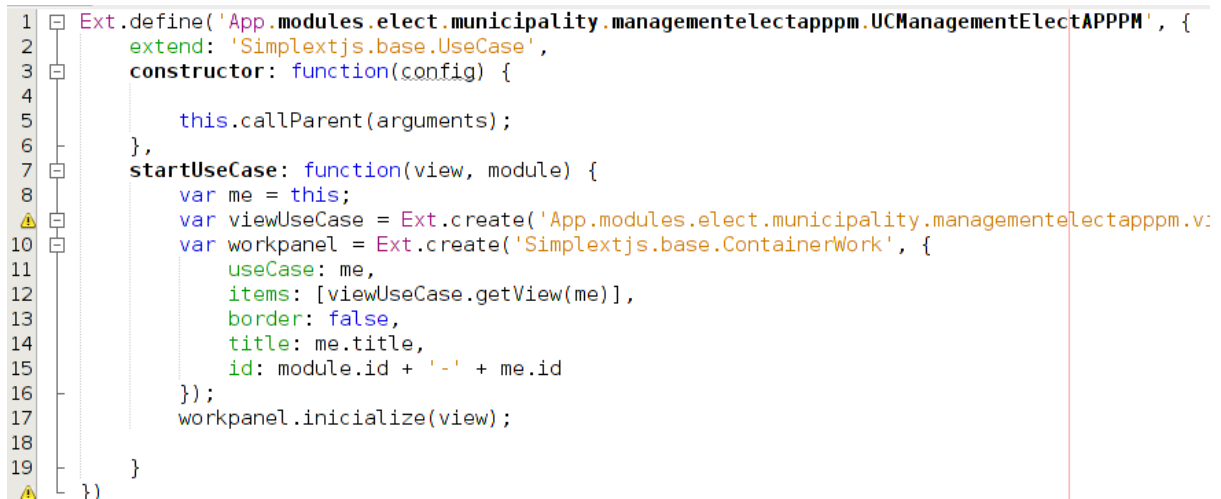
Un estándar de codificación completo comprende todos los aspectos de la generación de código. Al comenzar un proyecto de software, se debe establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada, permitiendo que todos los participantes lo puedan entender en menos tiempo y que el código en consecuencia presente mantenibilidad. De esta forma se pueden minimizar riesgos de incumplimiento de fechas de actividades importantes, de gastos excesivos en relación a los costos estimados e insatisfacciones de los usuarios de los sistemas (Bruegge, y otros, 2008).

Los estándares por los que se rige la implementación del sistema, propuestos por ExtJS 4.2, se describen a continuación:

- Los nombres de clase, métodos y variables sólo pueden contener caracteres alfanuméricos (no emplear subrayados, guiones, o cualquier otro carácter no alfanumérico).
- Los nombres de clase deben ser agrupados en paquetes cuando proceda y ubicadas en un espacio de nombres correctamente, utilizando la característica de objetos de notación de punto (.). Como mínimo, debería haber un único espacio de nombres de primer nivel, seguido del nombre de clase.
- Los espacios de nombres de nivel superior y los nombres de clase deben ser en *camelCased*, siendo todo lo demás escrito en minúscula. Igual sucede con los métodos y las variables.
- Los nombres de las clases se asignan directamente a las rutas de archivo en el que están almacenados. Como resultado, debe haber sólo una clase por archivo. Todas las clases deben permanecer dentro de un directorio y deben ser debidamente ubicadas en un espacio de nombres correctamente.

Capítulo 3: Implementación y prueba del sistema SILECTOS v2.0

Los mismos se evidencian en la **¡Error! No se encuentra el origen de la referencia.** (Ver Fig. 18).



```
1 Ext.define('App.modules.elect.municipality.managementelectapppm.UCManagementElectAPPPM', {
2     extend: 'Simplextjs.base.UseCase',
3     constructor: function(config) {
4
5         this.callParent(arguments);
6     },
7     startUseCase: function(view, module) {
8         var me = this;
9         var viewUseCase = Ext.create('App.modules.elect.municipality.managementelectapppm.v:
10        var workpanel = Ext.create('Simplextjs.base.ContainerWork', {
11            useCase: me,
12            items: [viewUseCase.getView(me)],
13            border: false,
14            title: me.title,
15            id: module.id + '-' + me.id
16        });
17        workpanel.inicialize(view);
18    }
19 }
```

Fig. 18: Ejemplo de código fuente evidenciando los estándares de codificación definidos por el marco de trabajo ExtJS v4.2

Por otra parte, Symfony 2.7 se rige por los estándares definidos en el documento PSR-1, descrito a continuación.

- Utilizar *camelCase* y no guiones bajos, para variables, funciones y nombres de métodos.
- Utilizar guiones bajos para definir opciones, argumentos y nombres de parámetros.
- Utilizar los *namespace* para todas las clases.
- Añadir como sufijo *Interface* a las interfaces.
- Utilizar caracteres alfanuméricos y guiones bajos para nombres de archivos.
- Agregar una línea en blanco antes de la sentencia *return*.
- Colocar las llaves en sus propias líneas para clases, métodos y declaración de funciones.
- Utilizar letras mayúsculas para constantes, con palabras separadas por guiones bajos.
- Definir una clase por archivo.
- Declarar las propiedades de las clases antes de los métodos.

Los mismos se evidencian en la **¡Error! No se encuentra el origen de la referencia.** (Ver Fig. 19).

```
7 use SilectosBundle\Libs\Decorator\CustomDecorator;
8 use Symfony\Component\HttpFoundation\Request;
9 use Symfony\Component\HttpFoundation\JsonResponse;
10
11 class UserProfileController extends BaseController{
12     /**
13      * Finds and displays a ManagementUserProfile UserProfile entity.
14      *
15      */
16     public function readManagementUserProfileAction(Request $request) {
17         $start = $request->get('start');
18         $limit = $request->get('limit');
19         $filters = $request->get('filter');
20         if ($filters) {
21             $filters = json_decode($filters);
22         } else {
23             $filters = array();
24         }
25         $columnsAlias = array();
26         $result = $this->filterModel('UserProfile', $start, $limit, $filters, $c
27
28         return new JsonResponse($result);
29     }
30 }
```

Fig. 19: Ejemplo de código fuente evidenciando los estándares de codificación definidos por el marco de trabajo Symfony v2.7.

3.2. Pruebas del software

Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente. Una estrategia de prueba del software debe incluir pruebas de bajo nivel que verifiquen que todos los pequeños segmentos de código fuente se han implementado correctamente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requisitos del cliente. De esta forma, el proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales para la detección de errores. Además de ser utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa (Pressman, 2011).

Niveles de pruebas

Las pruebas del software son aplicadas en diferentes niveles y, cada una de ellas, dependen del objetivo y del momento en el ciclo de vida que se encuentre la elaboración del proyecto. Los niveles de pruebas definidos por Pressman son: pruebas de desarrollador, independiente, integración, sistema y aceptación. En el sistema SILECTOS v2.0 se aplicarán las pruebas en los niveles desarrollador, sistema y aceptación para probar que el sistema cumpla con los requisitos funcionales previamente definidos en la fase de análisis. A continuación, se describen los tipos de pruebas aplicados en estos niveles.

Tipos de pruebas

Los tipos de pruebas que se realizan a un software dependen del nivel al que serán aplicadas. Al sistema SILECTOS v2.0 se le aplicarán las pruebas funcionales a nivel de desarrollador, mientras que las de seguridad y rendimiento se aplicarán a nivel de sistema. Las mismas se describen a continuación:

- Pruebas funcionales: se centran en comprobar que el sistema desarrollado funcione acorde a las especificaciones funcionales del sistema, pues permite comprobar los posibles defectos derivados en la fase de programación.
- Pruebas de seguridad: se centran en garantizar que los sistemas cumplan con determinados requisitos en materia de seguridad.
- Pruebas de rendimiento (Carga): se realizan para determinar la rapidez con que se realiza una tarea en el sistema bajo determinadas condiciones particulares de trabajo.
- Pruebas de aceptación (prueba alfa): se realizan para comprobar que el sistema puede ser usado por los usuarios finales para ejecutar aquellas funciones y tareas definidas.

Métodos de prueba

El método de prueba caja negra son las pruebas que se llevan a cabo sobre la interfaz del software y se centran en los requisitos funcionales del software. Se realizan para verificar que el comportamiento del sistema corresponda a las especificaciones del producto y a las expectativas del usuario (Pressman, 2011). Los casos de prueba se construyen a partir de las especificaciones del sistema.

El método de prueba caja blanca, en ocasiones llamada prueba de caja de vidrio, según (Pressman, 2011). es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba que: garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez, revisen todas las decisiones lógicas en sus lados verdadero y falso, ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y revisen estructuras de datos internas para garantizar su validez.

Pruebas funcionales aplicando el método Caja Negra

A continuación, se describen las variables del CU Gestionar delegado a la AMPP (M) en la Tabla 5 (Ver Tabla 1Tabla 5), para la realización del caso de prueba al escenario Buscar delegado a la AMPP (M) en la Tabla 6 (Ver Tabla 6).

Capítulo 3: Implementación y prueba del sistema SILECTOS v2.0

Tabla 5: Descripción de las variables para el caso de prueba Gestionar delegado a la AMPP (M)

N°	Nombre del campo	Clasificación	Valor Nulo	Descripción
1.	Nombre y Apellidos	Campo de texto	No	Cadena de caracteres solamente alfabéticos, con no más de 255 caracteres.
2.	Numero de identidad	Campo de texto	No	Cadena de caracteres solamente numéricos, con no más de 11 caracteres.
3.	Sexo	Campo lista desplegable	No	Campo lista desplegable. Muestra una lista con los dos tipos de sexo: femenino o masculino.
4.	Circunscripción	Campo lista desplegable	No	Campo lista desplegable. Muestra una lista con los números de las circunscripciones pertenecientes a determinado municipio.
5.	Estado de proceso	Campo lista desplegable	No	Campo lista desplegable. Muestra una lista con los tipos de estados procesos en los que se puede encontrar un electo: Plaza vacante, En Elaboración, Pendiente a cambio, Aprobado

Matriz de datos: Escenario Buscar Delegado a la AMPP.

Tabla 6: Caso de Prueba. Escenario *Buscar Delegado a la AMPP (M)*.

Escenario	Descripción	Variables					Respuesta del sistema	Flujo Central
		1	2	3	4	5		
EC 1.1 Buscar_a_de terminados_ delegados_a _la_AMPP_tr ue	Este es el escenario ideal donde el sistema muestra todos los delegados electos en la AMPP de sexo femenino, pertenecientes al municipio correspondiente	V	V	V	V	V	El sistema muestra el resultado de la búsqueda a partir de los parámetros que introdujo el usuario.	1. El usuario accede al sistema y despliega el módulo Electos en la barra lateral izquierda. 2. El usuario selecciona la opción: Gestionar delegados a la AMPP (M) 3. El sistema lista los delegados a la AMPP existentes en el sistema. 4. El usuario introduce los datos en los campos: Nombre y Apellidos, Número de identidad, Sexo, Circunscripción y Estado de proceso. 5. El sistema habilita la opción

Capítulo 3: Implementación y prueba del sistema SILECTOS v2.0

	al usuario autenticado.						Limpiar en aquellos campos donde se introdujeron datos.
EC 1.2 Buscar_a_de_terminados_delegados_a_la_AMPP_false	Este es el incorrecto donde se introducen datos en el campo Nombre y Apellidos y el sistema no muestra ningún resultado en la búsqueda.	I	V	V	V	V	El sistema no muestra el resultado de la búsqueda a partir de los parámetros que introdujo el usuario. 1. El usuario accede al sistema y despliega el módulo Electos en la barra lateral izquierda. 2. El usuario selecciona la opción: Gestionar delegados a la AMPP (M) 3. El sistema lista los delegados a la AMPP existentes en el sistema. 4. El usuario introduce los datos en los campos: Nombre y Apellidos, Número de identidad, Sexo, Circunscripción y Estado de proceso. 5. El sistema habilita la opción Limpiar en aquellos campos donde se introdujeron datos.
EC 1.3 Buscar_a_de_terminados_delegados_a_la_AMPP_false	Este es el incorrecto donde se introducen datos en el campo Número de identidad y el sistema no muestra ningún resultado en la búsqueda.	V	I	V	V	V	El sistema no muestra el resultado de la búsqueda a partir de los parámetros que introdujo el usuario. 1. El usuario accede al sistema y despliega el módulo Electos en la barra lateral izquierda. 2. El usuario selecciona la opción: Gestionar delegados a la AMPP (M) 3. El sistema lista los delegados a la AMPP existentes en el sistema. 4. El usuario introduce los datos en los campos: Nombre y Apellidos, Número de identidad, Sexo, Circunscripción y Estado de proceso. 5. El sistema habilita la opción Limpiar en aquellos campos donde se introdujeron datos.

Una vez finalizadas las pruebas funcionales mediante el método de caja negra, se comprobó el correcto funcionamiento y validación de los campos en el sistema. Al concluir el ciclo iterativo fueron detectadas un total de 15 no conformidades clasificadas: en errores de validación, errores en el diseño de los reportes y errores ortográficos. Las mismas fueron resueltas paulatinamente hasta garantizar su solución. (Ver Fig. 20)



Fig. 20: Iteraciones de las pruebas funcionales.

Pruebas funcionales aplicando el método Caja Blanca

Como parte de este tipo de prueba, se puede utilizar la técnica de camino básico, para lo cual es necesario conocer el número de caminos independientes de un determinado algoritmo mediante el cálculo de la complejidad ciclomática (Ver Fig. 21). Este se realiza de tres formas diferentes:

- El número de regiones del grafo de flujo coincide con la complejidad ciclomática.
- La complejidad ciclomática $V(G)$ de un grafo de flujo G se define como: $V(G) = A - N + 2$ donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.
- $V(G)$ también se calcula como el resultado de $P + 1$ donde P es el número de nodos predicados (nodos de los cuales parten dos o más aristas) que tiene contenido el grafo de flujo G .

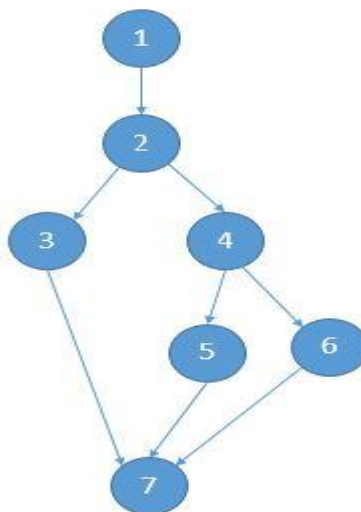


Fig. 21: Grafo de flujo asociado a la funcionalidad

Resultado del cálculo de la complejidad ciclomática:

- $V(G) = A - N + 2 = 8 - 7 + 2 = 3$
- $V(G) = P + 1 = 2 + 1 = 3$

Capítulo 3: Implementación y prueba del sistema SILECTOS v2.0

A partir del resultado obtenido se determina que la funcionalidad presenta una complejidad ciclomática de 3, lo que deriva que existe a lo sumo 3 caminos lógicos por donde ejecutarse dicha funcionalidad. A continuación, se muestran los caminos básicos (Ver tabla 7).

Tabla 7: Caminos básicos

No.	Camino básico
1	1-2-3-7
2	1-2-4-5-7
3	1-2-4-5-7

Pruebas de seguridad

Las pruebas de seguridad se realizaron a partir de las no conformidades detectadas por la Dirección de Tecnologías y Sistemas del Ministerio del Interior MININT, en el documento Evaluación de los requisitos de Seguridad. Los resultados se muestran a continuación. (Ver Tabla 8)

Tabla 8: : Resultados de las pruebas de seguridad

Casos de Pruebas de Seguridad	
Descripción	Resultado
Violación de la seguridad del sistema intentando acceder a éste con usuarios no existentes o utilizando usuarios reales con contraseñas erróneas.	No se logró acceder al sistema con usuarios no registrados previamente ni con falsas contraseñas, demostrando que el sistema cuenta con un mecanismo de autenticación seguro, garantizando que solamente trabajen con el programa personas autorizadas previamente. Luego del tercer intento de acceso con estas características, el usuario quedará desactivado.
Violación de la seguridad del sistema intentando acceder a éste con <i>captcha</i> incorrecto.	Luego de introducir tres veces seguidas el <i>captcha</i> incorrecto, el sistema bloquea la dirección IP desde la que se realiza la operación.
Violación de la seguridad del sistema intentando acceder al mismo una vez pasada la fecha de expiración de la contraseña.	No se logró acceder al sistema una vez expirada la fecha fin de validez de la contraseña.
Violación de la seguridad del sistema accediendo al mismo con un usuario deshabilitado.	No pudo acceder al sistema con usuarios deshabilitados.

Capítulo 3: Implementación y prueba del sistema SILECTOS v2.0

Violación de la seguridad del sistema accediendo al mismo desde una dirección IP deshabilitada o no permitida.	No se logró acceder al sistema desde direcciones IP no registradas ni habilitadas en la base de datos.
--	--

La realización de esta prueba demostró que el sistema cuenta con los mecanismos de seguridad establecidos por la Dirección de Tecnologías y Sistemas del Ministerio del Interior MININT, en el documento “Evaluación de los requisitos de Seguridad” para la protección de la información que se gestiona en las comisiones electorales del país.

Pruebas de rendimiento (Carga)

Con el objetivo de determinar la velocidad de respuesta del sistema se realizaron las pruebas de rendimiento, específicamente de Carga. Para esto se tuvo en cuenta como condiciones particulares de trabajo: la cantidad de peticiones simultáneas y la cantidad de información que se debe procesar. En su desarrollo se empleó la herramienta JMeter v2.10 que permite probar la resistencia y el rendimiento de aplicaciones web durante la simulación de cargas al servidor. Los resultados de la misma se reflejan en la Tabla 9, mostrando los tiempos de respuesta del sistema (en segundos) para determinadas cantidades de usuarios en SILECTOS v1.0 y v2.0 (Ver Tabla 9).

Tabla 9: Resultado de las pruebas de Rendimiento (Carga)

Cantidad de usuarios simulados conectados simultáneamente	SILECTOS v1.0 Tiempo (s)	SILECTOS v2.0 Tiempo (s)
100	1.064	0.572
150	1.744	0.632
200	2.003	0.724

Como se muestra en la **¡Error! No se encuentra el origen de la referencia.9**, los tiempos de respuestas obtenidos en la generación de información en reportes mediante el uso de SILECTOS v1.0 exceden a los alcanzados en el nuevo software desarrollado. Esto se debe a que, en SILECTOS v1.0, fue necesario procesar mayor volumen de información para mostrar en los reportes porque la herramienta externa GDR v1.8 no realizaba determinados cálculos. Sin embargo, el diseñador de plantillas *IReport* permite realizar varias operaciones sobre las variables, por lo que no es necesario que durante las consultas a la base de datos se realicen los cálculos. Por tanto, se puede afirmar que SILECTOS v2.0 responde con mayor rapidez las peticiones realizadas por los usuarios.

Pruebas de aceptación

Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales fue construido. Es realizada por el usuario final en lugar de por los desarrolladores del software, para que este valide todos los requerimientos definidos en la etapa de análisis (Pressman, 2011). Al sistema SILECTOS v2.0 se le aplicó la prueba alfa, ya que fue llevada a cabo en un entorno previamente preparado para su desarrollo. La prueba se llevó a cabo por un grupo representativo de usuarios finales del en presencia de los desarrolladores. La misma concluyó la aceptación por parte del cliente, evidenciada en el Acta de Aceptación.

3.3. Conclusiones del capítulo

Después de realizar los procesos de implementación y prueba del SILECTOS v2.0 se concluye que se realizó el diagrama de componentes con el propósito de mostrar los componentes del sistema y sus relaciones. Quedó implementado el sistema que cumplió con los estándares de codificación establecidos para lograr un estilo claro y organizado del código. Se aplicó pruebas funcionales empleando el método de caja negra basado en la técnica de partición de equivalencia y las no conformidades detectadas fueron resueltas satisfactoriamente. Además, se empleó el método de caja blanca que utilizó la técnica de camino básico y determinó la complejidad ciclomática definiendo el número de caminos independientes de un algoritmo específico. También se realizó pruebas de seguridad que comprobó los elementos de protección presentes en el sistema. A través de las pruebas de rendimiento por carga, automatizándola con la herramienta *JMeter* v2.10 se demostró que los tiempos de respuestas son menores a un segundo.

Conclusiones generales

Con la realización del trabajo de diploma se logró cumplir con los objetivos planteados, desarrollándose el Sistema para la Gestión de Electos SILECTOS v2.0 que ofrece apoyo a la toma de decisiones en la sustitución de los electos para las diferentes comisiones electorales de Cuba. Por lo que se concluye que:

- El estudio de los fundamentos teóricos de la investigación, permitió seleccionar una metodología para organizar de manera estructurada el proceso de desarrollo de software, así como las herramientas que cuentan con las características necesarias para desarrollar la solución propuesta.
- El análisis de soluciones existentes posibilitó un mejor entendimiento del proceso de negocio de las comisiones electorales y permitió la obtención de requerimientos necesario para satisfacer las necesidades del cliente.
- El diseño del sistema permitió obtener los artefactos para el desarrollo de SILECTOS v2.0 permitiendo la implementación de un software que responde a las necesidades del cliente.
- La implementación del Sistema para la Gestión de Electos SILECTOS v2.0 contribuye al proceso de apoyo a la toma de decisiones mediante el desarrollo de un software, garantizando la gestión y visualización de los electos, además de la generación de reporte.
- La validación de la solución mediante la aplicación de las pruebas: funcionales, de seguridad, de rendimientos y de aceptación, permitió obtener un sistema funcional con almacenamiento de información que cumple con los requerimientos definidos por el cliente.

Referencias Bibliográficas

Alegsa, Leandro. 2013. *Diccionario de Informática y Tecnología.* 2013.

Anglada, Alain y Garófalo Hernández, Abel. 2013. *Revista Cubana de Ciencias Informáticas. Revista Cubana de Ciencias Informáticas.* [En línea] 2013. http://scielo.sld.cu/scielo.php?pid=S2227-18992013000200006&script=sci_arttext. ISSN 2227-1899.

Apache. 2017. THE APACHE SOFTWARE FOUNDATION. *Apache Tomcat 7.* [En línea] 10 de mayo de 2017. <http://tomcat.apache.org/tomcat-7.0-doc/>.

Arenas, Enrique. 2015. Gestipolis. [En línea] 9 de julio de 2015. <http://www.gestipolis.com/tipos-y-clasificacion-de-los-sistemas-de-informacion-empresarial/>.

Astelehena. 2008. Observatorio Tecnológico. *Instituto Nacional de Tecnologías Educativas y Formación del Profesorado.* [En línea] 2008. <http://recursostic.educacion.es/observatorio/web/eu/software/servidores/580-elvira-mifsud>.

BERTINO, E. A. y MARTINO, L. A. 1995. *Sistemas de bases de datos orientadas a objetos.* s.l. : Ediciones Díaz de Santos, 1995, 1995.

Bruegge, Bernd y Dutoit, Allen. 2008. *Ingeniería se Software orientado a objetos.* 2008.

Choo, Chun. 2002. *Information management for the intelligent organization: the art of scanning the environment.* 2002.

Cruz Laguna, René Leandro, y otros. 2013. *Especificación de requisitos de software. Proyectos Sistema para la Gestión de Electos (SILECTOS v1.0).* 2013.

Davis, Gordon y Olson, Margrethe. 1984. *Management information systems: conceptual foundations, structure, and development (2nd ed.).* Medellín : Editorial Universidad de Antioquia, 1984. ISBN: 0-07-015828-2.

disca. 2012. disca. *disca.* [En línea] 2012. <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF..>

ExtJS, Programadores de. 2013. ExtJS. *ExtJS.* [En línea] 2013. <http://www.extjs.mx/>.

Fernández Baeza, Mario y Nohlen, Dieter. BIBLIOTECA CATÓLICA DIGITAL. [En línea] http://www.iidh.ed.cr/siii/index_fl.htm.

Gamma, Erich, y otros. 1994. TUTORIALSPPOINT. *SIMPLYEASYLEARNING*. [En línea] 1994. https://www.tutorialspoint.com/design_pattern/design_pattern_overview.htm.

García, Fabian. 2008. *Working at MINTIC-COMPUTADORES PARA EDUCAR*. 2008.

Generador Dinámico de Reportes. **Medina, Aldis Joan Abreu, y otros. 2012.** Cuba : s.n., 2012, Vol. 5.

Gutierrez, Damián. 2011. *Casos de Uso. Diagramas de Casos de Uso*. 2011.

Hill, Michael W. 2005. *The impact of Information on Society: An examination of its nature, value and usage*. London: Bowker-Saur : s.n., 2005. ISBN 3-598-11648-9.

IBM. 2009. IBM Knowledge Center. *IBM Knowledge Center*. [En línea] 2009. http://www.ibm.com/support/knowledgecenter/es/SSEPGG_8.2.0/com.ibm.db2.udb.doc/ad/c0009415.h.

Información tecnológica. **otros., Carlos A. Guerrero y. 2013.** 3, Colombia : Grupo de Investigación en Ingeniería del Software-GRIIS, 2013, Vols. Vol. 24 (3), 103-114 (2013).

INTECO, Laboratorio Nacional de Calidad del Software. 2010. *INGENIERÍA DEL SOFTWARE: METODOLOGÍAS Y CICLOS DE VIDA*. España : s.n., 2010.

Jackson, Wallace. 2014. *Setting Up Your Java 8 IDE: An Introduction to NetBeans 8.0. En Beginning Java 8 Games Development*. 2014.

Jacobson, Ivar. 2004. *"Use Cases - Yesterday, Today, and Tomorrow"*, Rational Software. 2004.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2004. *El Proceso Unificado*. 2004.

Jacobson, Ivar, Booch, Grady y Rumbaugh, Jim. 2007. OMG. [En línea] 2007. <http://www.omg.org/spec/UML/2.1.2/>.

Jaspersoft, Community. 2014. Jaspersoft Community. *Jaspersoft Community*. [En línea] 2014. <http://community.jaspersoft.com/documentation?version=33331>.

Kaneiwa, Ken, Mizoguchi, Riichiro y Nguyen, Philip HP. 2015. *A Logical and Ontological Framework for Compositional Concepts of Objects and Properties*. s.l. : New Generation Computing. 2015.

Larman, Craig. 2010. *UML y Patronos. 2a Edición*. 2010.

- Marin, Marvin David Arias. 2008.** Catedra de Programación. *Catedra de Programación*. [En línea] 16 de Octubre de 2008. <http://catedraprogramacion.forosactivos.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos>.
- Martínez, R. 2012.** Portal de Postgres en Español. [En línea] 2012. <http://www.postgresql.org.es/node/2984>.
- Mateu, Carles. 2004.** *Software libre. Desarrollo de aplicaciones web*. Barcelona : s.n., 2004. ISBN: 84-9788-118-4.
- MDN. 2015.** MDN. MDN. [En línea] 2015. <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introducci%C3%B3n>.
- Michaud, Frederic. 2015.** *Identifying Key Attack Surface Resources with Dynamic Analysis*. 2015.
- Moreiro, José Antonio. 1998.** *Introducción al estudio de la información y la documentación*. Medellín : Editorial de Antioquía, 1998.
- OMG. 2015.** Object Management Group. [En línea] marzo de 2015. <http://www.omg.org/spec/UML/2.5/>.
- ONE. 2011.** Proceso Electoral en Cuba. *Anuario Estadístico de Cuba 2010*. 2011, 2011.
- Oracle Corporation.** MySQL. *MySQL*. [En línea] <https://dev.mysql.com/doc/>.
- Paradigm, Visual. 2013.** Visual Paradigm for UML-UML tool for software. *Visual paradigm for uml*. [En línea] 2013. <https://www.visual-paradigm.com/features/>.
- Pavón, Juan. 2009.** *Estructura de las Aplicaciones Orientada a Objetos*. Madrid : s.n., 2009.
- Pegaso, Grupo. 2012.** Pegaso Tecnología. [En línea] 2012. <http://www.pegasotecnologia.com/tecnologia/aplicaciones.html>.
- PHP. 2011.** PHP. *PHP*. [En línea] 2011. <http://php.net/ChangeLog-5.php>.
- PostgreSQL, Global Development Group. 2014.** PostgreSQL. *PostgreSQL*. [En línea] 2014. <https://www.postgresql.org/about/press/presskit94/es/>.
- Potencier, Fabien. 2009.** *Symfony, la guía definitiva*. *Symfony, la guía definitiva*. [En línea] 2009. http://librosweb.es/libro/symfony_1_4/.
- Pressman. 2011.** Ingeniería del Software 6ta edición. [En línea] 2011. <http://ebookbrowse.com/roger-spressman-ingenieria-del-software-v-ed-pdf-d47183684>.

Prytherch, Ray. 2000. Harrod's Librarian's Glossary and Reference Book. Aldershot: Gower. [En línea] 2000.

https://books.google.com.cu/books?hl=es&lr=&id=aOEFDAAAQBAJ&oi=fnd&pg=PP1&ots=LqsCPJRjKp&sig=FBa3jbm4XgshhQbuR3GRQpPVEQc&redir_esc=y#v=onepage&q&f=false.

RAE. 2016. Real Academia española. *Diccionario de la Lengua Española*. [En línea] 2016. <http://dle.rae.es>.

Revista de derecho Extremadura. **Peralta CArrasco, Manuel. 2011.** 9, España : s.n., 2011. ISSN 1888-5519.

Revista Ingenierías Universidad de Medellín. **González Palacio, Liliana y Urrego Giraldo, Germán. 2010.** 17, Medellín : Publicación de la Facultad de Ingeniería, 2010, Vol. 9.

Revista Telem@tica. 2012. 1, 2012, Vol. 11. ISSN 1729-3804.

Robalino, Freddy. 2015. Gestipolis. [En línea] 9 de junio de 2015. <http://www.gestipolis.com/tipos-y-clasificacion-de-los-sistemas-de-informacion-empresarial/>.

Rodríguez, Tamara. 2014. *Metodología de Desarrollo para la Actividad Productiva de la UCI. Programa de Mejora.* 2014.

Rumbaugh, James, Jacobson, Ivar y Booch, Grady. 2009. *El Lenguaje Unificado de Modelado. Manual de Referencia.* España : s.n., 2009. ISBN: 84-7829-037-0.

Sencha. [En línea] www.sencha.com/.

SlideShare. [En línea]

Sommerville, I. 2005. *Ingeniería de software. 7ª edición.* 2005.

Sparx, Systems. 2007. Sparx Systems. Sparx Systems. [En línea] 2007. http://www.sparxsystems.com/resources/tutorial/use_case_model.html.

Symfony, Comunidad desarrolladora de. 2012. Symfony.es. [En línea] 10 de octubre de 2012. <http://symfony.es/noticias/2012/10/10/se-publica-symfony-1-4-19/>.

Symfony, Comunidad Desarrolladora de. 2014. Symfony.es. *Symfony.es*. [En línea] 4 de 11 de 2014. <http://symfony.es/noticias/2014/11/04/el-futuro-de-symfony-26-27-y-30/>.

The PostgreSQL Global Development Group. 2014. PostgreSQL. *PostgreSQL*. [En línea] 18 de diciembre de 2014. <https://www.postgresql.org/about/press/presskit94/es/>.

Wesley, Addison. 1999. *El proceso unificado de desarrollo de software.* 1999.

YAML. 2011. The Official YAML Web Site. *YAML.* [En línea] 2011. <http://yaml.org>.

Yañez, Julio. 2016. Código Xules . [En línea] 29 de enero de 2016. <http://codigoxules.org/creando-informes-java-jasperreports-desde-jaspersoft-studio/>.