

Universidad de las Ciencias Informáticas

Facultad 1



Trabajo de diploma para optar por el título de Ingeniero en
Ciencias Informáticas

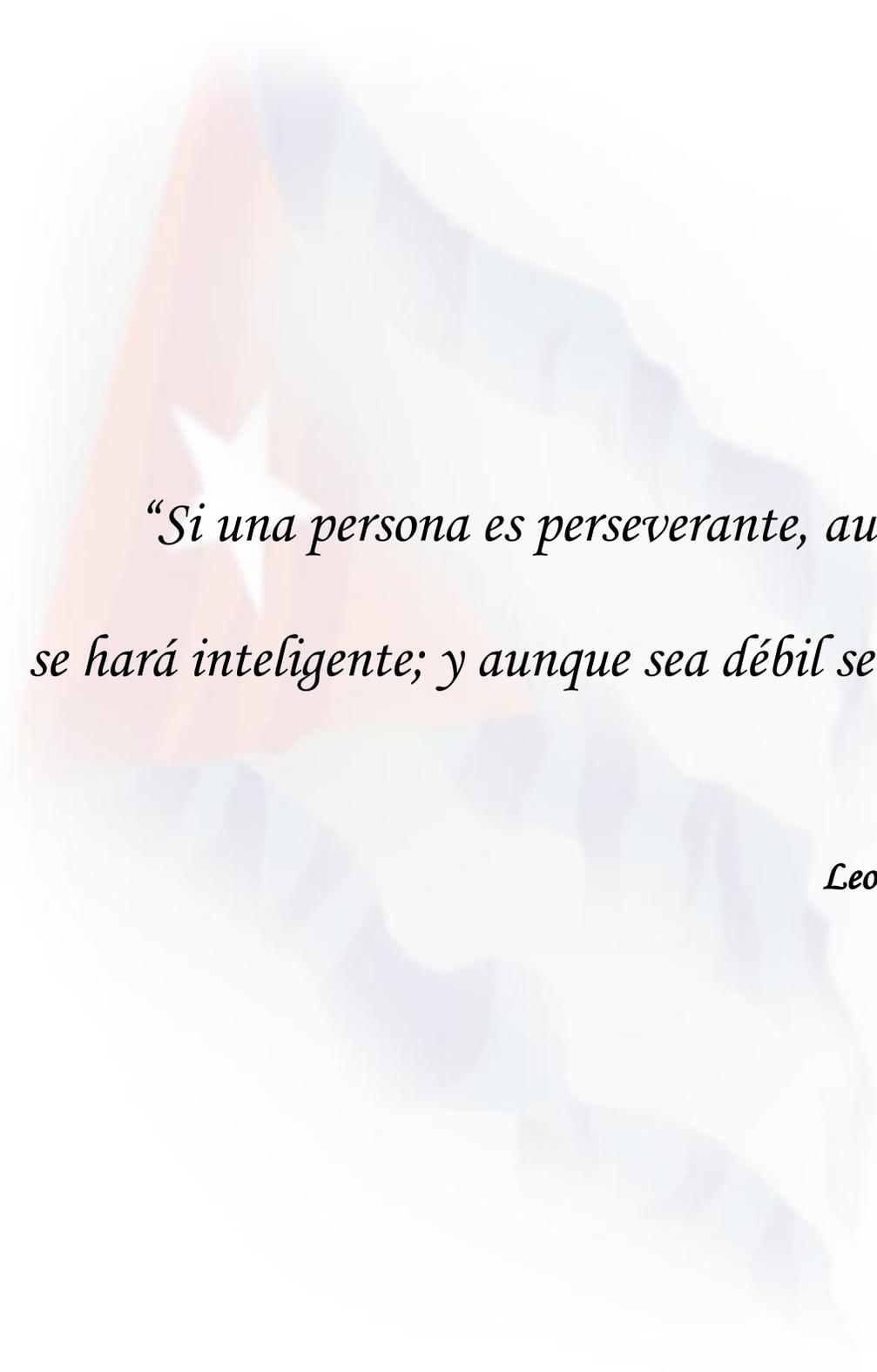
**Informatización del proceso de reprocesamiento del registro
electoral en el Sistema Único de Identificación Nacional
(SUIN)**

Autores: Alexis Medina Almarales
Rolando Fuentes Alarcón

Tutores: Ing. Denier Naranjo Oliva
Ing. Arnaldo Ramón Pedrero Varela

La Habana, Cuba

2013



“Si una persona es perseverante, aunque sea dura de entendimiento, se hará inteligente; y aunque sea débil se transformará en fuerte.”

Leonardo Da Vinci

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo con carácter exclusivo.

Para que así conste, firmamos el presente a los ____ días del mes ____ del año ____.

Firma del Autor

Alexis Medina Almarales

Firma del Autor

Rolando Fuentes Alarcón

Firma del Tutor

Ing. Denier Naranjo Oliva

Firma del Tutor

Ing. Arnaldo R. Pedrero Varela

Dedicatoria

A mis padres, porque creyeron en mí y porque me sacaron adelante, dándome ejemplos dignos de superación y entrega, porque en gran parte gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera, y porque el orgullo que sienten por mí, fue lo que me hizo ir hasta el final.

A Marilyn Rodríguez Oliva, por ser la persona más especial del mundo, por estar alentándome y aconsejándome cuando más lo necesitaba, por haberme hecho el hombre más feliz del universo aun estando lejos, y por encima de todo por darme la agraciada dicha de conocer un nuevo horizonte y convertirme en una persona más perfecta.

Esto va por ustedes.

Alexis Medina Almarales.

Le dedico este trabajo a esa persona que ha sabido ser madre y padre a la vez, esto es tanto tuyo como mío para ti madrecita Rebeca Alarcón Rodríguez.

Rofy

Gracias a Dios, por ser tantas veces fuerza y esperanza. Por cuidar de mí y de mis seres queridos.

A la Revolución y a Fidel, por la maravillosa oportunidad de ser parte de una Universidad que ha sido casa, escuela y lugar de impensados encuentros.

A mi maravillosa madre y mi espléndido abuelo, por su paciencia y su amor para todos, por aguantar tanto de la vida y aún tener fe.

A cada uno de mis profesores por su dedicación y entrega para ofrecerme sus conocimientos y convertirme en un digno profesional.

A esa familia maravillosa que me otorgó la vida por su constante preocupación en cada uno de mis logros y dificultades.

A mi tía Belkís, a mis tíos Luisi, Vicent, Raúl, a mis abuelitas Cada y Olivia, muchísimas gracias por su apoyo.

A cada vecino de mi barrio por siempre confiar en mí.

A todas las amistades de esta escuela, por darme esa fuerza para seguir adelante, tanto en las buenas como en las malas.

A Idia y Yiriam, por sus acertados consejos y su oportuna ayuda.

A Denier y Arnaldo, por representarnos y defendernos en todo momento y por ser los pilares fundamentales para hacer realidad esta obra, que hace un año atrás era un anhelado sueño.

A todos gracias.

Alexis Medina Almarales.

Les agradezco este logro a todas esas personas que me han sabido guiar y dar su apoyo en todos momentos:

A mis padres Rebeca y Rolando por estar siempre a mi lado cuando los necesite y por darme su apoyo en todo momento y ser un ejemplo digno de seguir.

A mi Abuelita, por ser como una madre, por creer siempre en mí, por sentirse orgullosa y por darme fuerzas para estudiar. Te adoro abuelita.

A mi prima Yuleikis y su novio Yaroslav por ser mis compañeros de eventos en todas mis escapadas de la escuela.

A mis tíos Marcia y David.

A mis hermanitos Rebequita y Ray.

A toda mi familia por confiar en mí y brindarme todo el cariño del mundo.

A mis amigos por estar conmigo en los buenos y malos momentos.

A mi compañero de tesis Alexis por saber soportarme y entenderme.

A mis Tutores Arnaldo y Denier por apoyarnos, defendernos y aconsejarnos en todo momento.

A todos los profesores que en algún momento me brindaron su ayuda.

Rofy

Resumen

En el año 2007, la Universidad de las Ciencias Informáticas (UCI) en colaboración con el Ministerio del Interior (MININT) desarrollan el Sistema Único de Identificación Nacional (SUIN) con el objetivo de registrar en un único sistema todos los datos identificativos de los ciudadanos cubanos y extranjeros residentes en Cuba.

Dicho sistema está integrado por diversos módulos entre los cuales se encuentra el Módulo del Registro Electoral, el cual comprende el proceso de reprocesamiento. Debido al gran cúmulo de información y el poco tiempo con que se cuenta para procesar la misma, se hizo prioridad para la Dirección de Identificación y Registro (DIR) la informatización de dicho proceso, el cual se describe en este trabajo de diploma como Informatización del proceso de reprocesamiento del registro electoral adjunto al Sistema Único de Identificación Nacional.

Dicho proceso tienen por objetivo lograr un acercamiento de los datos electorales a la realidad, actualizar las listas electorales de todas aquellas circunscripciones que pasan a segunda vuelta dentro de un período electoral, dar tratamiento a las inclusiones, inclusiones excepcionales y exclusiones, para así generar los nuevos resultados de las elecciones que serán emitidos.

La aplicación se desarrollará en un entorno orientado a procesos, haciendo uso de las tecnologías de *Microsoft*. La investigación abarca desde las actividades relacionadas con el modelado del negocio hasta la fase de prueba de las funcionalidades implementadas.

Palabras clave: exclusiones electorales, inclusiones electorales, inclusiones excepcionales, reprocesamiento electoral.

Introducción	1
Capítulo I: Fundamentación Teórica	5
Procesos eleccionarios.....	5
Inclusión.....	6
Exclusión.....	7
Reprocesamiento	7
Análisis de soluciones existentes y tecnologías asociadas.....	7
Sistema VOTESCRIPT	8
Sistemas Indra.....	8
Sistema Electoral Nacional (SEA).....	8
Ambiente de desarrollo.....	9
Conclusiones Parciales	20
Capítulo II: Análisis y Diseño	21
Flujo actual del proceso.....	21
Modelo de dominio	22
Vista global del proceso	23
Concepción de procesos mejorados.....	24
Requerimientos funcionales.	25
Requisitos no funcionales.....	30
Patrones de diseño	31
Patrones de workflow	34
Arquitectura de la solución	35
Descripción de la arquitectura	36
Especificación de clases	37
Servicios del sistema.....	41
Diseño de <i>workflow</i>	42

Transformación del modelo de procesos mejorados al workflow	43
Modelo de datos.....	45
Capítulo III: Implementación y Prueba	47
Estándares de codificación.....	47
Diagrama de componentes	48
Diagrama de despliegue.....	49
Interfaces del sistema.....	51
Diseño de casos de prueba	53
Resultado de las pruebas.....	56
Beneficios del proceso	57
Conclusiones Parciales	58
Conclusiones Generales.....	59
Recomendaciones	60
Referencias bibliográficas	61
Bibliografía Consultada	63
Anexos.....	65
Procesos mejorados.....	65
Descripción de requisitos funcionales.....	74
Casos de prueba	81

Introducción

El registro electoral es una lista de personas que constituye la base para hacer efectivo el derecho de los ciudadanos a ser electores y poder ser electos, teniendo en cuenta las distintas legislaciones, en el marco de un régimen de tipo democrático.

En Cuba, los procesos eleccionarios siempre han contado con el máximo apoyo de nuestro pueblo para garantizar con mayor rapidez y eficacia el flujo de información desde las estructuras bases hasta la Comisión Electoral Nacional. Estos procesos incluyen a todos los ciudadanos cubanos mayores de 16 años y que no presenten incapacidad mental con previa declaración judicial de esta, además de los incapacitados por causas penales (1). Además desde su implantación en 1976, instaura la inscripción de todos los ciudadanos con derecho al voto en el Registro Electoral Municipal propio (2).

A partir de la informatización de la mayoría de los procesos en Cuba, fue en las Elecciones Parciales del 2005 que se logró por vez primera automatizar de forma centralizada en cada provincia la información correspondiente a los Registros de Electores, lo cual permitió imprimirlos todos de forma homogénea y con una mayor calidad, además tenía por objetivo asegurar que a nivel provincial las personas solo estuviesen registradas una sola vez en dicho registro; luego en el 2007, la empresa cubana desarrolladora de software DATYS implementó el Sistema Electoral Nacional (SEA), el cual se mantiene en uso y conserva los mismos cánones del anterior.

Posteriormente, la Universidad de las Ciencias Informáticas (UCI), desarrolla el Sistema Único de Identificación Nacional (SUIN) en colaboración con el Ministerio del Interior (MININT), cuyo objetivo principal es la automatización de todos los procesos que se realizan en la Dirección de Identificación y Registro (DIR). Dentro de dicho sistema se encuentra el Módulo del Registro Electoral que se encarga de la gestión del Registro de Electores de la República de Cuba. Éste realiza los procesos de creación, actualización, control y supervisión del registro de electores del país.

Resaltando que en estos momentos se desarrolla para el SUIN la automatización de los procesos relacionados con el registro electoral, existe la necesidad de desarrollar dentro del Módulo del Registro Electoral, el proceso de reprocesamiento, debido a que en ocasiones las inclusiones y exclusiones de las personas en los registros electorales no son del todo fiables, por ejemplo; existen personas que por un motivo u otro deben viajar a otros lugares del país y no pueden votar en su ubicación electoral por lo que se

ven obligados a votar en el lugar donde se encuentran en ese momento y esta zona al incluirlo en su registro de electores puede provocar la duplicidad de dichos datos, que en la mayoría de las ocasiones propician problemas en la actualización de los registros. Han sucedido casos donde personas que no tienen derecho al voto, ya que están incapacitados penalmente o no posee la edad requerida para efectuar el sufragio, han aparecido en los registros electorales, por lo que esta situación puede traer serios problemas e incompatibilidades de información y emisión de estadísticas a la Comisión Electoral Nacional (CEN). Por lo que se hace necesaria la implementación de dicho proceso para garantizar la fiabilidad de las inclusiones y exclusiones como vía de aproximar los datos estadísticos a la realidad, la mayor exactitud en el conteo físico de los electores y una actualización de las listas electorales de aquellas circunscripciones que pasan a la segunda vuelta de elecciones.

Teniendo en cuenta lo antes expuesto, se plantea como **problema científico** a resolver: ¿Cómo compatibilizar la información emitida por la Comisión Electoral Nacional en el Registro Electoral en el Sistema Único de Identificación Nacional (SUIN), de manera que elimine la redundancia y garantice la fidelidad y exactitud de los datos?

Con vista a la solución del problema se plantea como **objeto de estudio**: Los procesos electorales dentro de un período electoral, como **objetivo general**: Informatizar el proceso de reprocesamiento del registro electoral del Sistema Único de Identificación Nacional (SUIN), para garantizar la fidelidad y exactitud de los datos, y se elimine la redundancia de los mismos, enmarcado en el **campo de acción**: El proceso de reprocesamiento en el registro electoral cubano.

Para complementar el objetivo de la investigación y solucionar la problemática planteada se proponen las siguientes **tareas investigativas**:

- Investigación sobre el proceso de inscripción de los ciudadanos con derecho al voto en el registro electoral según lo establecido por leyes cubanas.
- Investigación sobre el proceso de reprocesamiento del registro de electores según plantea la ley electoral cubana.
- Definición de soluciones ante posibles casos de inclusiones y exclusiones.
- Definición de reportes estadísticos que se puedan realizar con respecto a las situaciones tratadas.
- Definición de requerimientos funcionales y no funcionales del proceso a desarrollar.
- Investigación y análisis de herramientas para el desarrollo del proceso de reprocesamiento en cuestión.
- Diseño del *workflow* correspondiente al proceso que se desarrolla.

- Definición de las nuevas tablas en la base de datos.
- Análisis y diseño de la solución en dependencia de la metodología definida.
- Realización de pruebas funcionales al proceso implementado.

Los **métodos de investigación** utilizados se desglosan a continuación:

Métodos teóricos:

Histórico–lógico: permite hacer un estudio del estado del arte relacionada con la situación problemática de la investigación, así como analizar las ventajas y desventajas de las herramientas, tecnologías y metodología utilizadas actualmente en la informatización del proceso de reprocesamiento.

Analítico–sintético: Se utilizará para captar y resumir los documentos y procedimientos legales por los cuales se rige la DIR y la Asamblea Nacional en la gestión del Registro de Electores. De ellos se extraerán las ideas fundamentales y al mismo tiempo se detallará la información necesaria para una correcta fiabilidad en los procesos de negocio.

Modelación: Este método permite la creación de modelos (propuestas, alternativas y estrategias) que visualizan una reproducción simplificada de la realidad, consisten en descubrir y estudiar nuevas relaciones además de las cualidades del objeto de estudio.

Métodos empíricos:

Entrevista: Se utilizará la entrevista como una conversación planificada con los clientes para obtener información acerca del problema que se investiga. Su uso constituye un medio para el conocimiento cualitativo de las características particulares de un proceso y puede influir en el posterior análisis y diseño del producto de software.

Esta investigación estará estructurada en tres capítulos, donde se describe el trabajo realizado y los resultados obtenidos del mismo.

Capítulo I: Fundamentación Teórica. En este capítulo se estudian temas referentes a censos electorales, actualización de listas y conteo de votos de sistemas electorales y sus conceptos. Además se realiza un estudio de las diferentes herramientas a utilizar, donde se arriba a conclusiones que permiten dar respuesta a la problemática de la investigación.

Capítulo II: Análisis y diseño. Se realiza una valoración sobre el diagnóstico del campo de acción y de la propuesta solución. Se definen los principales requisitos a implementar, así como la arquitectura y los

patrones de diseño a utilizar. Contiene la construcción del sistema a partir de la realización de las tareas de ingeniería para darle cumplimiento a los requisitos planteados.

Capítulo III: Implementación y prueba. Contiene la descripción de los elementos establecidos para el proceso de desarrollo empleado, se define la estructura y ubicación de los componentes que se van a desplegar, se describen los casos de prueba que se le realizaron al sistema, así como los estándares de codificación empleados, con el objetivo de validar el correcto funcionamiento de la aplicación.

Capítulo I: Fundamentación Teórica

En el presente capítulo se abordarán los principales conceptos asociados al tema, la realización del estudio de los sistemas existentes para la gestión electoral y las tecnologías asociadas, además de sus ventajas aportadas al trabajo. También se fundamentarán la metodología y herramientas utilizadas para dar respuesta a la propuesta de solución.

Procesos eleccionarios

“Se conoce como proceso eleccionario a la serie continua y concatenada de actos complejos y con efecto preventivo, destinados a instrumentar y facilitar la realización de los comicios y la posterior asignación de cargos entre las distintas fuerzas políticas participantes en base al resultado por ellos obtenido” (3). Entre otras de sus definiciones se encuentra que son el conjunto de decisiones, actos, tareas y actividades que realizan los organismos político-electorales, los partidos políticos y los ciudadanos. Por su parte el COFIPE¹ define al proceso electoral como “el conjunto de actos ordenados por la constitución y este código, realizados por las autoridades electorales, los partidos políticos nacionales y los ciudadanos, que tiene por objetivo la renovación periódica de los integrantes de los poderes legislativo y ejecutivo de la unión” (4). En fin, en su totalidad está organizado por un organismo electoral que cada país estructura y nombra de disímiles formas, aunque básicamente sus responsabilidades sean similares.

Una visión panorámica del marco institucional en los países de América Latina nos permite verificar grandes diferencias. Por ejemplo, existen países como Colombia, Costa Rica y Panamá que han concentrado en una sola institución, la tarea del registro civil, identificación, cedulação y electoral. En Costa Rica, el Tribunal Supremo de Elecciones se compone de dos departamentos, uno Civil y el otro Electoral y en cada uno hay un oficial mayor que es responsable de su gestión. En otros como Haití, Paraguay y Venezuela, las responsabilidades se han dividido, correspondiendo al registro civil la tarea de inscripción de nacimientos y defunciones, a otra institución la identificación y cedulação de la población y a otra lo electoral.

En Cuba, el registro electoral es permanente, el cual se mantiene organizado y vigente aún en tiempos que no comprenden procesos electorales. De esta tarea se encargan las oficinas de Dirección de Identificación y Registro de la población, en conjunto con las demás instituciones de registro de ciudadanos como son el Registro Civil o la Dirección de Inmigración y Extranjería.

¹ *Código Federal de Instituciones y Procedimientos Electorales*

En resumen, se puede afirmar que el marco institucional es consecuencia de una voluntad política y que en los países de gran dispersión institucional el énfasis debe orientarse a mejorar las relaciones de coordinación, o sea, insistir en las interrelaciones, en tanto, en un marco institucional concentrado, el énfasis está en plantear estrategias de protección al ciudadano en cuanto a defender su privacidad y que su violación no se transforme en una violación a los derechos humanos.

En Cuba, La Ley No. 72 de 1992, Ley Electoral, establece que los procesos electorales son de dos tipos:

- a) Elecciones generales, en las que se elige a los Diputados a la Asamblea Nacional del Poder Popular, su Presidente, Vicepresidente y Secretario al Presidente, Primer Vicepresidente, Vicepresidentes, Secretario y demás miembros del Consejo de Estado, a los Delegados a las Asambleas Provinciales y Municipales del poder Popular y a su vez Presidentes y Vicepresidentes. Éstas se realizan cada cinco años.
- b) Elecciones parciales, en las que se elige a los Delegados a las Asambleas Municipales del Poder Popular y sus Presidentes y Vicepresidentes. Son realizadas cada dos años y medio.

El derecho al voto por cada elector está respaldado por el registro de electores al cual Marta León-Roesch² lo define como “el conjunto organizado de inscripciones de quienes reúnen los requisitos para ser elector y no se hallen privados, definitiva o temporalmente, del derecho al sufragio” (5).

Este concepto nos muestra que el objetivo del registro electoral no es solamente incorporar a la ciudadanía por la edad admitida de 16 años, sino que tener presente que existen otras restricciones, generalmente, que impiden integrar el registro, aunque éstas tengan carácter transitorio, como pueden ser, según las legislaciones, los condenados, ciudadanos residentes en el exterior e incapacitados mentales.

Tanta importancia tiene el registro electoral que el científico y político uruguayo Juan Rial, lo definen de la siguiente manera:

“El registro electoral es una pieza clave para asentar y consolidar un régimen democrático”.

En general en Cuba, el registro electoral es un archivo el cual guarda los datos identificativos de los electores, la condición electoral y la ubicación electoral con el objetivo de garantizar las listas electorales dentro de un período electoral.

Inclusión

Proceso mediante el cual una persona pasa a ser miembro en la lista electoral de la circunscripción a la que pertenece su domicilio o en casos excepcionales, de la lista electoral de la circunscripción especial a la cual

² *Escritora y asesora de los registros electorales en Argentina*

está vinculado en ese momento; y una vez que es miembro posee la capacidad de ser elector y tener derecho al sufragio.

Exclusión

Proceso mediante el cual una persona es privada de su derecho al sufragio producto a condiciones que no le hacen posible su participación social plena. Existen disímiles ejemplos donde las personas son excluidas de la lista electoral a la cual pertenecían privándolos de esta manera de su derecho al voto, como por ejemplo, aquellas personas que cumplen alguna sanción penal, los incapacitados mentales, los desertores del país, etc.

Reprocesamiento

Censo que se realiza luego de la primera vuelta de las elecciones con el objetivo de actualizar aquellas circunscripciones que pasan a segunda vuelta de dichas elecciones. Da tratamiento a las inclusiones y exclusiones para evitar resultados erróneos al finalizar el período electoral, trata de lograr un acercamiento a la realidad de los datos electorales, generando de esta forma los nuevos resultados de las elecciones.

Análisis de soluciones existentes y tecnologías asociadas

La incorporación de las tecnologías en los procesos electorales ha generado gran interés alrededor del mundo. Tal es así que los países más desarrollados las han acatado con tal fuerza que actualmente sus sistemas electorales dependen totalmente de estas nuevas tecnologías, las cuales se usan para compilar las listas de electores, diseñar la geografía electoral, administrar y capacitar al personal, imprimir las papeletas, conducir las campañas de educación al electorado, registrar la emisión del voto, escutar y consolidar los resultados de la votación y publicar los resultados electorales. La aplicación adecuada de la tecnología en las elecciones puede incrementar el margen de eficiencia administrativa, reducir los costos a largo plazo y mejorar el marco de transparencia política. Estas pueden incluir tanto aquellas que son de reciente aparición como las que ya se utilizaron con anterioridad, como prensas de impresión, plumas, máquinas de escribir manuales, calculadoras electrónicas y radios, o las llamadas nuevas tecnologías como computadoras, escáneres ópticos, sistemas de mapeo digital e internet. El nivel de complejidad de la tecnología que se aplica para la administración electoral a lo largo del mundo varía significativamente debido a que para organizar una elección moderna a gran escala, puede representar un reto significativo para países que no tienen acceso a los avances tecnológicos.

Sistema VOTESCRIPT

Sistema de voto telemático basado en criptografía avanzada y en el empleo de tarjetas inteligentes. Basa todo su funcionamiento en llaves públicas y privadas contenidas en dichas tarjetas inteligentes. Para la realización del conteo de votos el Gestor del Sistema Administrador y los Interventores con sus tarjetas inteligentes, proporcionarán de forma conjunta al contador automático su clave privada, la cual ha permanecido guardada y oculta hasta ese momento, necesaria para que el contador entre en funcionamiento. Luego de recibir toda la información procedente de la urna, el contador descifra cada uno de los votos a través de sus distintas llaves privadas y verifica que esté correcta la firma del Administrador y los Sistemas de Intervención para la realización del conteo y la publicación de los resultados.

Aunque este sistema sería entre tantas opciones uno de los mejores para implementar en el sistema electoral cubano debido a la integridad y confiabilidad de la información, tiene el inconveniente de que en estos momentos el país no se encuentra económicamente preparado para adquirir dicha tecnología.

Sistemas Indra

Es uno de los sistemas de avanzada utilizado en los países desarrollados con vista al perfeccionamiento de los procesos electorales. El conteo de los votos parte de una combinación de captura de imágenes y datos que aporta mayor fiabilidad, velocidad y transparencia al proceso. Utiliza identificadores biométricos, llaves públicas y privadas y urnas electrónicas dotadas de pantalla táctil con el objetivo de realizar un proceso de recuento más fiable y eficiente además de lograr mayor rapidez en la obtención de los resultados electorales; también entre sus principales ventajas esta la transparencia e información en todas las etapas del proceso y la eliminación de los procesos manuales.

Si bien, estos sistemas serían la solución al sistema electoral cubano, cuenta con la desventaja de contar con tecnologías muy sofisticadas a las cuales Cuba no le es posible adquirir debido a su alto costo en el mercado internacional.

Sistema Electoral Nacional (SEA)

Desde el año 2007, Cuba viene utilizando este sistema para los procesos electorales. Implementado por la empresa cubana desarrolladora de software DATYS, con el objetivo de gestionar los procesos electorales en la DIR. Comprende el módulo cliente, el cual da la posibilidad de especificar las direcciones electorales, la búsqueda de personas, establecer los reportes estadísticos y de auditorías; además cuenta con el módulo de administración donde se pueden salvar, exportar y revisar los documentos, da la posibilidad de conectar

y desconectar las distintas bases de datos y la administración de los distintos usuarios y el acceso de estos a archivos externos.

No obstante, este sistema aunque hasta el momento ha servido de mucha ayuda para la gestión de los procesos electorales cubanos, cuenta con la deficiencia de una base de datos independiente a la del Carné de Identidad, lo que trae consigo que puedan existir dos personas semejantes cuando en realidad es la misma y se traten de manera arbitraria, además de esto, todos los ingresos de los datos son realizados por el administrador del sistema, por lo que la información no es del todo confiable, a la misma vez que utiliza tecnologías desactualizadas, su interfaz es poco amigable, difícil y posee poca integridad en sus datos estadísticos. Además, dicho sistema no guarda en base de datos todos aquellos electores que fueron censados, ni los períodos electorales en que fueron censados, por lo que es imposible establecer comparaciones con próximos períodos electorales de si el elector ya existía o no y también es incapaz de comprobar si se realizó el proceso de reprocesamiento de un período a otro.

Ambiente de desarrollo

A continuación se caracterizan la metodología, tecnologías y herramientas que permitirán la realización de la solución propuesta, las cuales fueron definidas por las políticas del proyecto. Para la confección de cualquier sistema, se hace necesario conducir todo este proceso a través de una metodología la cual representará una guía que va indicando qué hacer y cómo actuar cuando se quiere obtener algún tipo de investigación.

MSF³ for CMMI⁴ Process Improvement

MSF for CMMI Process Improvement es una metodología ágil de desarrollo de software, que cumple con los requerimientos para el nivel tres de **CMMI**. Define cinco fases durante el ciclo de vida del proyecto que encapsula flujo de actividades y actividades. Las fases son visión, planificación, desarrollo, estabilización e implantación. Cada punto de control proporciona una oportunidad para autorizar el trabajo, continuar o cancelar el proyecto (6).

Visión: obtener la visión y el alcance del proyecto, el cual debe estar compartido, comunicado, entendido y alineado con los objetivos del negocio. En esta fase el equipo y el cliente integran el proyecto y definen los requerimientos funcionales, sus alcances y restricciones, identifican los beneficios del proyecto y también

³ *Microsoft Solution Framework*

⁴ *Capability Maturity Model Integration*

los riesgos del proceso. La fase culmina con el hito Visión y Alcance aprobados. Dentro de los entregables de esta fase se encuentran: el Documento de Visión y Alcance y la Matriz de Identificación de riesgos.

Planificación: el equipo del proyecto creará un borrador del plan maestro del proyecto, además de la especificación funcional del proyecto y un cronograma que identifica puntos de control específicos. Esta fase culmina con el hito Plan del Proyecto (Especificaciones y Cronograma) aprobado. Dentro de los entregables de esta fase se encuentran: Documento de Especificaciones, Cronograma del proyecto y Documentos de proceso de licitación aprobados de acuerdo a la Norma Operativa vigente.

Desarrollo: involucrar la serie de comunicadores internos o entregas parciales del producto, desarrollados por partes para medir su progreso y para asegurarse que todos sus módulos o partes están sincronizados y pueden integrarse. La fase culmina con el hito Alcance completo. Dentro de los entregables de esta fase se encuentran: Código fuente y ejecutables, Actas de aceptación de entregas parciales, Plan de Pruebas, Manual de Instalación y Operación, Documento de Arquitectura, Actas de control de cambios aprobadas.

Estabilización: centrarnos en probar el producto. El proceso de prueba hace énfasis en el uso y el funcionamiento del producto en las condiciones del ambiente real. La fase culmina con el hito Aceptación de Pruebas. Dentro de los entregables de esta fase se encuentran: Acta de Aceptación de Pruebas, Acta de capacitación a usuarios y Mesa de Servicios, Acta de Entrega.

Implantación: en esta fase el equipo implanta la tecnología y los componentes utilizados por la solución, apoya el funcionamiento y la transición del proyecto, y obtiene la aprobación final del cliente. En ocasiones en esta fase se ejecutan planes piloto de implementación. La fase termina con el hito Cierre de la Entrega. Dentro de los entregables de esta fase se encuentran: Acta de Implantación, Encuesta de satisfacción del cliente y Acta de Cierre de proyecto.

Dicha metodología lideró en cada una de sus fases el proceso de reprocesamiento para de esta forma lograr una organización y una planificación en el desarrollo y entrega de los artefactos generados en todo el ciclo de creación de la solución.

Altova UModel

Herramienta de modelado que permite crear e interpretar diseños de software mediante la potencia del estándar UML⁵ 2.1. Dibujar su diseño de aplicación y generar código Java o C# a partir de sus planos, o

⁵ Lenguaje Unificado de Modelado.

hacer ingeniería inversa de programas existentes a diagramas UML claros y precisos para abarcar rápidamente su arquitectura de software. Puede corregir el código generado o sus modelos y completar la ronda produciendo automáticamente nuevos diagramas o regenerando el código (7).

Dentro de sus principales características están:

- Soporte para los 14 tipos de diagramas UML.
- Generación de código fuente en lenguajes Java, C#, y VB.NET.
- Creación de diagramas de secuencia desde el código fuente de la ingeniería inversa.
- Generación de documentación personalizable de proyecto.
- Integración con sistemas de control de versiones.
- Ofrece un diagrama para esquemas XML similar a UML y diagramas para modelado de procesos de negocio (BPMN).
- Compatibilidad con el estándar XML 2.1 de intercambio de modelos UML.
- Funciones para arquitectura dirigida por modelos.
- Estrecha integración con Visual Studio y Eclipse.

Esta herramienta permitió generar los diferentes procesos mejorados asociados a los ya existentes con el objetivo de lograr una solución más avanzada del producto. Además posibilitó la creación del diagrama de componentes y el de despliegue.

Business Process Management Notation (BPMN)

Nuevo estándar de modelado de los procesos de negocio capaz de representar gráficamente las diferentes etapas del mismo. Esta notación ha sido diseñada para coordinar la secuencia de eventos y los mensajes que fluyen entre los diferentes procesos participantes, haciendo de BPMN un estándar capaz de expresar más patrones que los diagramas de actividad, pues es gráficamente más rico y con menos símbolos fundamentales, lo que facilita su comprensión por parte del personal no experto; además cubre casi totalmente los patrones de *workflow* con lo cual se le supone una gran expresividad a la hora de especificar procesos (8).

Con BPMN fueron conformados los procesos mejorados y su utilización está orientada específicamente al negocio.

CSharp (C#)

Es un lenguaje de programación orientado a objetos⁶, diseñado por la empresa de software *Microsoft* para ser utilizado en la plataforma .NET. Dentro de sus principales ventajas se encuentran:

- Sencillo ya que elimina elementos que otros lenguajes incluyen y que son innecesarios en .NET.
- El código escrito en C# es auto contenido lo que significa que no necesita de ficheros adicionales tales como ficheros de cabecera.
- Es orientado a componentes lo que permite definir cómodamente propiedades (similares a campos de acceso controlado), eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros).
- Robusto ya que provee un ambiente elegante y unificado, el manejo de errores está basado en excepciones, soporta los conceptos como encapsulación, herencia y polimorfismo de la programación orientada a objetos y soporta los modificadores de acceso *private*, *protected*, *public* y agrega un cuarto modificador *internal*.
- Eficiente debido a que da la posibilidad de saltarse algunas restricciones de seguridad manipulando objetos a través de punteros al declarar regiones de código como inseguras y hacer uso en ellas de punteros de modo similar a como se utilizan en C++.

Predefinido como lenguaje de desarrollo, permitió la implementación de todos los métodos y clases asociados a la solución.

Microsoft Visual Studio 2010

Es un conjunto de herramientas que simplifica el desarrollo de aplicaciones para desarrolladores individuales o equipos, haciendo posible la creación de aplicaciones escalables y de alta calidad. Proporciona nuevos beneficios, como la capacidad de optimizar su entorno de desarrollo con compatibilidad para varios monitores y la capacidad de soportar varias versiones de *.NET Framework* con una sola herramienta. Además, Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de *.NET Framework*, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML. Sobre dicha

⁶ Se basa en la idea de un objeto, que es una combinación de variables locales y procedimientos llamados métodos que juntos conforman una entidad de programación.

plataforma de desarrollo se implementaron los servicios, los métodos, las clases, las interfaces y se realizaron las pruebas unitarias haciendo uso del método *UnitTest()*.

Framework .Net 4.0

Es un componente integral de *Windows* que admite la compilación y la ejecución de la siguiente generación de aplicaciones y servicios Web. Los componentes clave de *.NET Framework* son Common Language Runtime (CLR) y la biblioteca de clases *.NET Framework*, que incluye ADO.NET, ASP.NET, formularios Windows Forms y *Windows Presentation Foundation (WPF)*. *.NET Framework* proporciona un entorno de ejecución administrado, una implementación y desarrollo simplificado y la integración con una gran variedad de lenguajes de programación (9).

Este *framework* presenta nuevas mejoras y características que lo hacen diferente a sus versiones anteriores, aunque posee la capacidad de funcionar en paralelo con ellas, dentro las que se encuentran:

- Mejoras en *Common Language Runtime (CLR)*, ASP.Net e innovaciones en el lenguaje CSharp. *Entity Framework*, donde se permite a los desarrolladores programar con bases de datos relacionales usando objetos .NET y Language Integrated Query (LINQ).
- Servicios de datos de *Windows Communication Foundation (WCF)*, componente que permite crear servicios y aplicaciones que usen protocolo de datos abierto para exponer y usar datos a través de la web.
- Más control sobre HTML, identificadores de elementos y hojas CSS personalizadas que facilitan enormemente la creación de formularios Web que admiten optimización del motor de búsqueda y son conformes a los estándares.
- Compatibilidad con formularios para nuevas mejoras en la librería AJAX (10).

Bison Framework

Es un *framework* para la orquestación de procesos de negocio con *Windows Workflow Foundation (WWF)*. Su principal objetivo es proporcionar un componente que permita gestionar las instancias de *workflow*. Además encapsula un conjunto de actividades y servicios que le dan mayor dinamismo al desarrollo de sistemas centrado en la orquestación de procesos de negocio con WF, específicamente para un ambiente web. Entre las ventajas que brinda *Bison Framework* se pueden mencionar:

- Proporciona una mayor aproximación a los usuarios de negocio.

- Brinda rapidez y flexibilidad para modelar y cambiar los procesos según las necesidades.
- Aporta escalabilidad o capacidad de crecer.
- Fortifica el puente creado por el *workflow* para la comunicación entre el analista y el desarrollador.
- Propone una arquitectura donde se encuentran bien definidas las capas de presentación y negocio.
- Posee actividades y servicios especializados en la orquestación de interfaces de usuario, que permiten definir su flujo de una manera gráfica dentro del *workflow*.
- Uno de los principales objetivos del framework es lograr modelar la mayor parte de la lógica de negocio de forma gráfica en un workflow. (11)

Ajax

Por sus siglas en inglés significa *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML) y la importancia de ésta técnica está en que al utilizarla, ésta se ejecuta en el lado del cliente – es decir el navegador Web-, pero por debajo mantiene una comunicación asíncrona con el servidor. Con ésta asincronía utilizando XML, cuando se desee hacer algún cambio en el desarrollo Web, no será necesario que la página se recargue nuevamente; lo que en últimas se traduce en aplicaciones mucho más interactivas, rápidas, veloces y eficientes. Esta no es una tecnología en sí misma, sino que se trata de varias tecnologías independientes que se unen para conformarla.

Las tecnologías que forman AJAX son:

- ✓ XHTML y CSS, para crear una presentación basada en estándares.
- ✓ DOM, para la interacción y manipulación dinámica de la presentación.
- ✓ XML, XSLT y JSON, para el intercambio y la manipulación de información.
- ✓ XMLHttpRequest, para el intercambio asíncrono de información.
- ✓ JavaScript, para unir todas las demás tecnologías. (12)

Esta técnica es útil para aumentar la velocidad y el rendimiento de la aplicación, por ejemplo, cuando se están cargando en los componentes visuales los municipios dada una provincia no sea necesario recargar la página completa nuevamente y solo se recarga el componente especificado mediante la comunicación asincrónica de Ajax.

ASP.NET

Es la parte del .NET Framework dedicada al desarrollo web. A través del servidor web (IIS)⁷ nuestras aplicaciones ASP.NET se ejecutarán bajo el *Common Language Runtime (CLR)* y podremos usar el conjunto de clases del .NET Framework para desarrollarlas, obteniendo así una versatilidad y una potencia nunca antes conseguida en las aplicaciones ASP.

Windows Communication Foundation (WCF)

Es un componente de *Microsoft .NET Framework* que define un solo enfoque para la creación de servicios de aplicaciones empresariales que permite unificar tecnologías existentes como *Web Services (ASMX)*, *.NET Remoting*, *Microsoft Message Queue (MSMQ)*, *Enterprise Services (COM +)* y *Web Services Enhancements (WSE)*. Cabe recalcar que no reemplaza estas tecnologías sino que provee un único modelo de desarrollo que utilice todos los beneficios de estas tecnologías en una sola aplicación. Con *WCF* es posible independizarse de los mecanismos de transporte y de los protocolos, ya que se puede tener un único servicio programado de una forma particular y exponerlo de diferentes formas. Con dicha tecnología se logró la implementación de los servicios de negocio.

Windows Workflow Foundation (WWF)

El término *workflow* se refiere a toda un área dentro de la informática cuyo objetivo es el modelado y automatización de procesos que manejan información normalmente no estructurada dentro de una organización o entre distintas organizaciones. Esta información, habitualmente en forma de documentos, fluye a través de un grupo de personas y/o máquinas que participan en el proceso trabajando sobre dichos documentos (13).

Windows Workflow Foundation es una plataforma que permite a los usuarios crear un flujo de trabajo en sus aplicaciones. Consiste en un espacio de nombres, un motor de flujos de trabajo y diseñadores para Visual Studio que pueden ser utilizados para modelar escenarios simples como sería mostrar controles de interfaz de usuario basados en entradas de usuarios o escenarios complejos como el procesamiento de pedidos y control de inventarios. *WWF* viene con un modelo de programación extensible y un diseñador para construir actividades que encapsulan la funcionalidad del flujo de trabajo para usuarios finales o para ser reutilizado a través de múltiples proyectos (14).

Las principales funcionalidades que *WWF* provee son:

⁷ Del inglés *Internet Information Services*. Es un servidor web y un conjunto de servicios para el sistema operativo *Windows*.

- Asignar actividades a las personas de forma automática y según cualquier criterio o carga de trabajo.
- Optimizar la colaboración entre personas que comparten actividades.
- Automatizar o controlar el flujo de documentos, datos e imágenes.
- Proveer métricas para responsables de áreas, organizadores, gestores de procesos y calidad tanto para efectos de mejora continua como de indicadores de calidad y gestión.

Con la utilización de esta tecnología se logró modelar todos los flujos de trabajo automatizados de la solución.

Oracle Database 11g

Oracle Database 11g es uno de los gestores de base de datos más utilizado en el mundo. Permite a las empresas tener el control de su información, garantiza una mejor vista interna del negocio y adaptarse de manera rápida y confiable al entorno competitivo en constante cambio de hoy en día. Para lograrlo, esta versión amplía las capacidades exclusivas de Oracle de agrupación de base de datos, automatización de centro de datos y gestión de carga de trabajo. Con redes altamente disponibles y escalables en servidores de bajo costo y sistemas de almacenamiento, los clientes de Oracle pueden hacer frente al procesamiento de transacciones, almacenamiento de datos y aplicaciones de gestión de contenido, más exigentes. (15)

Oracle Database 11g es el primer gestor de base de datos en el mundo en incluir funcionalidades que permiten hacer pruebas de cambios en aplicaciones simulando las cargas reales generadas por los usuarios en los entornos de producción. Ofrece a los desarrolladores la posibilidad de elegir herramientas de desarrollo y un mejor proceso de desarrollo de aplicaciones que aprovecha al máximo las funciones clave de este. Las cuales incluyen nuevas funciones, tales como almacenamiento en caché del lado del cliente, XML binario para un desempeño de aplicaciones más rápido, procesamiento XML y el almacenamiento y recuperación de archivos. Incluye integración nativa con *Visual Studio 2010* para desarrollar aplicaciones .NET sobre *Oracle*; herramientas de migración de acceso con *Oracle Application Express*⁸; y creación de consulta fácil *SQL Developer* para una codificación veloz de las rutinas SQL y PL/SQL.

Oracle Database 11g incluye importantes mejoras en el desempeño para XML DB, una función de la base de datos de *Oracle* que permite a los clientes almacenar y manipular datos XML de forma nativa. Se ha incorporado soporte para XML binario, lo cual hace posible que los clientes puedan elegir opciones de almacenamiento XML para cumplir con los requerimientos específicos de desempeño y de aplicaciones.

⁸ *Oracle Application Express* o *APEX* es una herramienta de desarrollo rápido de aplicaciones que se ejecuta con una base de datos *Oracle*.

Dentro de las características que lo definen como uno de los mejores gestores de bases de datos en el mundo se encuentran que: en cuanto a soporte se encuentra disponible sobre varias plataformas (*Windows, Linux, Unix*), es fiable ya que presenta una continua disponibilidad de las aplicaciones y datos, es escalable ya que puede adaptarse al crecimiento de trabajo de manera fluida y hacerse más grande sin perder calidad en los servicios ofrecidos, brinda seguridad y protección de datos permitiendo compartir la red de recursos de una empresa con la confianza de que la privacidad se mantiene y es auto-gestionado debido a que automatiza muchas de las funciones de infraestructura de modo que una sola persona puede administrar cientos de servidores.

ADO.NET Entity Framework

Entity Framework es un conjunto de tecnologías de *ADO.NET*⁹ que permite a los programadores trabajar con datos en forma de objetos y propiedades específicas del dominio, por ejemplo, con clientes y direcciones, sin tener que pensar en las tablas de las bases de datos subyacentes y en las columnas en las que se almacenan estos datos, elevando así el nivel de abstracción en la que los programadores pueden trabajar al tratar con datos y reduciendo el código requerido para crear y mantener las aplicaciones orientadas a datos.

Las aplicaciones de *Entity Framework* ofrecen las siguientes ventajas:

- Las asignaciones entre el modelo conceptual y el esquema específico de almacenamiento pueden cambiar sin tener que cambiar el código de la aplicación.
- Los programadores pueden trabajar con un modelo de objeto de aplicación coherente que se puede asignar a diversos esquemas de almacenamiento, posiblemente implementados en sistemas de administración de base de datos diferentes.
- Se pueden asignar varios modelos conceptuales a un único esquema de almacenamiento.
- La compatibilidad con LINQ¹⁰ proporciona validación de la sintaxis en el momento de la compilación para consultas en un modelo conceptual. (16)

La utilización de este modelo entidad-relación (ORM) posibilitó el mapeo de las tablas de la base de datos, facilitando que cada una de estas tablas fueran tratadas como un objeto.

⁹ Es un conjunto de componentes del software que pueden ser usados por los programadores para acceder a datos y a servicios de datos. Es una parte de la biblioteca de clases base que están incluidas en el Microsoft .NET Framework.

¹⁰ *Language Integrated Query.*

LINQ

El acrónimo LINQ define un lenguaje que brinda la posibilidad de expresar, de una manera uniforme e integrada, consultas contra las más diversas fuentes de datos que éstos utilizan en sus aplicaciones (colecciones de objetos en memoria, documentos XML o bases de datos relacionales). Es un modelo de programación que simplifica y unifica la implementación de acceso a cualquier tipo de dato. LINQ no impone a usar una arquitectura específica, más bien facilita la implementación de varias arquitecturas existentes para acceso a datos.

Las ventajas más importantes que aporta LINQ son:

- Permite reducir a una mínima expresión el desajuste de impedancias¹¹ en la programación actual.
- Permite elevar el nivel de abstracción y claridad de la programación y en gran medida indicar únicamente qué se desea obtener, dejando a un motor de evaluación de expresiones los detalles sobre cómo lograr ese objetivo.
- Seguridad mejorada gracias a la verificación de sintaxis y de tipos por parte de los compiladores.

(17)

Permitió elaborar consultas para la obtención de los datos con formas estructuradas y simples.

Embarcadero ER/Studio 8.0

Herramienta para el modelado de datos que permite documentar y reutilizar los activos en datos. Dentro de sus características se encuentran que existe una verdadera separación de modelos físicos y lógicos con integración completa, transformación automatizada y mapeo de tipos personalizados de datos. Es capaz de generar esquemas XML desde modelos físicos y lógicos, posee soporte completo al ciclo de vida de bases de datos, además de tener ingeniería inversa y directa. Cuenta con generación automatizada de código de bases de datos, almacén de datos y soporte a la integración. (18)

Haciendo uso de dicha herramienta se logró elaborar el modelo de datos y sus relaciones con los demás procesos.

Crystal Report

¹¹ Existencia de limitaciones al transmitir información de un medio a otro.

Crystal Reports ha formado parte de *Visual Studio* desde 1993, y ahora es el estándar de elaboración de informes de *Visual Studio*. Se incluye en todas las copias de *Visual Studio Professional* y se integra directamente en el entorno de desarrollo.

Crystal Reports para *Visual Studio* incorpora la posibilidad de crear contenido interactivo con calidad de presentación al entorno de *Windows*. Con *Crystal Reports* para *Visual Studio*, puede crear informes complejos y profesionales en un programa basado en GUI¹². Después puede conectar el informe a casi todos los orígenes de base de datos, así como a datos proxy, como un conjunto de resultados (por ejemplo, un ADO.NET *DataSet*). Los asistentes del diseñador de GUI le permiten establecer fácilmente los criterios de formato, agrupamiento y gráficos.

Puede almacenar el informe en una aplicación Web o para *Windows*, con uno de los controles de visores de *Crystal Reports* para *Visual Studio*. La presentación de informes, tanto en clientes *Windows* como en HTML 3.2 ó 4.0, es muy interactiva y proporciona funciones como la profundización en gráficos, la exploración de informes y la búsqueda de texto. (19)

Con la utilización de dicho estándar se lograron crear diferentes tipos de reportes como por ejemplo, los resultados del proceso de reprocesamiento por provincia y por municipio, además de los datos de las comisiones electorales.

Team Foundation Server 2010

Es la plataforma de colaboración central en la solución de administración del ciclo de vida de las aplicaciones de *Microsoft*, que automatiza el proceso de entrega de *software* y permite a las organizaciones administrar proyectos de desarrollo de *software* de forma eficiente durante su ciclo de vida. Además permite al equipo ser más ágil, colaborar de un modo más eficaz y ofrecer *software* de más calidad, así como adquirir y compartir conocimiento institucional. Se usa como controlador de versiones para la asignación de tareas y control de estas.

Los datos y artefactos del proyecto de seguimiento de elementos de trabajo, control de código fuente, compilaciones y herramientas de pruebas se acumulan en un almacén de datos y eficaces informes que proporcionan tendencias históricas, seguimiento completo y visibilidad en tiempo real de la calidad y el progreso respecto al propósito empresarial. (20)

¹² *Interfaces de usuario*

Conclusiones Parciales

Luego del estudio realizado sobre algunos de los sistemas que en materia de censos electorales, actualización de listas y conteo de votos, de las ventajas y desventajas que cada uno tiene y atendiendo al problema científico planteado al inicio de esta investigación, además de tener en cuenta los grandes costos que tendrían para el país la adquisición de cualquiera de los sistemas mencionados anteriormente, se determinó la necesidad de desarrollar para el Sistema Único de Identificación Nacional, el proceso de reprocesamiento, el cual se encuentra dentro del Módulo del Registro Electoral. Además, se hizo un estudio de las herramientas aquí argumentadas, demostrando que cuentan con el nivel de actualidad y características adecuadas para ofrecer un sistema que responda al problema en cuestión.

Capítulo II: Análisis y Diseño

En el presente capítulo se abordarán los elementos relacionados con el modelado del negocio del proceso de reprocesamiento que se lleva a cabo actualmente en las oficinas de la DIR, lo cual permitirá obtener una idea precisa de cómo se desarrollan actualmente los procesos en dichas oficinas y posibilitará además, establecer las pautas para la informatización de las distintas actividades asociadas a este proceso. Se realiza el levantamiento de requisitos que contiene la definición de los requisitos funcionales y no funcionales del proceso a desarrollar y se definen los prototipos de interfaz gráfica. Se realiza el modelo de diseño del sistema que se va a implementar, además de definir y especificar el conjunto de clases a utilizar. Se describe la arquitectura a utilizar, los patrones para el desarrollo del sistema y se presentan los servicios del mismo y el modelo de datos usado.

Flujo actual del proceso

El registro electoral comprende disímiles procesos, todos ellos con el propósito de mantener actualizados los datos de los electores. Entre estos se encuentra el proceso de reprocesamiento, el cual da tratamiento a las inclusiones y exclusiones dentro de un período electoral de todos los ciudadanos considerados electores, trata de lograr un acercamiento de los datos electorales a la realidad, además de generar los nuevos resultados de las elecciones y actualizar todas aquellas circunscripciones que van a segunda vuelta de dichas elecciones. Actualmente, la Comisión Electoral Nacional envía los datos estadísticos y los padrones electorales usados en la primera vuelta de las elecciones hacia las oficinas de la DIR, donde estos son procesados por las oficinas especializadas en las tareas acerca del registro electoral con el objetivo de lograr una mayor exactitud en los datos. Una vez culminado el proceso de las inclusiones, inclusiones excepcionales y exclusiones de todas las listas electorales de cada una de las circunscripciones envueltas en dicho proceso, se actualizan estos datos enviados por la Comisión Electoral Nacional, se emiten dichos datos y por último se actualizan las listas electorales de aquellas circunscripciones que van a segunda vuelta.

En este momento, entre los datos estadísticos que son enviados por la Comisión Electoral Nacional se encuentran los modelos de inclusión e inclusión excepcional, donde están contenidos todos los nuevos electores que fueron agregados en el proceso electoral, además las exclusiones son tachadas dentro de las listas electorales especificando la causa de dicha exclusión.

Modelo de dominio

Luego de haber realizado un estudio sobre las dificultades existentes en el SEA, se presenta el siguiente modelo de dominio, con el objetivo de tener una idea más precisa del proceso y establecer soluciones que conlleven a eliminar todas las imperfecciones del proceso de reprocesamiento que es realizado por dicho sistema para un mejoramiento de las actualizaciones de las listas electorales y la generación de los nuevos resultados de aquellas circunscripciones que van a segunda vuelta dentro de un período electoral. (Ver Fig.1 Modelo de Dominio)

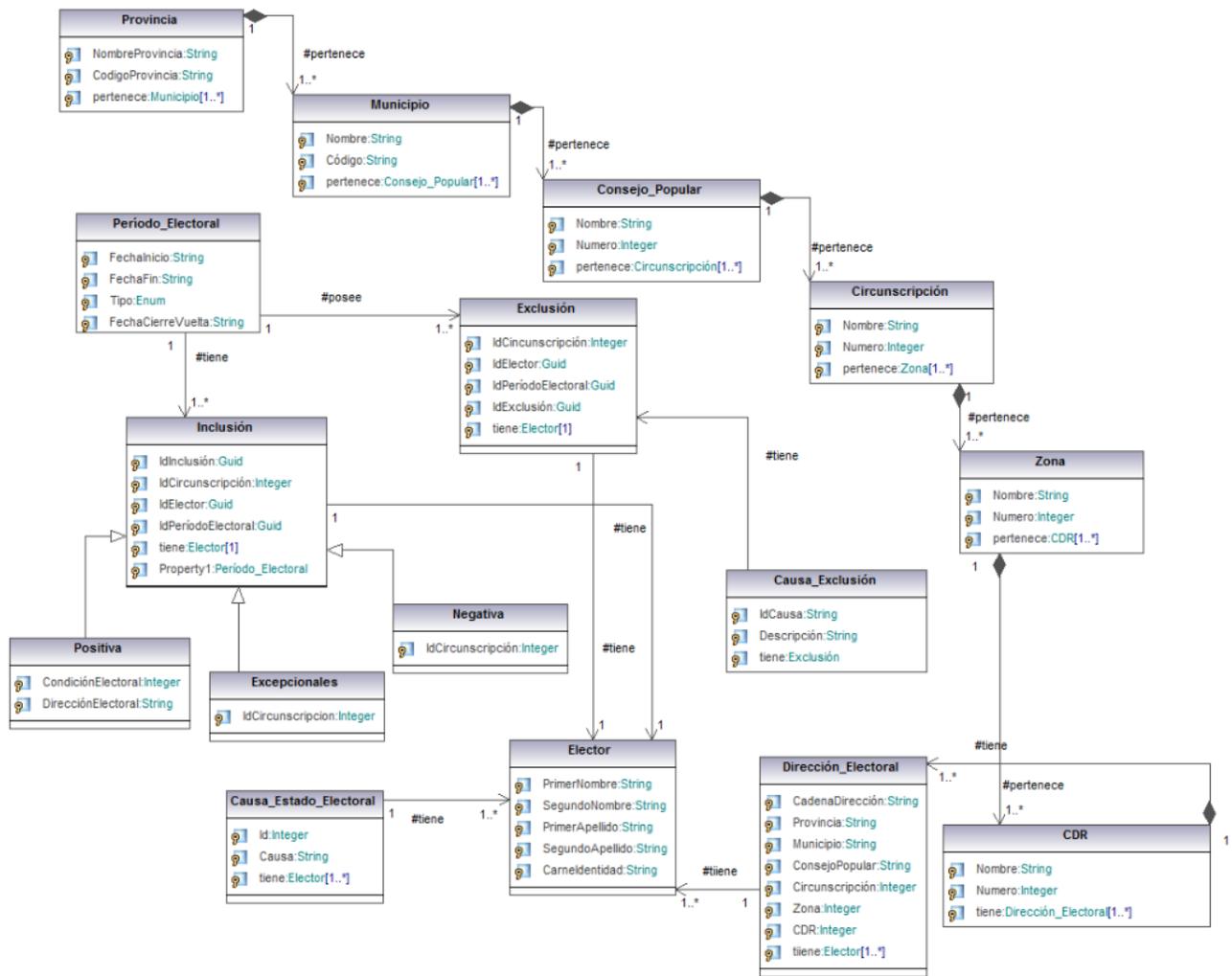


Fig.1 Modelo de Dominio¹³

¹³ Todas las imágenes son de elaboración propia, en caso contrario de especificará la fuente.

En el diagrama anterior se pueden observar las relaciones entre las clases que definen la estructura electoral, de modo que una Provincia está formada por Municipios, que a su vez están divididos en Consejos Populares; los consejos se forman por las Circunscripciones Ordinarias que contienen las Zonas y que a su vez están conformadas por los CDRs, los cuales están formados por direcciones electorales. Cada Elector posee una única dirección electoral y es la única entidad que puede ser incluida o excluida de las listas electorales.

Dentro de un período electoral solo se definen las inclusiones y las exclusiones. Las inclusiones pueden ser tanto positivas, negativas, como excepcionales. Se definen como positivas a toda persona que dentro de las listas electorales no posea ubicación o condición electoral, o ambas; las negativas se referencian cuando una persona se define como inclusión, pero realmente no es de tal forma ya que esta persona sí existía anteriormente en otra lista electoral, contando con ubicación y condición electoral y las excepcionales se presentan cuando el elector emite su voto por un colegio al cual no pertenece debido a problemas que se le presentaron que no pudo emitir éste en su propio colegio.

Vista global del proceso

A continuación se muestra la vista global del proceso, el cual se divide en subprocesos para confeccionar cada uno de sus procesos mejorados. (Ver Fig.2 Diagrama de Macroprocesos).

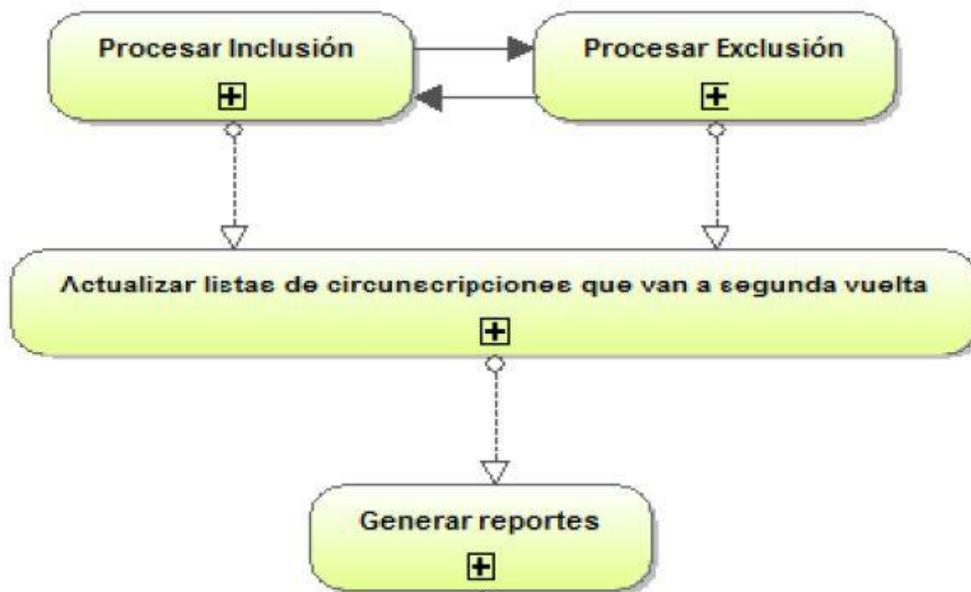


Fig.2 Diagrama de Macroprocesos

Concepción de procesos mejorados.

Una vez comprendidos los procesos del negocio, es momento de proponer las mejoras a dichos procesos y definir las funcionalidades que se comprenderán en la solución propuesta. Como ejemplo se definirán las secuencias de los procesos mejorados Procesar inclusión y Procesar exclusión, donde el primero ilustra la inserción de un ciudadano en la lista electoral a la que corresponde, y el segundo la exclusión de su lista electoral. Para un mejor entendimiento de esta funcionalidad se presenta el siguiente diagrama de proceso mejorado. (Ver Fig.3 Diagrama de proceso mejorado Procesar Inclusión), (Ver Fig.4 Diagrama de proceso mejorado Procesar Exclusión)

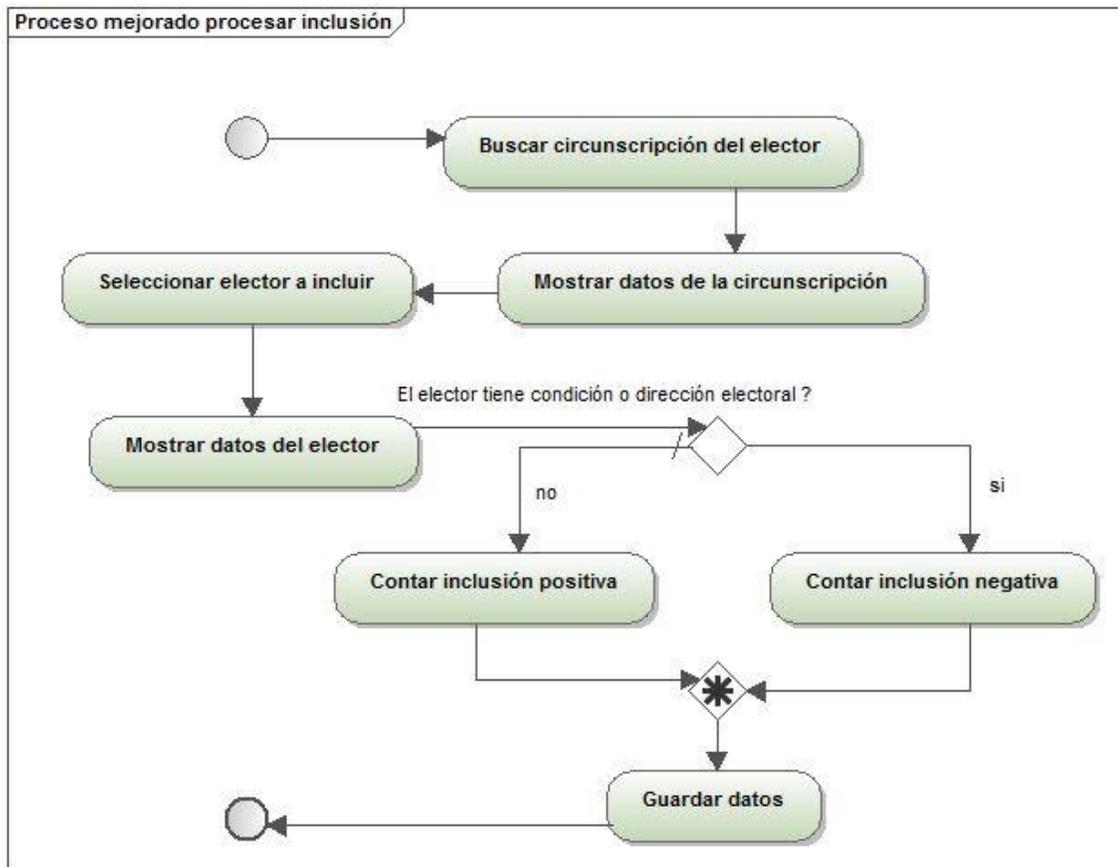


Fig.3 Diagrama de proceso mejorado Procesar Inclusión

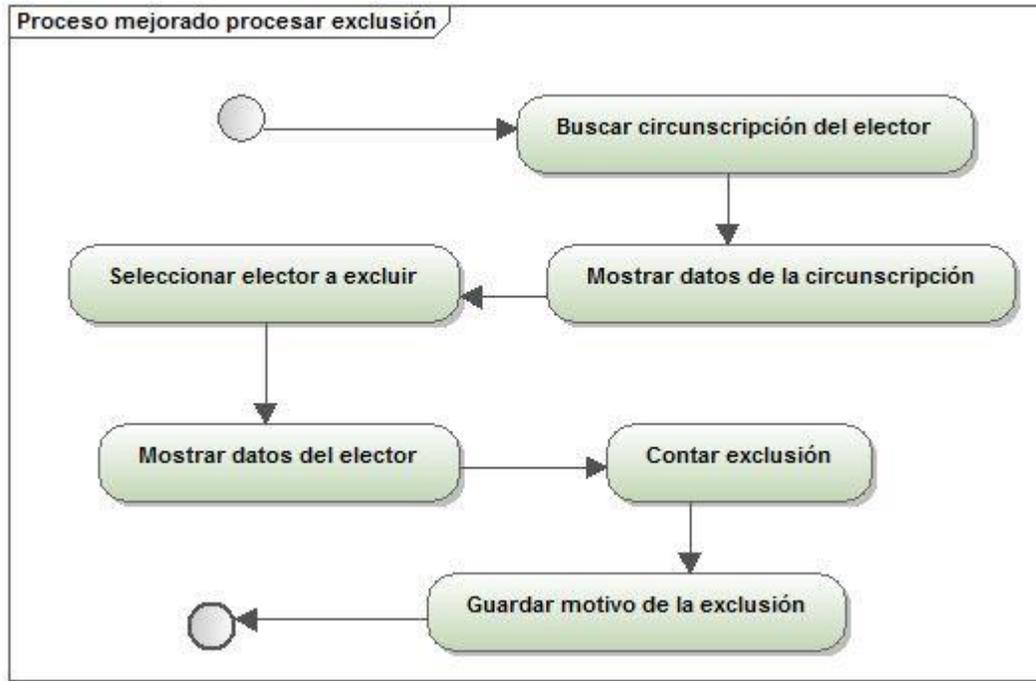


Fig.4 Diagrama de proceso mejorado Procesar Exclusión

Requerimientos funcionales.

Los requerimientos funcionales de un sistema describen la funcionalidad o los servicios que se espera que este provea. Son entendidos como capacidades que debe exhibir una aplicación con el fin de resolver un problema. (21)

La especificación de los requisitos es uno de los artefactos fundamentales a realizar que propone la metodología de desarrollo utilizada. Para ejemplificar la especificación de requisitos funcionales del proceso de reprocesamiento del Sistema Único de Identificación Nacional de la República de Cuba, a continuación se describe el RF1. Procesar inclusión. (Ver Tabla 1)

RF1. Procesar inclusión

Propósito	Permite censar una inclusión
Roles	Administrador municipal
Precondiciones	El administrador municipal debe haberse autenticado.

Entidades tratadas	Entidad	Atributos
	dpersona	nombre apellido dirección Electoral condición Electoral dirección Permanente
Descripción	<p>1.1 Buscar circunscripción del elector:</p> <ul style="list-style-type: none"> a) Provincia. b) Municipio. c) Circunscripción. <p>1.2 Mostrar datos de la circunscripción:</p> <ul style="list-style-type: none"> a) Nombre circunscripción b) Número circunscripción c) Consejo popular. d) Provincia. e) Municipio. f) Listado de electores <ul style="list-style-type: none"> a) <i>Foto.</i> b) <i>Nombres.</i> c) <i>Apellidos.</i> d) <i>Carné de Identidad.</i> e) <i>Dirección electoral.</i> <ul style="list-style-type: none"> • <i>Provincia.</i> • <i>Municipio.</i> • <i>Consejo Popular.</i> • <i>Circunscripción.</i> • <i>Dirección permanente del candidato.</i> <p>1.3 Seleccionar el elector a incluir</p> <p>1.4 Mostrar datos del elector:</p> <ul style="list-style-type: none"> a) <i>foto.</i> b) <i>Nombre.</i> c) <i>Apellido.</i> d) <i>Carné de Identidad.</i> e) <i>Dirección electoral.</i> <ul style="list-style-type: none"> • <i>Provincia.</i> • <i>Municipio.</i> • <i>Consejo Popular.</i> • <i>Circunscripción.</i> • <i>Dirección permanente del candidato.</i> <p>1.5 Seleccionar la inclusión en “positiva” o “negativa”</p> <p>1.5.1 Seleccionado “positiva”.</p>	

	<p>1.5.1.1 Contar inclusión positiva. 1.5.1.2 Guardo la dirección y condición electoral.</p> <p>1.5.2 Seleccionado “negativa”. 1.5.2.1 Contar inclusión negativa. 1.5.2.2 Mantengo dirección y condición electoral.</p>
Validaciones	-
Postcondiciones	Quedan actualizadas las listas electorales
Prototipo	<p align="center">Prototipo Interfaz: Procesar inclusión</p>

Tabla 1. Especificación del requisito Procesar inclusión

RF5. Actualizar electores en CDR

Propósito	Actualizar el Registro de Electores.	
Roles	Administrador provincial	
Precondiciones	El administrador provincial debe haberse autenticado.	
Entidades tratadas	Entidad	Atributos
	dpersona	nombre apellido dirección Electoral condición Electoral dirección Permanente

	dCDR	idcdr idzona idconsejopopular idcircunscripcion idmunicipio idprovincia
Descripción		<p>4.1 Buscar CDR</p> <p>4.2 Mostrar datos del CDR</p> <ul style="list-style-type: none"> a) Provincia b) Municipio c) Consejo popular d) Circunscripción e) Zona f) CDR <p>4.3 Mostrar electores del CDR</p> <p>4.4 Se muestran las opciones:</p> <ul style="list-style-type: none"> a) Nuevo b) Mover c) Eliminar d) Cancelar <p>4.4.1 Si selecciona la opción <i>Nuevo</i>:</p> <ul style="list-style-type: none"> 4.4.1.1 Se busca el elector 4.4.1.2 Se muestran los datos del elector <ul style="list-style-type: none"> a) Nombre b) Apellidos c) Dirección Electoral d) Dirección Permanente e) Condición Electoral 4.4.1.3 Se muestra la dirección electoral del CDR 4.4.1.4 Asigno la dirección electoral del CDR al elector 4.4.1.5 Guardo los cambios realizados 4.4.1.6 Actualizar la lista del CDR

	<p>4.4.2 Si seleccionar la opción <i>Mover</i>:</p> <p>4.4.2.1 Seleccionar al elector</p> <p>4.4.2.2 Mostrar datos del elector</p> <ul style="list-style-type: none">a) Nombreb) Apellidosc) Dirección Electorald) Dirección Permanentee) Condición Electoral <p>4.4.2.3 Se muestran las opciones para <i>Mover</i>:</p> <ul style="list-style-type: none">a) por electorb) por dirección <p>4.4.2.3.1 Si selecciona la opción <i>por elector</i>:</p> <ul style="list-style-type: none">4.4.2.3.1.1 Se busca la referencia del elector hacia donde se movió4.4.2.3.1.2 Se le asigna la dirección electoral de este nuevo elector al otro4.4.2.3.1.3 Guardo los cambios realizados4.4.2.3.1.4 Actualizar las listas de electores <p>4.4.2.3.2 Si selecciona la opción <i>por dirección</i>:</p> <ul style="list-style-type: none">4.4.2.3.2.1 Se busca dicha dirección devolviendo la dirección electoral a la que pertenece4.4.2.3.2.2 Se asigna dicha dirección al elector4.4.2.3.2.3 Guardo los cambios realizados4.4.2.3.2.4 Actualizar las listas de electores <p>4.4.3 Si selecciona la opción <i>Eliminar</i>:</p> <p>4.4.3.1 Selecciono al elector</p> <p>4.4.3.2 Mostrar datos del elector</p> <ul style="list-style-type: none">a) Nombreb) Apellidosc) Dirección Electorald) Dirección Permanentee) Condición Electoral <p>4.4.3.3 Cambiar la condición electoral</p> <p>4.4.3.4 Guardo los cambios realizados</p> <p>4.4.3.5 Actualizar las listas de electores</p>
--	--

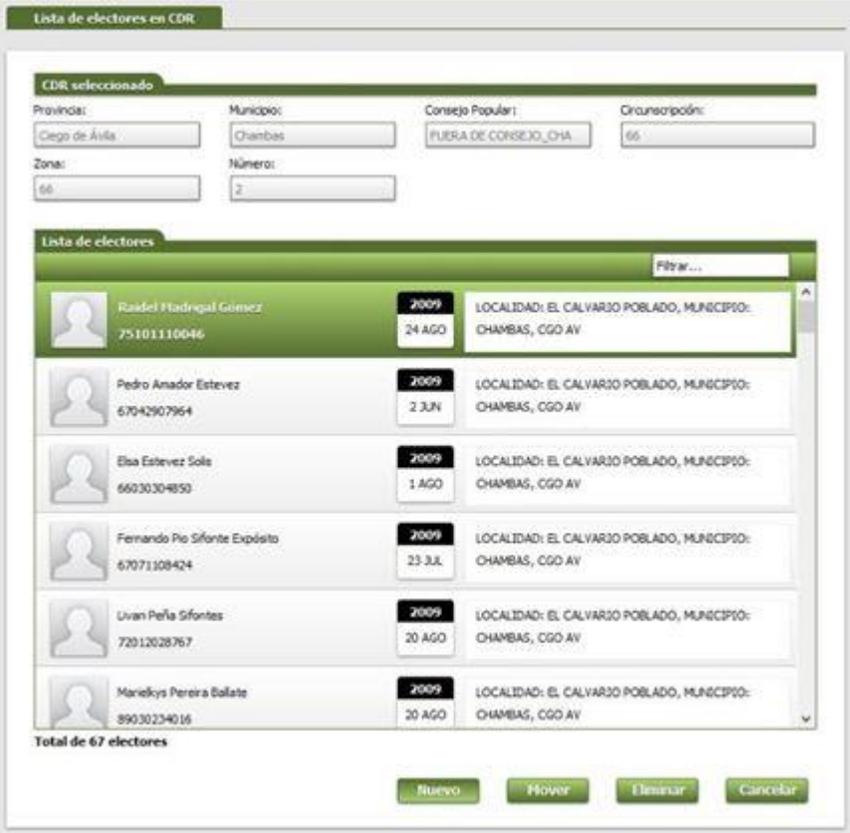
	<p>4.4.4 Si selecciona la opción <i>cancelar</i>:</p> <p>4.4.4.1 Cancelar proceso</p> <p>4.4.4.2 Redireccionar a pantalla anterior</p>
Validaciones	-
Postcondiciones	Se actualizan las listas electorales de las circunscripciones que van a segunda vuelta de las elecciones
Prototipo	 <p>Prototipo Interfaz: Actualizar electores en CDR</p>

Tabla 2. Especificación del requisito Actualizar electores en CDR

Requisitos no funcionales.

Según la IEEE los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable,

rápido y confiable. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto (22). A continuación se muestran los requerimientos no funcionales definidos para el proceso de reprocesamiento del Sistema Único de Identificación Nacional de la República de Cuba.

RNF1. Usabilidad

El sistema podrá ser usado por personas con conocimientos básicos de informática.

RNF2. Disponibilidad

El sistema estará en funcionamiento en los horarios establecidos y podrá ser usado por cualquier usuario según el rol que desempeñe.

RNF3. Interfaz de usuario

El sistema contará con una interfaz accesible e intuitiva. Además de poseer consistencia de la aplicación entre los distintos navegadores.

RNF4. Seguridad

El sistema registrará todas las acciones y contará con diferentes niveles de acceso: de los usuarios, de las oficinas y de las estaciones de trabajo.

RNF5. Interfaz de Hardware

El sistema contará con una interfaz capacitada para conectarse a los dispositivos de impresión.

RNF6. Interfaz de Software

Se deben exponer servicios web que permitan la comunicación con los distintos sistemas que necesiten notificar u obtener información del sistema.

Patrones de diseño

Los patrones de diseño son un conjunto de prácticas de óptimo diseño que se utilizan para abordar problemas recurrentes en la programación orientada a objetos, son el esqueleto de las soluciones a problemas comunes en el desarrollo de software brindando una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. En otras palabras son un

mecanismo popular para describir soluciones generales de problemas de diseño que pueden ser reusadas en la construcción de aplicaciones. Cada patrón prescribe una estructura de clases, sus roles y colaboraciones, y una adecuada asignación de métodos para resolver un problema de diseño en una manera flexible y adaptable. Con el objetivo de una mayor claridad y fortaleza del código, se utilizaron los siguientes patrones de diseño:

- **Encapsulación:** propone esconder algunos componentes, permitiendo sólo accesos estilizados al objeto. Se hace uso de este patrón en casi todas las clases que componen al sistema, permitiendo que estas solo posean como elementos públicos aquellos que son exclusivamente necesarios. (Ver Fig.5 Uso del patrón Encapsulamiento)

```
public class ReprocessingController
{
    private static IReprocessingServiceConnector Connector[...]
    private static IReprocessingServiceConnector _repConnector;
    public string GetData(Número value)[...]
    public string Nombre(string nombre)[...]
```

Fig.5 Uso del patrón Encapsulamiento

- **Subclase:** propone heredar miembros por defecto de una superclase, seleccionando la implementación correcta a través de resoluciones sobre qué implementación debe ser ejecutada. Se puede encontrar este patrón con más fuerza en las entidades de negocio, que por su conceptualización, las funciones y la información que almacenan pueden estar diferenciadas en cierta medida. (Ver Fig.6 Uso del patrón Subclase)

```
[Serializable]
public class OrdinaryCircuncription: Circuncrip
```

Fig.6 Uso del patrón Subclase

- **Excepciones:** propone introducir estructuras de lenguaje para arrojar e interceptar excepciones. Se identificaron los diferentes tipos de errores a tratar dentro del sistema, creando clases que permitan identificar cada tipo de error en el momento de ejecución. (Ver Fig.7 Uso del patrón Excepciones)

```
catch (Exception)
{
    messageMost.Controls.Clear();
    messageMost.Controls.Add(SuinMessages.MensajeErrorSuin((string.Format("Existen errores en la comunicación con el servidor"))));
}
```

Fig.7 Uso del patrón Excepciones

- **Fábrica:** provee de una interfaz para crear familias de objetos relacionados o dependientes sin especificar los tipos concretos de clases. Su uso se encuentra centrado a la creación de los conectores correspondientes al acceso a datos que se esté utilizando, así como en la obtención de los servicios a utilizar. (Ver Fig.8 Uso del patrón Fábrica)

```
Circuncrip circun = new OrdinaryCircuncription
{
    description = cir.DESCRIPCION,
    idcircuncrip = int.Parse(cir.IDCIRCUNSCRIPCION.ToString()),
    IdCircuncription = int.Parse(cir.IDCIRCUNSCRIPCION.ToString()),
    idmunicipality = int.Parse(cir.IDMUNICIPIO),
    idpopularity = int.Parse(cir.IDCONSEJOPOPULAR.ToString()),
    idprovince = int.Parse(cir.IDPROVINCIA),|
};
```

Fig.8 Uso del patrón Fábrica

- **Singleton:** se asegura que solo se pueda crear una instancia de la clase y ofrece un punto global de acceso a esa instancia. El uso de este patrón permite que los servicios puedan ser creados solo una vez. (Ver Fig.9 Uso del patrón Singleton)

```
public List<ComisionData> GetCircuncriptionRegister(string provincia, string municipio)
{
    try
    {
        var referencia = new ReprocessingServiceClient();
        var respuesta = referencia.GetCircuncriptionRegister(provincia, municipio).ToList();
        referencia.Close();
        respuesta = (from proc in respuesta
                    orderby int.Parse(proc.IdCircunscripcion.ToString())
                    select proc).ToList();
        return respuesta;
    }
}
```

Fig.9 Uso del patrón Singleton

Patrones de workflow

Con el propósito de lograr un mejor entendimiento y una mayor claridad en el esquema para el diseño de los workflow, se emplearon diferentes patrones, los cuales se definen a continuación:

Patrones de control de flujo básico: estos patrones están presentes en la mayoría de los lenguajes de workflow y sirven para modelar procesos secuenciales, paralelos o aquellos que incluyan alguna decisión.

- **Sequence:** permite modelar los pasos consecutivos en un workflow, además basa su aplicación en unir dos actividades con una flecha de control de flujo incondicional. (Ver Fig.10 Uso del patrón de flujo básico Sequence)



Fig.10 Uso del patrón de flujo básico “Sequence”

- **Exclusive Choice:** permite decidir el camino a seguir en el workflow sobre la base de una decisión o de los datos de control del workflow. (Ver Fig.11 Uso del patrón de flujo básico Exclusive Choice)

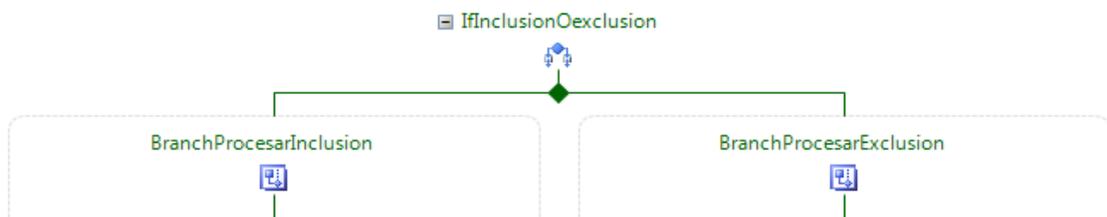


Fig.11 Uso del patrón de flujo básico “Exclusive Choice”

Patrones estructurales: permiten terminar un subproceso cuando ya no haya nada que hacer o permiten definir ciclos de forma arbitraria.

- **Arbitrary Cycles:** permite realizar una o más actividades en el workflow repetidas veces. (Ver Fig. 12 Uso del patrón estructural Arbitrary Cycles)

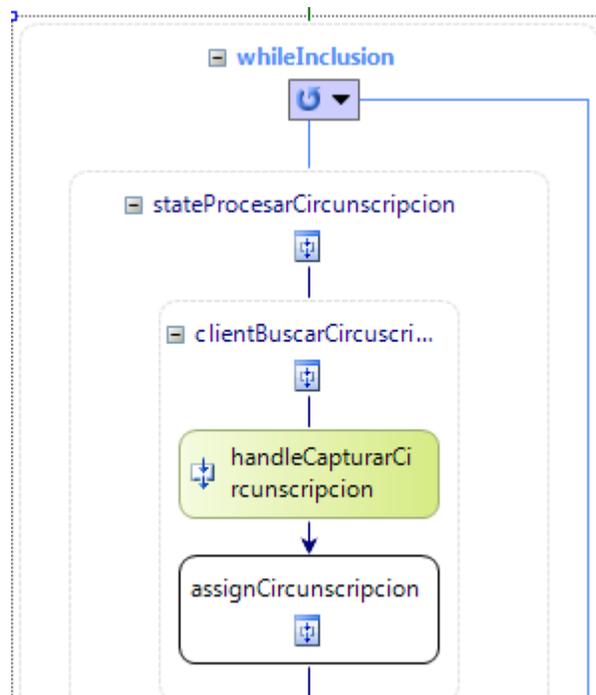


Fig.12 Uso del patrón estructural “Arbitrary Cycles”

Arquitectura de la solución

La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del software. (23)

El modelo de la arquitectura planteado se compone de cinco capas lógicas que lo dotan de un alto nivel de encapsulamiento, permitiendo de esta manera reducir al máximo el acoplamiento y aumentar la reutilización de las mismas. Esta distribución permite que al realizar grandes cambios en una de las capas no se

necesiten modificar las demás. La comunicación entre las capas se realizará a través de interfaces que permiten trabajar de manera transparente a las instancias reales. Además, es una arquitectura cliente-servidor orientada a servicios y procesos. (Ver Fig.13 Capas de la arquitectura)

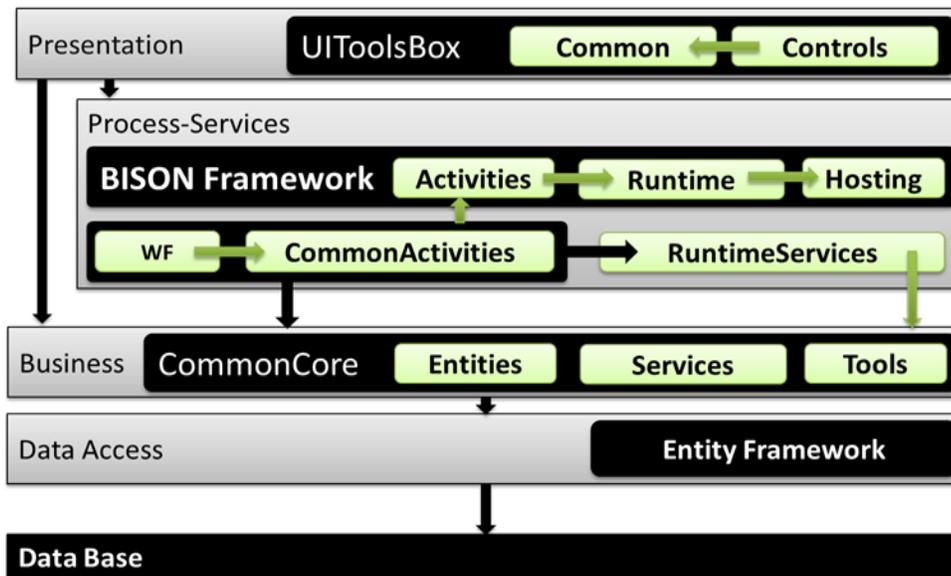


Fig.13 Capas de la arquitectura. Fuente: Documento de la arquitectura del SUIN

Descripción de la arquitectura

Presentación

Conformada por todas las interfaces de usuario y los componentes necesarios para su correcto funcionamiento. Estos elementos pueden ser ficheros Java Script, CSS, HTML, servicios web, interfaces de usuario, etc.

Se encuentra representada por el proyecto Web de la aplicación y tiene interacción directa con la capas de Procesos y Servicios y con la de Negocio.

Procesos - Servicios

Tiene contenida la lógica de los procesos de negocio representados por *workflows*, las actividades que por su nivel de reutilización o importancia lógica se encapsulan en una actividad propia del proyecto y los servicios de tiempo real, los cuales son los encargados de la interacción con los *workflows* definidos. Estos a su vez interactúan con los servicios de la capa de negocio haciendo uso para ello de una fábrica de

servicios que le da la instancia correcta. Al mismo nivel se encuentra definida una fábrica para los servicios de tiempo real que le permite a la capa de presentación interactuar con estos sin necesidad de que se sepa la instancia concreta del servicio que utilizan. Todos estos elementos se encuentran vinculados directamente con el *framework Bison*.¹⁴

Negocio

En esta capa se recogen todos los servicios necesarios para darle solución a los requerimientos de negocio que no pueden ser satisfechos por el *workflow*. Los servicios se encuentran definidos según el contexto en el que se desenvuelven y tienen la responsabilidad de manejar todas las operaciones sobre una entidad de negocio en específico así como todas las entidades que por conceptos de composición se encuentran relacionadas con esta. Por cada entidad de negocio se crea un controlador y una interfaz que debe ser implementada por el acceso a dato que le dará soporte.

Acceso a Datos

La capa de acceso a datos está directamente relacionada con los servicios definidos en el negocio. Para establecer esta relación hace uso de la interfaces de conectores y de la fábrica de conectores que define la capa de negocio. De esta manera es posible realizar cambios en esta capa sin que se vean afectadas las demás capas. Su principal función es realizar una implementación de las interfaces definidas en la capa de negocio y al mismo tiempo trabajar directamente con la fuentes de datos establecida.

Base de Datos

Por la importancia que posee el desarrollo de procedimientos y vistas dentro del negocio, se ha decidido separar la base de datos en una nueva capa donde se concentra una pequeña parte de la lógica de las funcionalidades dentro de la aplicación. La única capa encargada de interactuar con esta es la de Acceso a Datos.

Especificación de clases

Los datos y las operaciones comunes a un conjunto de objetos forman un conjunto que se denomina clase. Para la implementación de la solución propuesta se definieron diferentes tipos de clase, entre ellas se

¹⁴ Es un *framework* para la orquestación de procesos de negocio con Windows Workflow Foundation. Su principal objetivo es proporcionar un componente que permita gestionar las instancias de *workflow*.

encuentran las clases conectoras y las clases gestoras para el acceso directo a la base de datos y las clases entidades para la persistencia de la información.

Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y las interfaces en una aplicación, además de mostrar el flujo de informaciones en el sistema. (Ver Fig.14 Diagrama de clases del diseño).

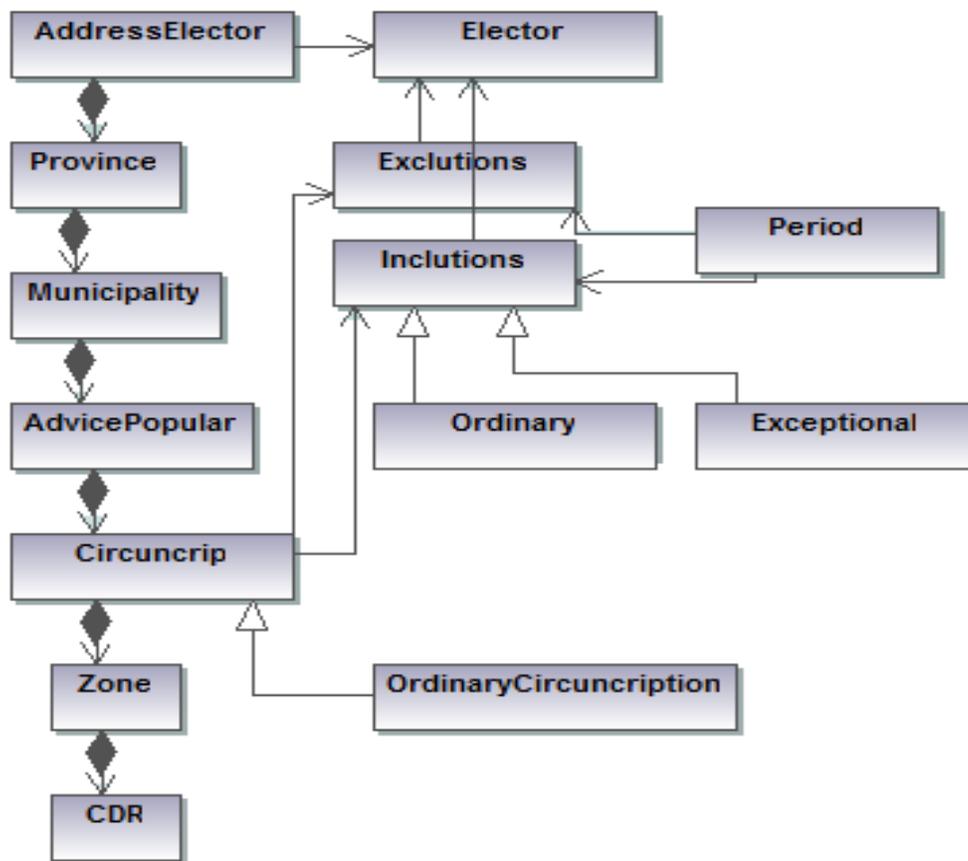


Fig.14 Diagrama de clases del diseño

Clases conectoras

Estas clases tiene la responsabilidad de manejar las entidades del negocio que son necesarias persistir en una base de datos. Permiten establecer una conexión entre las entidades del negocio y la base datos para proporcionar la total libertad del gestor de base de datos ante cualquier cambio de alguna de sus clases. (Ver Tabla 3. Descripción de la clase conectora *ReprocessingServiceConnector*)

Nombre	ReprocessingServiceConnector
Descripción	Interfaz que permite la conexión del acceso a datos con el negocio.
Métodos	Descripción
GetDataComisionByldMunicipality	Permite obtener los datos de la comisión electoral por cada municipio.
InsertDataComisionByConstituency	Permite verificar que coincidan los datos de la provincia, el municipio y la circunscripción.
DeleteDataComisionByldData	Permite eliminar los datos de una comisión electoral en específico.
InsertInclusionPositive	Permite añadir una inclusión positiva.
InsertNegativeInclusion	Permite añadir una inclusión negativa.
InsertExclusion	Permite añadir una exclusión.
GetElectorThatWasProcessing	Permite obtener los datos de un elector que ya fue procesado.
GetCircunscriptionWithoutRegister	Permite obtener todas las circunscripciones sin registrar.
GetCircunscriptionRegister	Permite obtener todas las circunscripciones registradas.
GetCategoryOfExclusion	Permite obtener la categoría de la exclusión.
GetCauseOfExclusion	Permite obtener la causa de la exclusión.

Tabla 3. Descripción de la clase conectora *ReprocessingServiceConnector*

Clases gestoras

Las clases gestoras emplean a las clases conectoras haciendo uso de los datos que devuelven y conformando las clases del negocio, lo cual facilita el mantenimiento de las aplicaciones disminuyendo el esfuerzo de actualización en caso de ocurrir cambios en la fuente de datos y además permite la realización de las pruebas a la aplicación mediante implementación de conectores de pruebas que no accedan a datos reales. (Ver Tabla 4. Descripción de la clase controladora *VoterRegistrationController*)

Nombre	VoterRegistrationController
---------------	-----------------------------

Descripción	Es la encargada de gestionar las relaciones entre las entidades y los conectores.
Métodos	Descripción
LoadAddressElectorsByCdr	Permite obtener la dirección electoral por CDR.
LoadAddressElectorsForPerson	Permite obtener la dirección electoral de una persona.
UpdateAddressElectors	Permite actualizar las direcciones electorales.
UpdateElectoralEstated	Permite actualizar los estados electorales.
UpdateElectoralPeriod	Permite actualizar los períodos electorales.
UpdateZone	Permite actualizar las zonas.
UpdateCDR	Permite actualizar los CDR.
AddPopularAdvince	Permite añadir un consejo popular.
AddZona	Permite añadir una zona.
MoveElectorsAddressOfCdr	Permite asignarle una nueva dirección electoral a un CDR.

Tabla 4. Descripción de la clase controladora VoterRegistrationController

Clases entidades

Las clases entidades representan los datos que van a persistir en el tiempo y permiten la presencia de los mismos a lo largo del ciclo de vida del workflow. Estas ofrecen soporte a los servicios y manejan actividades específicas. La siguiente tabla describe la clase entidad (*AddressElectors*), una de las clases necesarias en la implementación de la propuesta solución. (Ver tabla 5. Descripción de la clase entidad *AddressElectors*)

Nombre	AddressElectors	
Descripción	Entidad que representa los datos de una dirección electoral	
Atributos	Tipo de Dato	Definición
idaddress	Guid	Identificador único de la dirección.
idprovince	Int	Identificador de la provincia
idmunicipality	Int	Identificador del municipio
ncdr	Int	Número del CDR
nadvicepopular	Int	Número del Consejo Popular
nzone	Int	Número de la Zona

Tabla 5. Descripción de la clase entidad AddressElectors

Servicios del sistema

Un servicio es una función auto-contenida, que acepta llamada(s) y devuelve respuesta(s) mediante una interfaz que es capaz de comunicarse con sistemas externos a él. Los servicios pueden también ejecutar acciones discretas de trabajo como serían editar y procesar una transacción. Estos no dependen del estado de otras funciones o procesos, y los consumidores solo trabajarían en la orquestación de estos, secuenciar los servicios y proveer la lógica adicional para procesar datos.

La solución propuesta posee una arquitectura cliente-servidor orientada a servicios, donde estos conforman una parte esencial en su funcionamiento. En el proceso de reprocesamiento los servicios han sido desglosados, según las funciones específicas en las que trabaje, en dos grupos: Servicios de Runtime (*Runtime Services*) y Servicios de Negocio (*Business Services*).

Servicios de *Runtime*

Los servicios de *Runtime* son los encargados de la interacción con los *workflows* definidos y a su vez interactúan con los servicios de la capa de negocio haciendo uso para ello de una fábrica de servicios que le da la instancia correcta de dicho servicio. (24)

- ***ICaptureMunicipality***: Interfaz que permite la comunicación con el servicio para gestionar los municipios, permitiendo la captura de los datos para adicionarlos o modificarlos.
- ***IntegerService***: Representa un servicio que permite la conversión de los datos a enteros.
- ***ICaptureCircuncription***: Representa la interfaz de comunicación con el servicio que gestiona la adición o modificación de las circunscripciones.
- ***IDataPerson***: Interfaz que comunica el servicio que permite trabajar con los datos de las personas.
- ***IAddressElectoralService***: Representa la interfaz encargada de la comunicación con el servicio que gestiona las direcciones electorales que son asignadas.
- ***ICaptureCDR***: Es la interfaz encargada de capturar los datos de un CDR.

Servicios de Negocio

Los servicios de negocio surgen para darle solución a los requerimientos de negocio que no pueden ser satisfechos por el *workflow* y tienen la responsabilidad de manejar todas las operaciones sobre una entidad de negocio en específico, así como todas las entidades que por conceptos de composición se encuentran relacionadas con esta. (25)

- ***IAddressElectoralRService***: Comprende todas las funcionalidades para el manejo de los procesos electorales permitiendo la gestión de las instancias electorales y de los procesos de actualización de las bases de datos.
- ***IEncoderService***: Interfaz que permite la utilización del servicio para la comunicación y filtrado de los nomencladores de la Base Nacional de Identidad.
- ***ISearchPersonService***: Es la interfaz de comunicación con el servicio que permite la búsqueda de las personas que cumplan con determinadas condiciones en la Base Nacional de Identidad. Se utiliza en la búsqueda de los electores en procesos como el cambio de la dirección electoral.

Diseño de *workflow*

El soporte funcional del sistema propuesto se basa en la tecnología de modelado de procesos basados en flujos de trabajo o bien conocida como *workflow* el cual es definido por la *Workflow Management Coalition (WfMC)*; la cual es una entidad promotora de estándares para los productos de *workflow*, como “la automatización de los procesos de negocio durante el cual documentos, información y tareas son pasados de un participante a otro, incluso al cliente, acorde a un conjunto de reglas procedimentales”. (26)

Con el objetivo de administrar cada una de las entidades con las que interactúa el *workflow* se utilizaron estructuras para manipular la información contenida dentro del proceso las que, entre otras funciones, aportan herramientas similares a las usadas en distintos lenguajes y cumplen tareas equivalentes. Algunas de estas estructuras de control de flujo fueron:

- Condicionales (*IfElseActivity*).
- Bucles While – Until (*WhileActivity*).
- Disparo y captura de excepciones (*ThrowActivity*).
- Manejo de eventos (*HandleExternalEventActivity*).

Las actividades son las encargadas de manipular y procesar la información, así como, la comunicación del *workflow* con el mundo externo al flujo de procesos. Estas son bloques con los que se construyen los flujos de trabajo y pueden ser simples (las creadas por el propio desarrollador al heredar de la clase *Activity*) o compuestas (combinando varias actividades simples). En la solución propuesta se utilizaron las siguientes actividades:

- **IfElseActivity:** Permite ejecutar condicionalmente una de varias condiciones. Coloca una condición en cada rama del bloque *ifElse*. Si la condición se evalúa como verdadera, se ejecutan las actividades contenidas en la actividad *IfElseBranchActivity* quien representa una bifurcación de una actividad *IfElseActivity*.
- **WhileActivity:** Permite que el flujo de trabajo se ejecute en bucle hasta que se cumpla la condición indicada.
- **ClientActivity:** Permite la navegación de forma gráfica para el usuario y tiene como principal función mostrar una interfaz de usuario. Para su uso es necesario especificar la URL¹⁵ del componente representado.
- **CallExternalMethodActivity:** Utilizada para la comunicación de entrada y salida con un servicio local externo al *workflow*. La clase *CallExternalMethodActivity* invoca el método especificado por las propiedades *InterfaceType* y *MethodName* quienes indicarán el lugar donde se aloja el servicio y el método a invocar.
- **CodeActivity:** Es un forma sencilla que permite agregar código de C# (en este caso el lenguaje de programación utilizado aunque también soporta *Visual Basic*) a cualquier parte de un flujo de trabajo.
- **HandleExternalEventActivity:** Bloquea el *workflow* hasta que el evento especificado por la propiedad *InterfaceType* sea lanzado, generalmente se utiliza para la interacción con el usuario y provee la principal fuente de extracción de información con el ámbito exterior al flujo de actividades.
- **ConfigurationActivity:** Habilita el *workflow* para encuestarlo sobre alguna información que contenga o se haya generado en su transcurso.

Transformación del modelo de procesos mejorados al workflow

A continuación se presenta la transformación del proceso mejorado *ReprocesamientoWF* al *workflow*, la cual es una versión más avanzada que la existente, donde la complejidad de todos sus procesos depende del alto nivel en que se encuentren las acciones efectuadas en ese momento y la cantidad de entidades envueltas en ésta. (Ver Fig. 15 Diseño del workflow *ReprocesamientoWF*)

¹⁵ Del inglés *Uniform Resource Locator*. Es una dirección única que especifica la localización de un recurso en Internet.

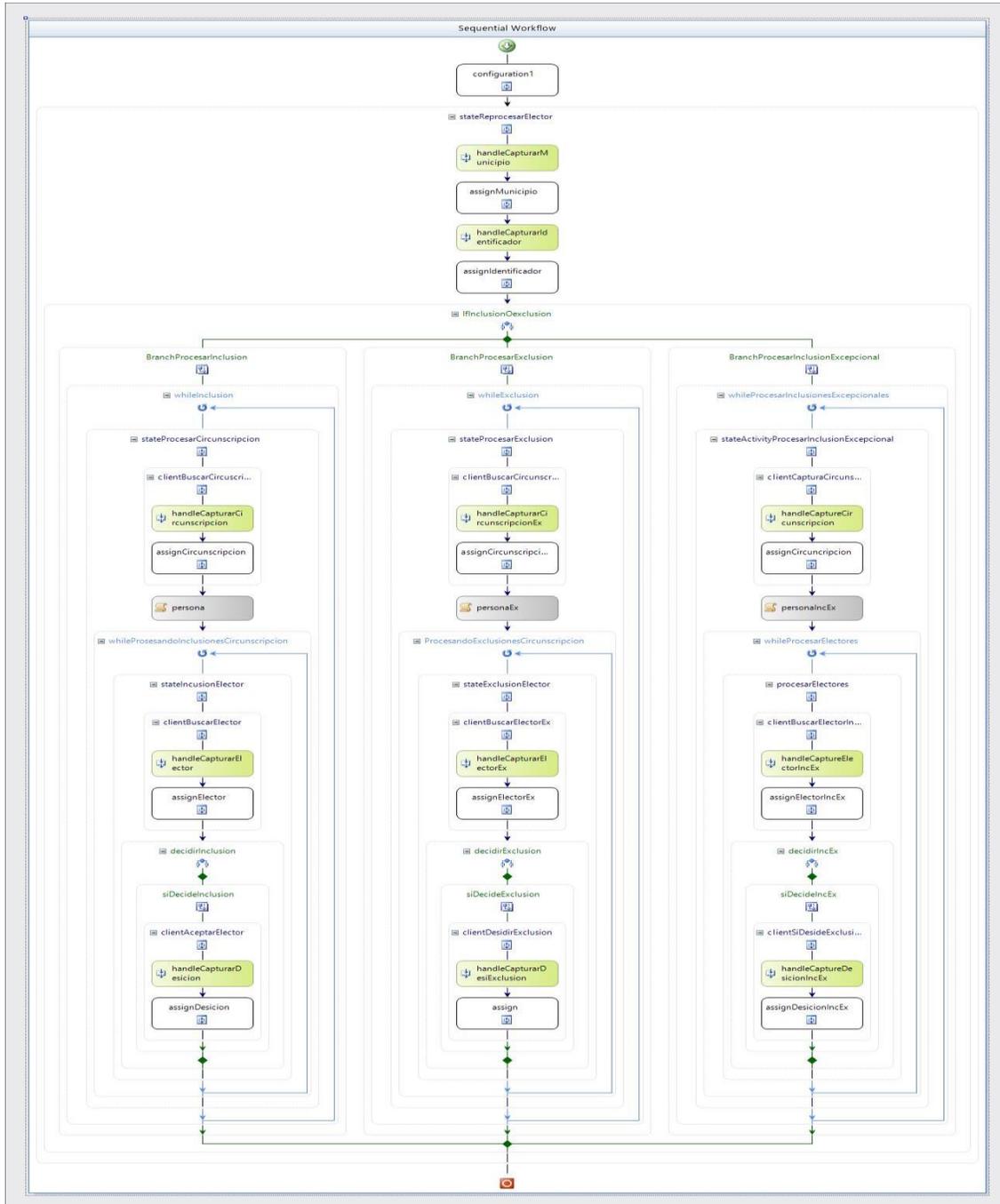


Fig.15 Diseño del workflow ReprocesamientoWF

Modelo de datos

El diseño del modelo de datos es fundamental para obtener una base de datos en la que logren persistir los datos necesarios para cualquier aplicación. La solución propuesta se sustenta fundamentalmente en el trabajo con procesos por lo que se tiene por concepción persistir los datos necesarios e imprescindibles para el negocio. A continuación se muestra el modelo de datos correspondiente al proceso de reprocesamiento donde se seleccionaron las principales entidades persistentes necesarias para la solución propuesta, además de la creación de varios nomencladores con el objetivo de generalizar el tratamiento sobre algunos tipos de datos. (Ver Fig.16 Modelo de datos)

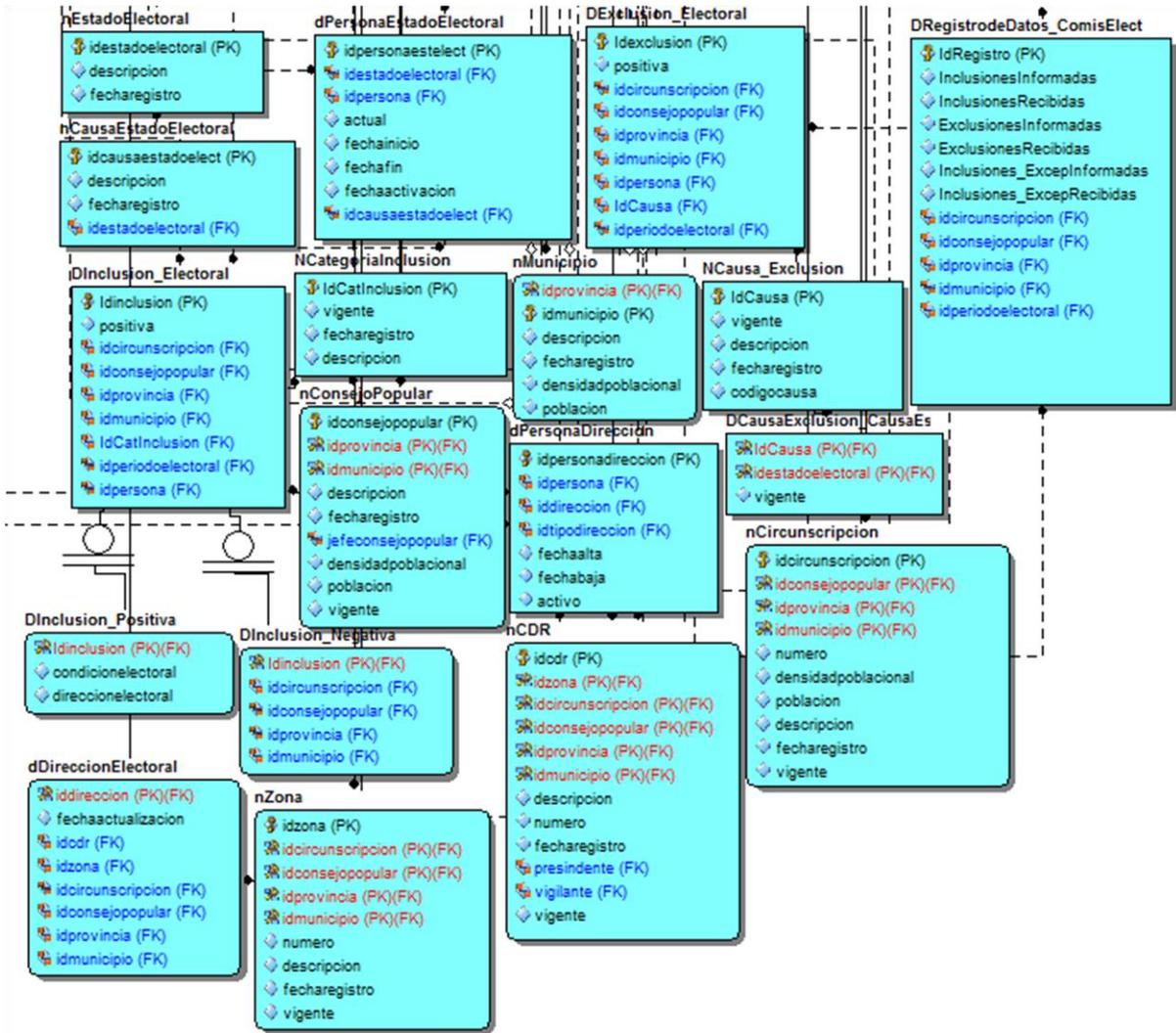


Fig.16 Modelo de datos

Conclusiones Parciales

La confección de este capítulo posibilitó determinar que la arquitectura de software es uno de los elementos más importantes en el desarrollo de un sistema informático debido a que establece la estructura que debe tener el sistema a desarrollar y guía todas las actividades de implementación del mismo. A la misma vez se identificaron las estrechas relaciones existentes entre las diferentes entidades (clases) que componen el diagrama de clases del proceso de reprocesamiento y los diseños de los *workflow* correspondientes, ya que ambos responden a la arquitectura definida donde cada capa hace uso de otras más inmediatas pero a su vez mantiene cierto grado de independencia. Además, permitió el análisis de la similitud que existe entre el diagrama de clases y el modelo de datos debido a que en ambos están presentes las principales entidades que contienen los atributos, métodos y eventos esenciales para la implementación del proceso de reprocesamiento con su posterior integración al Módulo de Registro de Electores en el SUIN.

Capítulo III: Implementación y Prueba

La implementación se inicia a partir de los resultados obtenidos durante la fase de desarrollo. Aquí se define cómo se organizan las clases y objetos en términos de componentes y cómo estos se organizan de acuerdo con los nodos específicos en el modelo de despliegue. Dentro de sus principales objetivos se encuentra implementar los elementos de diseño en términos de elementos de implementación, abriendo el camino a la fase de prueba. Las pruebas constituyen un elemento vital para garantizar la calidad del software y representan una revisión final de las especificaciones, del diseño y la codificación. El objetivo de la etapa de pruebas es garantizar una versión funcional del sistema. Estas constituyen una actividad en la cual un sistema y sus componentes son ejecutados bajo condiciones o requerimientos específicos, los resultados son observados y registrados para su posterior evaluación o corrección.

En el presente capítulo se tratan los temas referentes a la validación del sistema donde se describen los estándares de codificación definidos, el diagrama de despliegue que contiene los componentes del sistema, el diagrama de componentes el cual muestra las relaciones entre los elementos que dan sustento a las funcionalidades del software y los tipos de pruebas al sistema.

Estándares de codificación

Un estándar de codificación es un conjunto de políticas por las que se tiene que regir un programador a la hora de desarrollar un programa, además de indicar cómo deberá estar estructurado el código. Estos estándares se definen por el equipo de desarrollo para lograr la generalización en la programación del software, permitiendo la comunicación uniforme entre los desarrolladores y haciendo más entendible el núcleo de la aplicación. (27)

El desarrollo del proceso de reprocesamiento se rigió por los estándares de codificación definidos por el grupo de desarrollo del proyecto Identidad Cuba.

Estilos de capitalización

Pascal: la primera letra en el identificador y la primera letra de cada subsiguiente palabra concatenada se capitalizan. Se pueden utilizar los identificadores de Pascal case en caso de tres o más caracteres.

Camello: la primera letra en el identificador está en minúscula y la primera letra de cada subsiguiente palabra concatenada es mayúscula.

Mayúscula: Todas las letras en el identificador se capitalizan. Esta convención se utiliza sólo para los identificadores que constan de dos o menos letras.

Reglas sobre el uso de mayúsculas

- No se deberá crear dos *namespaces* que se diferencien solo en el uso de las mayúsculas.
- No crear funciones con nombres de parámetros que se diferencian solo en el uso de la mayúscula.
- No utilizar nombres o identificadores que requieran ser diferenciados solo por el uso de mayúsculas.

Tratamiento de errores

El código implementado se encuentra encerrado dentro de bloques *try-catch*, de forma tal que pueda capturarse cada excepción que sea lanzada por el sistema en tiempo de ejecución, ser tratada de forma independiente en cada una de las funcionalidades implementadas y ser mostradas en las interfaces de usuario mediante componentes *Labels*.

Cada uno de los campos en la captura de datos posee reglas de validación que son proporcionadas por los componentes de interfaz de usuario *UIToolsbox*, evitando que se introduzcan datos erróneos o campos vacíos en los formularios. (Ver Fig.8 Fragmento de interfaz de usuario donde se validan los campos a nivel de cliente.)

Diagrama de componentes

El diagrama de componentes contiene cada uno de los elementos generados a partir de la implementación de las clases definidas en la arquitectura. Los componentes *Suin.dll*, *Suin.Services.dll*, *Suin.Entities.dll*, *Suin.Workflow.dll*; están asociados a cuatro proyectos respectivamente. Donde *Suin* es la solución principal, *Suin.Services* realiza el manejo de los servicios, *Suin.Entities* maneja las entidades y *Suin.Workflow* efectúa todo lo relacionado a los workflows. En la librería *Common.Entities.dll* se encuentran las clases comunes a los diferentes negocios. El componente *CommonServices.VoterRegistrationServices.dll* representa el controlador de los servicios de Windows Communication Foundation (WCF), *CommonServices.VoterRegistrationServices.DAL.dll* es el que se encarga de las conexiones a la base de

datos y *CommonServices.VoterRegistrationServices.Svc.dll* constituye la interfaz de conexión a la base de datos. Por último, el componente ejecutable *DGM.exe* responde por la autorización al sistema de las estaciones de trabajo y los servicios. (Ver Fig.17 Diagrama de componentes)

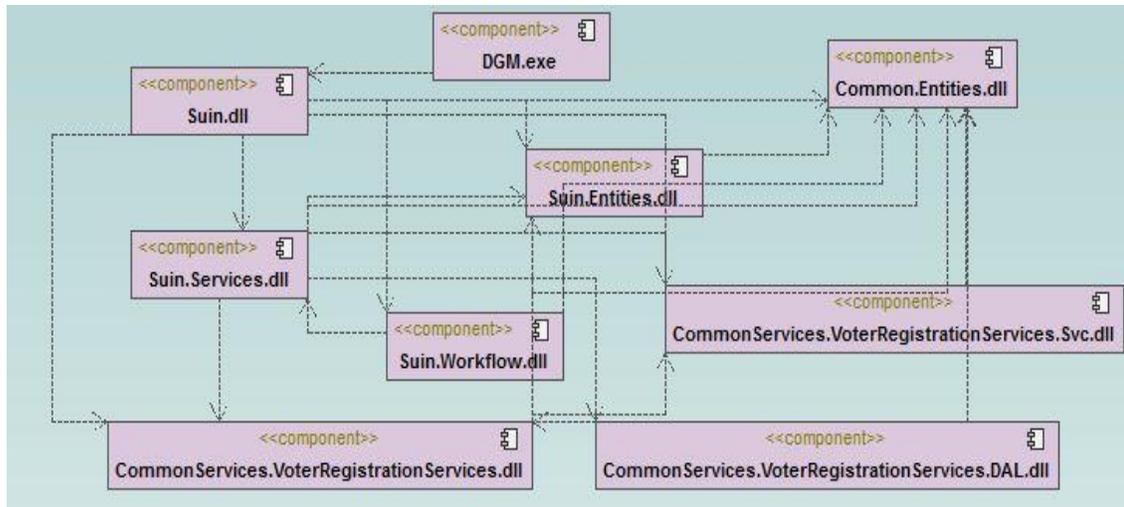


Fig.17 Diagrama de componentes

Diagrama de despliegue

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo¹⁶ puede contener instancias de componentes de software, objetos y procesos.

Los diagramas de despliegue muestran las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes de software (procesos y objetos que se ejecutan en ellos). (28)

El diagrama de despliegue para el proceso de reprocesamiento muestra la representación gráfica de los nodos y dispositivo que lo componen, además de las relaciones entre ellos. Además, cada uno de estos nodos y el dispositivo tiene una función en particular.

Impresora: Dispositivo que permitirá obtener en formato duro los distintos reportes realizados por funcionario del sistema.

¹⁶ Objeto físico en tiempo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento.

PCs Cliente: Representa la(s) PC que será utilizada por el funcionario para interactuar con la aplicación desarrollada. La impresora se conecta mediante un puerto USB¹⁷ (Bus de Serie Universal, por sus siglas en español) a la PCs Cliente, para imprimir información de interés y ésta se comunica por medio del protocolo *HTTP* con el servidor de aplicaciones. Las computadoras clientes deben responder a los requisitos no funcionales que se definieron en un principio.

Servidor de aplicaciones: Un servidor de aplicaciones hace uso extensivo del contenido dinámico y tiene una fuerte integración con las bases de datos, ya que maneja la mayoría de las transacciones relacionadas con la lógica y el acceso a los datos de la aplicación, utilizando para ello el protocolo *TCP/IP*, además de trabajar como intermediario -middleware- para la seguridad y el mantenimiento. (29)

El servidor de aplicaciones es quien permite que la PCs Cliente interactúe y tenga acceso a las diferentes aplicaciones que responden al proceso de reprocesamiento mediante una conexión *HTTP*.

Servidor de base de datos: Permitirá manejar volúmenes de información de una manera estable, fiable, coherente y segura. Además, el servidor de aplicación estará conectado al de base de datos por el protocolo *TNS*¹⁸ Oracle, manteniendo un sistema de réplicas de datos que posibilita la consistencia y la actualización de la información. (Ver Fig. 18 Diagrama de despliegue)

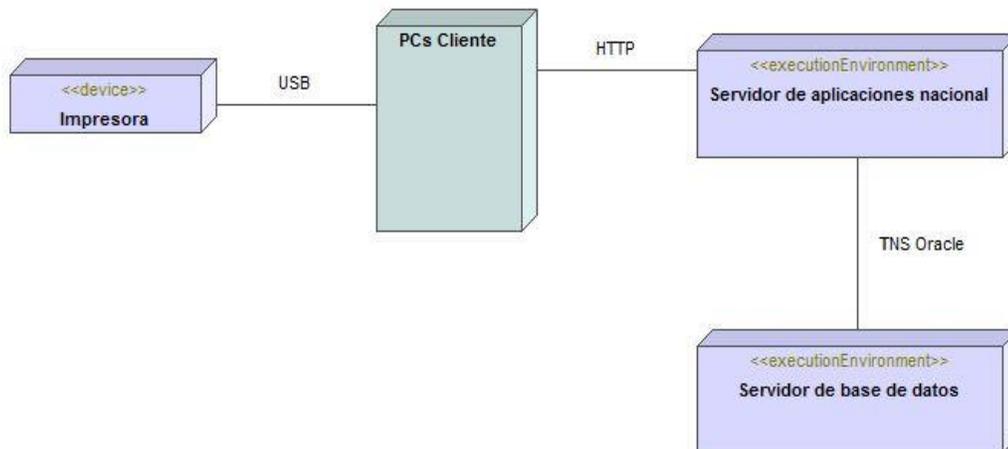


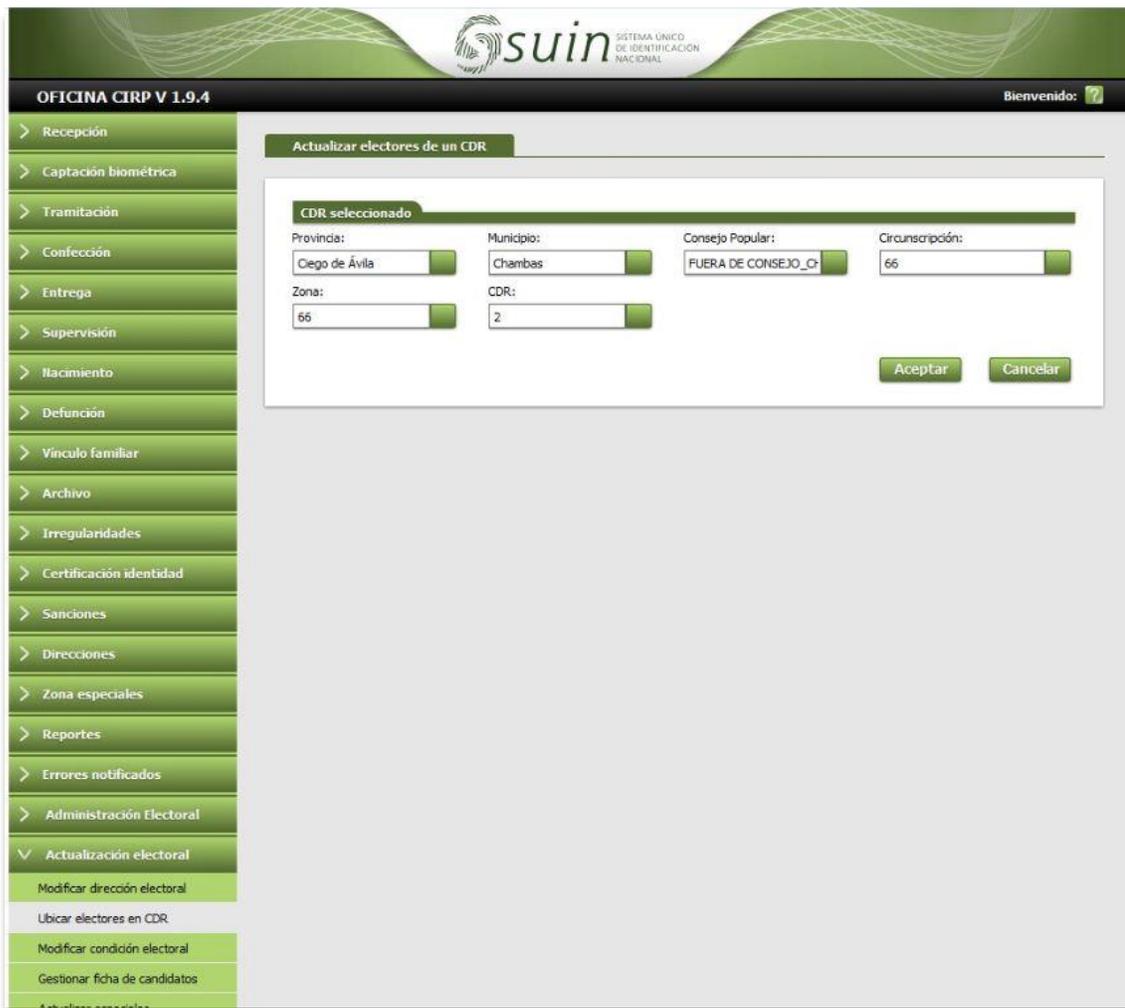
Fig.18 Diagrama de despliegue

¹⁷ Entrada o acceso que permite al usuario compartir información almacenada en diferentes dispositivos.

¹⁸ Transparent Network Substrate (Sustrato de red transparente). Capa de comunicación que utilizan las bases de datos Oracle.

Interfaces del sistema

El diseño de las interfaces de la propuesta de solución se adapta a cada uno de los prototipos de diseño definidos previamente. Las mismas fueron implementadas con el objetivo de proporcionarle al usuario una aplicación que a primera vista sea capaz de revelar un sistema que posee tanta calidad en su codificación como en su apariencia y usabilidad. (Ver Fig.19 Interfaz “Actualizar electores de un CDR”)



OFICINA CIRP V 1.9.4 Bienvenido: ?

Actualizar electores de un CDR

CDR seleccionado

Provincia:	Municipio:	Consejo Popular:	Circunscripción:
<input type="text" value="Ciego de Ávila"/>	<input type="text" value="Chambas"/>	<input type="text" value="FUERA DE CONSEJO_C"/>	<input type="text" value="66"/>
Zona:	CDR:		
<input type="text" value="66"/>	<input type="text" value="2"/>		

Fig.19 Interfaz “Actualizar electores de un CDR”



Fig.19.1 Interfaz “Lista de electores en CDR”



Fig.19.2 Interfaz “Ubicar electores en CDR”

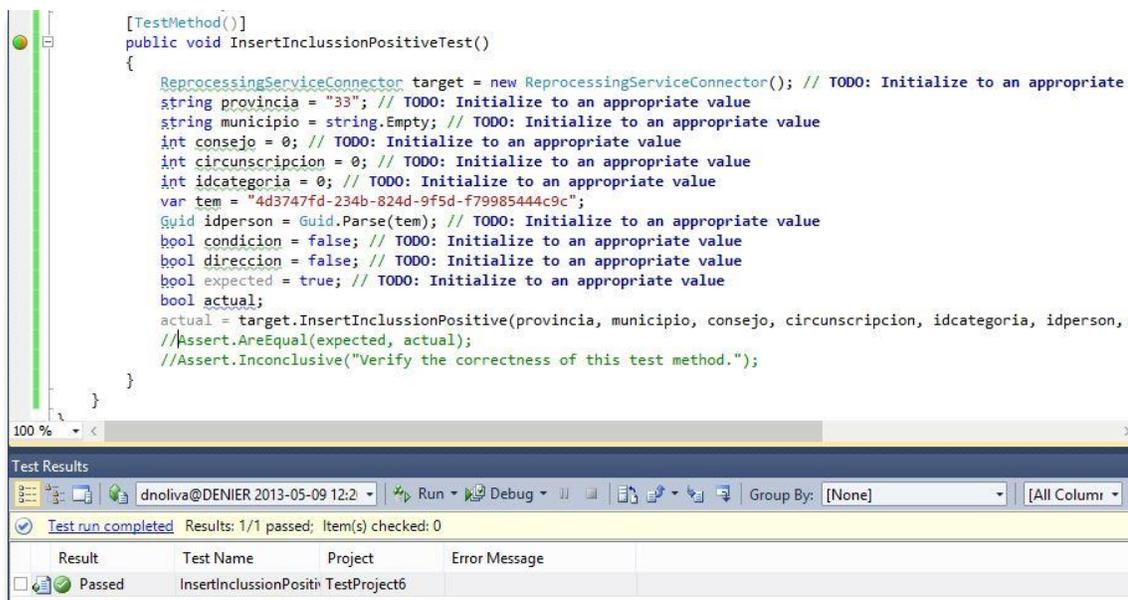
Diseño de casos de prueba

Una vez generado el código fuente, es necesario probar el software para descubrir y corregir la mayor cantidad de errores posibles antes de entregarlo al cliente. El diseño de casos de prueba se fundamenta a partir de una serie de técnicas definidas que persiguen como objetivo esencial validar el software. Para ello fueron seleccionadas como técnicas, las pruebas de caja blanca (pruebas unitarias) usando como unidad los métodos de mayor responsabilidad y peso dentro de la solución y las pruebas de caja negra (pruebas de sistema) a las diferentes funcionalidades del sistema.

Pruebas unitarias

Las pruebas unitarias constituyen pruebas de caja blanca con el objetivo de garantizar que las operaciones internas se realizan de acuerdo a las especificaciones y que se han probado todos los componentes internos de manera adecuada, además de tener por objetivo el aislamiento de partes del código y la demostración de que estas partes no contienen errores.

Con el objetivo de validar el funcionamiento interno de la solución se aplicaron un conjunto de pruebas unitarias a diferentes funcionalidades, como ejemplo se muestra a continuación las pruebas unitarias que se le realizaron a los servicios. (Ver Fig.20 Prueba de caja blanca)



```
[TestMethod()]
public void InsertInclusionPositiveTest()
{
    ReprocessingServiceConnector target = new ReprocessingServiceConnector(); // TODO: Initialize to an appropriate value
    string provincia = "33"; // TODO: Initialize to an appropriate value
    string municipio = string.Empty; // TODO: Initialize to an appropriate value
    int consejo = 0; // TODO: Initialize to an appropriate value
    int circunscripcion = 0; // TODO: Initialize to an appropriate value
    int idcategoria = 0; // TODO: Initialize to an appropriate value
    var tem = "4d3747fd-234b-824d-9f5d-f79985444c9c";
    Guid idperson = Guid.Parse(tem); // TODO: Initialize to an appropriate value
    bool condicion = false; // TODO: Initialize to an appropriate value
    bool direccion = false; // TODO: Initialize to an appropriate value
    bool expected = true; // TODO: Initialize to an appropriate value
    bool actual;
    actual = target.InsertInclusionPositive(provincia, municipio, consejo, circunscripcion, idcategoria, idperson, direccion);
    //Assert.AreEqual(expected, actual);
    //Assert.Inconclusive("Verify the correctness of this test method.");
}
```

Test Results

Test run completed Results: 1/1 passed; Item(s) checked: 0

Result	Test Name	Project	Error Message
Passed	InsertInclusionPositi	TestProject6	

Fig.20 Resultado de la prueba de caja blanca

Pruebas de sistema

Las pruebas de sistema o pruebas de caja negra se concentran en los requisitos funcionales del software además de realizarse cuando la aplicación está funcionando como un todo y están dirigidas a verificar el programa final después que todos los componentes de hardware y software han sido integrados.

Para la validación de cada uno de los requisitos funcionales del sistema se aplicaron diferentes casos de pruebas a dichos requisitos definidos previamente. En la siguiente tabla se muestra el caso de prueba realizado al RF5. Actualizar electores en CDR. (Ver Tabla 6. Caso de prueba Actualizar electores en CDR)

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Buscar CDR	Permite seleccionar el CDR al cual se le actualizará su lista de electores	Muestra los electores del CDR	1.Buscar CDR 2.Aceptar
EC 1.2 Mostrar datos del CDR	Permite visualizar los datos del CDR seleccionado	Se muestran los datos del CDR	1.Buscar CDR 2.Aceptar
EC 1.3 Mostrar electores del CDR	Permite ver la lista de electores del CDR	Se muestra la lista de electores del CDR	1.Buscar CDR 2.Aceptar
EC 1.4 Seleccionar opciones: a)Nuevo b)Mover c)Eliminar d)Cancelar	Permite seleccionar alguna de estas opciones	Se muestran los botones de las distintas opciones	<p>Si selecciona Nuevo:</p> <ul style="list-style-type: none"> • Se busca el elector • Se muestran los datos del elector • Se muestra la dirección electoral del CDR • Asigno la dirección electoral del CDR al elector • Se actualiza la lista de electores de ese CDR <p>Si selecciona Mover:</p> <ul style="list-style-type: none"> • Permite mover por Elector o por Dirección <p>Si selecciona por Elector:</p> <ul style="list-style-type: none"> • Se busca la referencia del elector hacia donde se movió • Se le asigna la dirección electoral de este nuevo elector al otro • Se actualiza la lista de electores de ese CDR <p>Si selecciona por Dirección:</p> <ul style="list-style-type: none"> • Se busca dicha dirección devolviendo la dirección electoral a la que pertenece • Se asigna dicha dirección al elector • Se actualiza la lista de electores de ese CDR <p>Si selecciona Eliminar:</p> <ul style="list-style-type: none"> • Cambiar la condición electoral del elector seleccionado • Se actualiza la lista de electores de ese CDR <p>Si selecciona Cancelar:</p> <ul style="list-style-type: none"> • Redireccionar a pantalla anterior

Tabla 6. Caso de prueba Actualizar electores en CDR.

Resultado de las pruebas

A partir de las pruebas realizadas, las que se encuentran en correspondencia con lo implementado según los requisitos funcionales definidos, se obtuvieron diferentes resultados que indican cómo está funcionando la aplicación. Estos resultados se reflejan en una tabla que contiene la información referente al conjunto de las no conformidades, las cuales se fueron resolviendo gradualmente. A continuación se muestran las no conformidades que fueron detectadas en la Iteración 1. (Ver *Tabla 7. Resumen de las no conformidades de la Iteración 1*)

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección del error	Importancia
RF1	1	Cambiar la posición del botón “Aceptar” hacia la izquierda	Buscar circunscripción del elector	Al mostrar la interfaz	No significativa
RF1	2	Adicionar opción de “Cancelar”	Procesar inclusión	Al mostrar la interfaz	Significativa
RF2	3	El botón “Positiva” debe intercambiar su posición con el botón “Negativa”	Procesar inclusión excepcional	Al mostrar la interfaz	No significativa
RF3	4	El botón “Cancelar” debe comenzar con mayúscula	Procesar exclusión	Al mostrar la interfaz	No significativa
RF5	5	Cambiar el término “Borrar” por el término “Eliminar”	Actualizar electores en CDR	Al mostrar la interfaz “Lista de electores en CDR”	No significativa
RF5	6	Intercambiar la posición de los botones “Mover” y “Nuevo”	Actualizar electores en CDR	Al mostrar la interfaz “Lista de electores en CDR”	No significativa

Tabla 7. Resumen de las no conformidades de la Iteración 1

Los resultados negativos obtenidos durante la primera iteración de desarrollo del proceso se reflejan como no conformidades, es decir, funcionalidades que no cumplen los requisitos preestablecidos y que por ende, son errores que el desarrollador debe solucionar. La evolución de las no conformidades durante el

transcurso de iteraciones para el proceso de reprocesamiento aparece graficada en la **Fig.21**. En la misma se observa cómo en la primera revisión se detectaron 6 no conformidades, número que se fue reduciendo hasta la cuarta revisión donde fueron mitigadas todas estas no conformidades. (Ver Fig.21 Gráfica del comportamiento de las no conformidades según la iteración.)

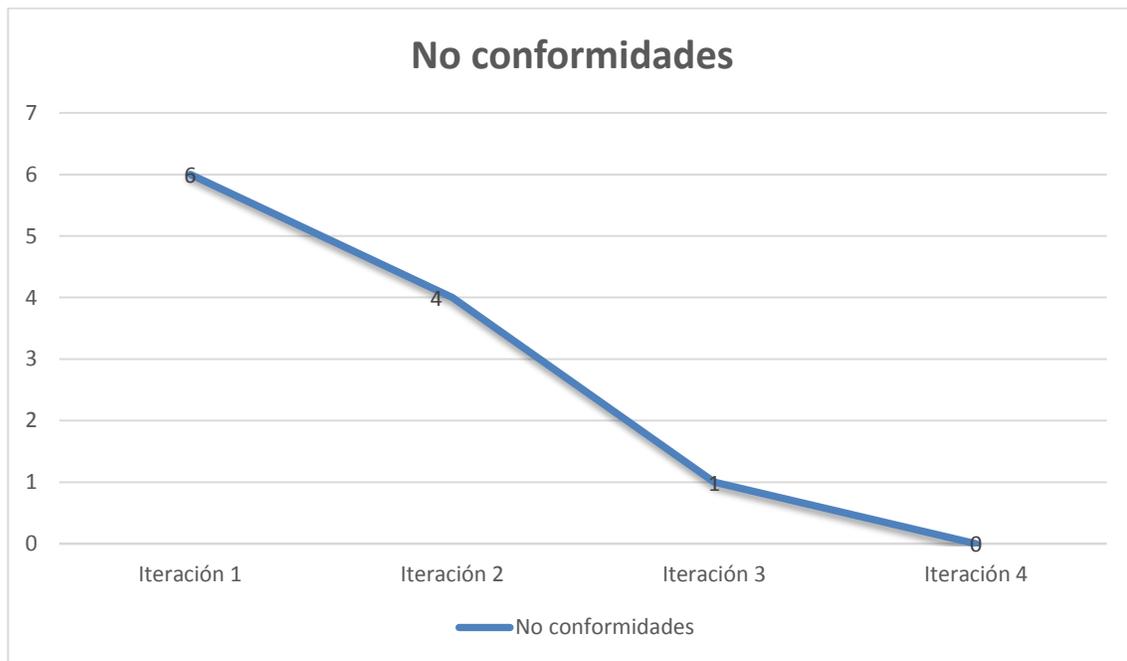


Fig.21 Gráfica del comportamiento de las no conformidades según la iteración

La propuesta de solución en estos momentos se encuentra inmersa en un período de pruebas, por lo que podrían surgir nuevas no conformidades, las cuales serán resueltas en próximas versiones.

Beneficios del proceso

El proceso desarrollado reportará diferentes beneficios para las oficinas de la Dirección de Identificación y Registro (DIR). Entre ellos se pueden encontrar que permite una mayor eficiencia, en cuanto a tiempo de respuesta, del proceso de actualización de aquellas circunscripciones que pasan a segunda vuelta dentro de un período electoral, mediante la informatización de dicho proceso, la posibilidad de garantizar una mayor seguridad en el proceso a partir de las listas de control de acceso (ACL), las cuales se encargan de proveer los permisos y los recursos a los diferentes funcionarios, y evitar la duplicidad de los datos estadísticos una vez culminado el proceso y el acercamiento de dichos datos a la realidad objetiva. Además de la entrega de un diseño de reportes, más descriptivo y legible que el anterior, a la Comisión Electoral Nacional para la

obtención de algunos indicadores en el sistema que necesitan ser consultados por los funcionarios periódicamente para su control, logrando elevar la integridad de los datos almacenados al disponer de una base de datos común con el Carné de Identidad que proporciona la información verídica de los ciudadanos que ahora pueden ser tratados como electores. También, una vez almacenada la información de cada período electoral permite establecer comparaciones entre distintos períodos para verificar si fue llevado a cabo el proceso de actualización, además de guardar todos los electores del período electoral en curso con el objetivo de evitar la duplicidad de los datos.

Conclusiones Parciales

Con el desarrollo de este capítulo se ha podido evidenciar que la especificación de estándares de codificación ofrece disímiles ventajas para las tareas de implementación, orientadas principalmente a lograr una mejor organización e interpretación del código. Además, los artefactos generados en la fase de implementación y prueba constituyen elementos fundamentales en el desarrollo de la solución propuesta. Con el diagrama de componentes se representa la vista lógica de la implementación, posibilitando la comprensión más a fondo del proceso. De igual forma con el diagrama de despliegue se obtiene una visión de la distribución de los nodos que serán necesarios para su expansión. Una vez implementada la solución fueron diseñados los diferentes casos de prueba logrando obtener en cada una de las iteraciones las no conformidades, de forma tal que se pudieran corregir y ofrecer versiones menos sensibles a errores.

Conclusiones Generales

Con la realización de esta investigación se efectuó un estudio del arte el cual permitió conocer la situación actual sobre aplicaciones relacionadas con censos electorales, actualización de listas y conteo de votos a nivel nacional e internacional, analizando sus ventajas y desventajas, e indicando la necesidad de implementar un nuevo sistema que se adapte a los estándares y normativas establecidas por la DIR.

Tras un estudio minucioso se validaron las herramientas y tecnologías que definieron los clientes para el desarrollo del proceso en cuestión. Además, el análisis de cada uno de los procesos involucrados en el negocio permitió ofrecer una propuesta mejorada de los mismos a partir de lo cual fueron determinados los requisitos funcionales y no funcionales de la propuesta de solución.

Como resultado del diseño se logró una aplicación web más robusta, desarrollada en ASP.NET, CSharp como lenguaje de programación y WWF como sustento del proceso en general apoyada en una arquitectura Cliente-Servidor de cinco capas estrechamente relacionadas e independientes entre sí.

La fase de implementación permitió establecer estándares de codificación con el objetivo de identificar los elementos de código que estarán presentes en dicho proceso, además de obtener una versión funcional de éste. A la vez de la realización de pruebas a las funcionalidades definidas, garantizando la calidad de las mismas, además de lograr identificar un conjunto de no conformidades las cuales se fueron erradicando en cada una de las iteraciones pautadas.

Recomendaciones

Se considera importante que una vez concluido el proceso de desarrollo se lleven a cabo las siguientes recomendaciones:

- Realizar el despliegue del proceso de reprocesamiento en las oficinas de la DIR.
- Una vez realizado el despliegue atender a la explotación de la aplicación a fin de encontrar elementos que puedan mejorarse en próximas versiones.
- Incrementar el paquete de reportes con el objetivo de brindar más flujo de información a los funcionarios y electores interesados.

Referencias bibliográficas

1. *Decreto ley, Ley No. 72.* 1992.
2. *Decreto Ley 248.* 2007.
3. Mujica, Ing. Héctor. *III Seminario para Técnicos Gestores de la Administración Electoral.*
4. Pérez, Jorge Lezcano. El sistema político de Cuba. *Trabajadores.* 4 de octubre de 2004.
5. Roesch, Marta León-. *Centro de Asesoría y Promoción Electoral (CAPEL): "El Registro Electoral. Los modelos de Argentina, Colombia, Costa Rica". Cuaderno de CAPEL No. 30, . 1era.. Edición, 1989.*
6. Microsoft Corporation. *MSF for CMMI Process Improvement.* Citado el: 22 de Enero de 2012.
7. Altova. [En línea] 26 de Marzo de 2012. <http://www.altova.com/es/umodel/uml-code-generation.html>..
8. BPMN. *BPMN.* [En línea] 2009. <http://www.bpmn.org/>.
9. Microsoft Corporation. MSDN. *MSDN.* [En línea] <http://msdn.microsoft.com/es-es/library/w0x726c2%28v=vs.100%29.aspx>.
10. Microsoft Corporation. MSDN. *MSDN.* [En línea]
11. Z, Reynier Blanco. *Documento de la arquitectura.*
12. Manual de Referencia Departamento Informática. [En línea] http://informatica.iessanclemente.net/manuais/index.php/Iniciaci%C3%B3n_en_Ajax.
13. Microsoft Corporation. [En línea] [http://msdn.microsoft.com/en-us/library/aa697427\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/aa697427(VS.80).aspx).
14. —. [En línea] <http://msdn.microsoft.com/en-us/netframework/aa663328>.
15. Oracle. *Oracle Database 11g.* [En línea] 26 de Marzo de 2012. <http://www.oracle.com/es/products/database/index.html>..
16. millan, jorge. blog. *blog.* [En línea] <http://jmillan.blog.com.es/2010/09/08/introduccion-a-entity-framework-parte-9342191/>.

17. Hernández, Octavio. *La tecnología LINQ: un avance revolucionario en la programación para .NET Framework*.
18. Embarcadero. E/R Studio Modelado de datos empresarial. Database Tools and Developer Software. [En línea] 10 de enero de 2011. <http://www.embarcadero.com>.
19. MSDN. *MSDN*. [En línea] <http://msdn.microsoft.com/es-es/library/ms225593%28v=vs.90%29.aspx>.
20. Corporation, Microsoft. [En línea] [Citado el: 5 de febrero de 2013.] <http://www.msdn.com/epages/62042259.sf?ObjectPath=/Shops/62042259/Products/%22Embarcadero%20ER/Studio%22..>
21. Thayer, Richard H. y Dorfman, Merlin. *IEEE Software Requirement Engineering, Second Edition*. . New York : s.n., 1997.
22. —. *IEEE Software Requirement Engineering, Second Edition*. New York : s.n., s.n., 1997. ISBN: 978-0-8186-7738-0.
23. EcuRed. [En línea] [Citado el: 19 de febrero de 2013.] http://www.ecured.cu/index.php/Arquitectura_de_software.
24. Identificación Inmigración y Extranjería de la República de Cuba, Proyecto. *Documento de Arquitectura de Software*. La Habana, Cuba : s.n., 2010.
25. —. *Documento de Arquitectura de Software*. La Habana, Cuba : s.n., 2010.
26. de Laurentiis Gianni, Renato. *Workflow-Tecnología para la Integración y Orquestación de Procesos, Sistemas y Organización*. Madrid, España : IBERICA IT Group.
27. Vega, Erik de la. Estándares de Codificación para el Proyecto Identidad Cuba.
28. Jacobson, I. and Rumbaugh, G. Booch y J. *El Proceso Unificado de Desarrollo de software*.
29. [En línea] [Online] <http://www.editum.org/autor=-.html>.

Bibliografía Consultada

<http://www.base100.com/trabajos.html>

http://www.diuc.uc.edu.ve/d_desarrollo_productos.php

<http://www.efec.edu.do/Programas/ProgramadelRegistrodelEstadoCivil.aspx>

<http://www.indracompany.com/soluciones-y-servicios/solucion/procesos-electorales/109/nuestra-oferta>

<http://msdn.microsoft.com/es-es/library/bb972240.aspx>

http://www.ecured.cu/index.php/Patr%C3%B3n_de_dise%C3%B1o_de_software

<http://sourcemaking.com/patr%C3%B3n-de-dise%C3%B1o>

<http://siul02.si.ehu.es/~alfredo/iso/06Patrones.pdf>

<http://es.kioskea.net/contents/221-workflow-gestion-de-los-procesos-comerciales>

<http://msdn.microsoft.com/en-us/vstudio/jj684582.aspx>

<http://www.crystalreports.com/>

www54.sap.com/solution/.../crystal-reports/index

<http://mipais.cuba.cu/cat.php?idcat=77&idpadre=76&nivel=2>

http://www.ecured.cu/index.php/Sistema_electoral_cubano

http://www.conevyt.org.mx/cursos/cursos/mexico/contenidos/recursos/revista/4_1.htm

<http://www.cip.cu/contenido/menu2/compendios-informativos-1/elecciones-en-cuba-2010.pdf>

<http://es.wikipedia.org/wiki/Elecciones>

<http://www.buenastareas.com/ensayos/Que-Es-Un-Proceso-Electoral/213818.html>

Cillero, Manuel. Actividad IAS 6: Pruebas de Aceptación del Sistema. [Online] [Cited: Mayo 12, 2011.]
<http://manuelcillero.com/metrica-3/procesos-principales/ias/actividad-6/tarea-1>

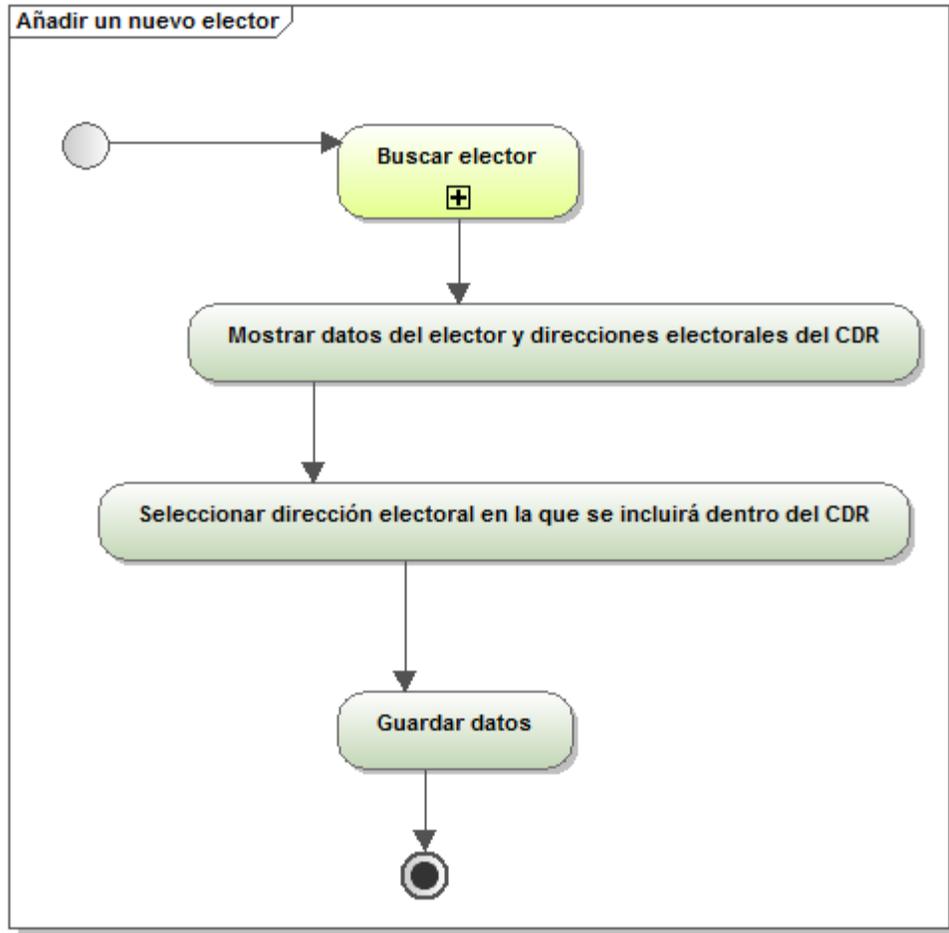
DanySoft. Cursos de Visual Studio 2010. Servicios Profesionales Danysoft. [Online] Septiembre 7, 2010.
[Cited: Diciembre 9, 2010.] www.danysoft.com.

Gracia, Joaquin. CMM-CMMI. Calidad. Ingenieria de Software. [Online] Agosto 14, 2005. [Cited: Julio 24,
2010.] <http://www.ingenierossoftware.com/calidad/cmm-cmmi.php>.

Visconti, Marcello and Astudillo, Hernán. Fundamentos de Ingeniería de Software. [Online] [Cited: Mayo
9, 2011.] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/15-Implementacion.pdf>.

Anexos

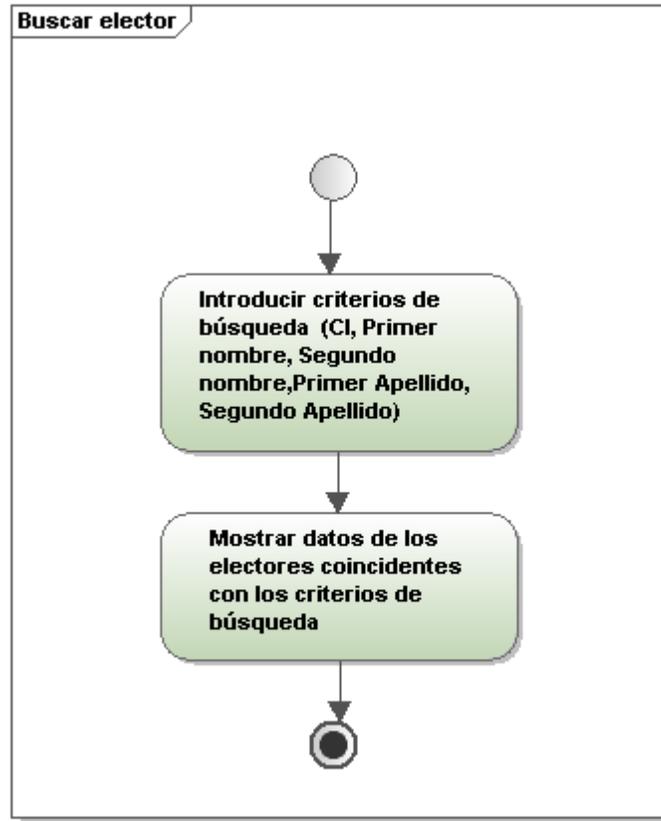
Procesos mejorados



Generated by UModel

www.altova.com

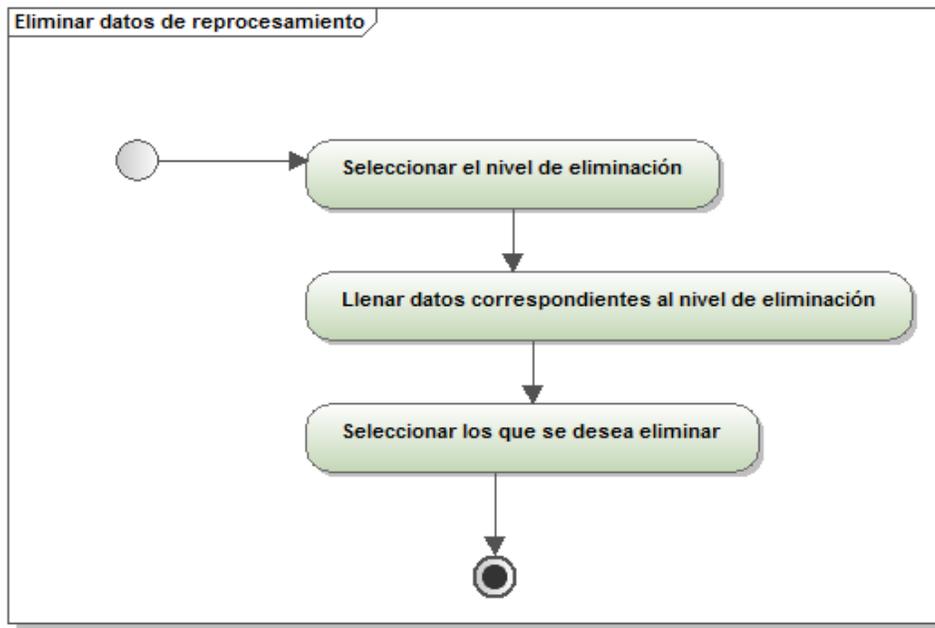
Fig.22 Diagrama "Añadir un nuevo elector"



Generated by UModel

www.altova.com

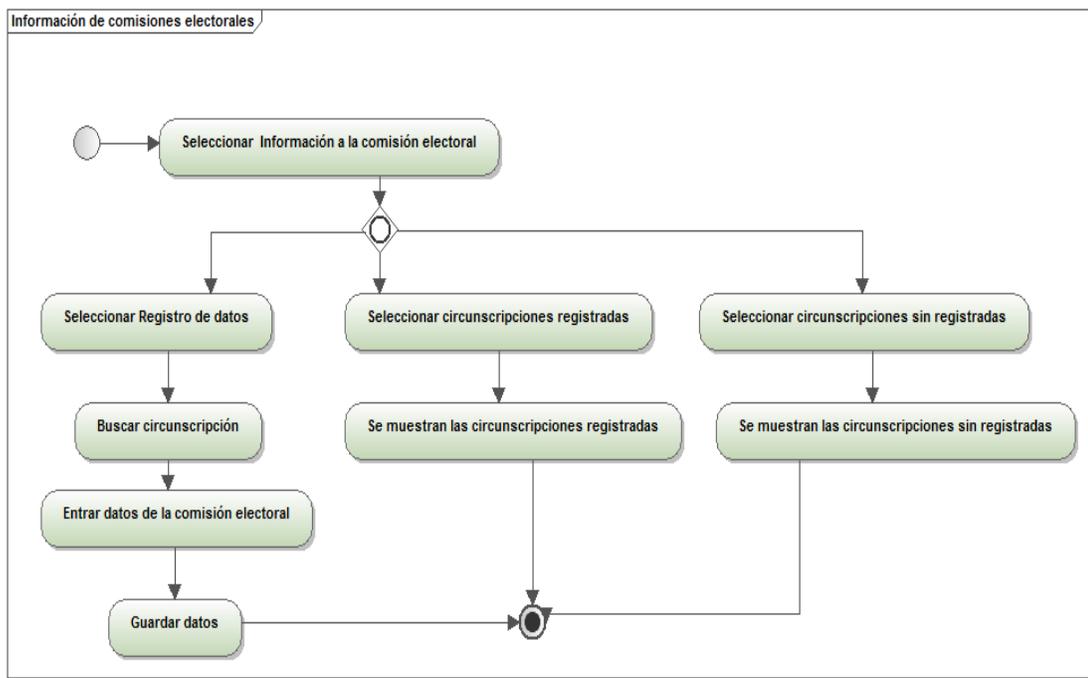
Fig.23 Diagrama "Buscar elector"



Generated by UModel

www.altova.com

Fig.20 Fig.24 Diagrama “Eliminar datos del reprocesamiento”



Generated by UModel

www.altova.com

Fig.25 Diagrama “Información de comisiones electorales”

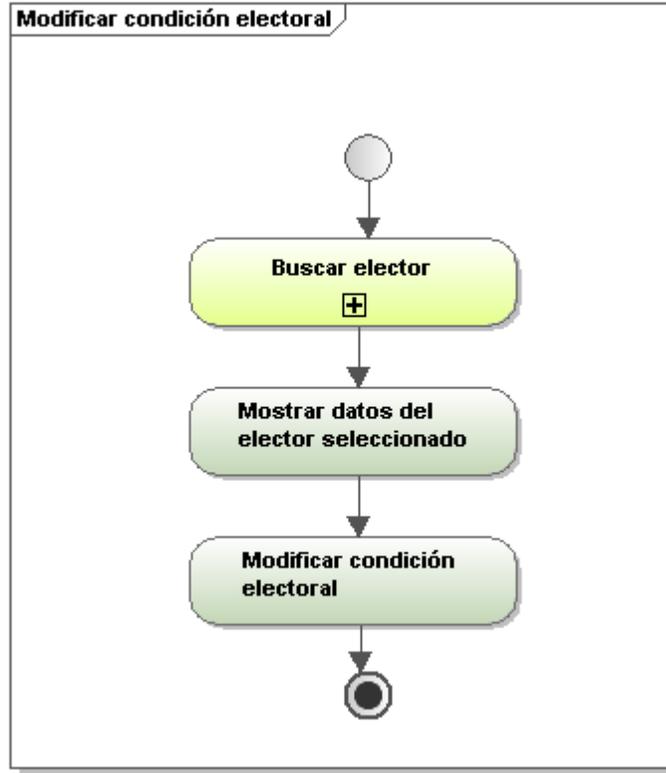


Fig.26 Diagrama "Modificar condición electoral"

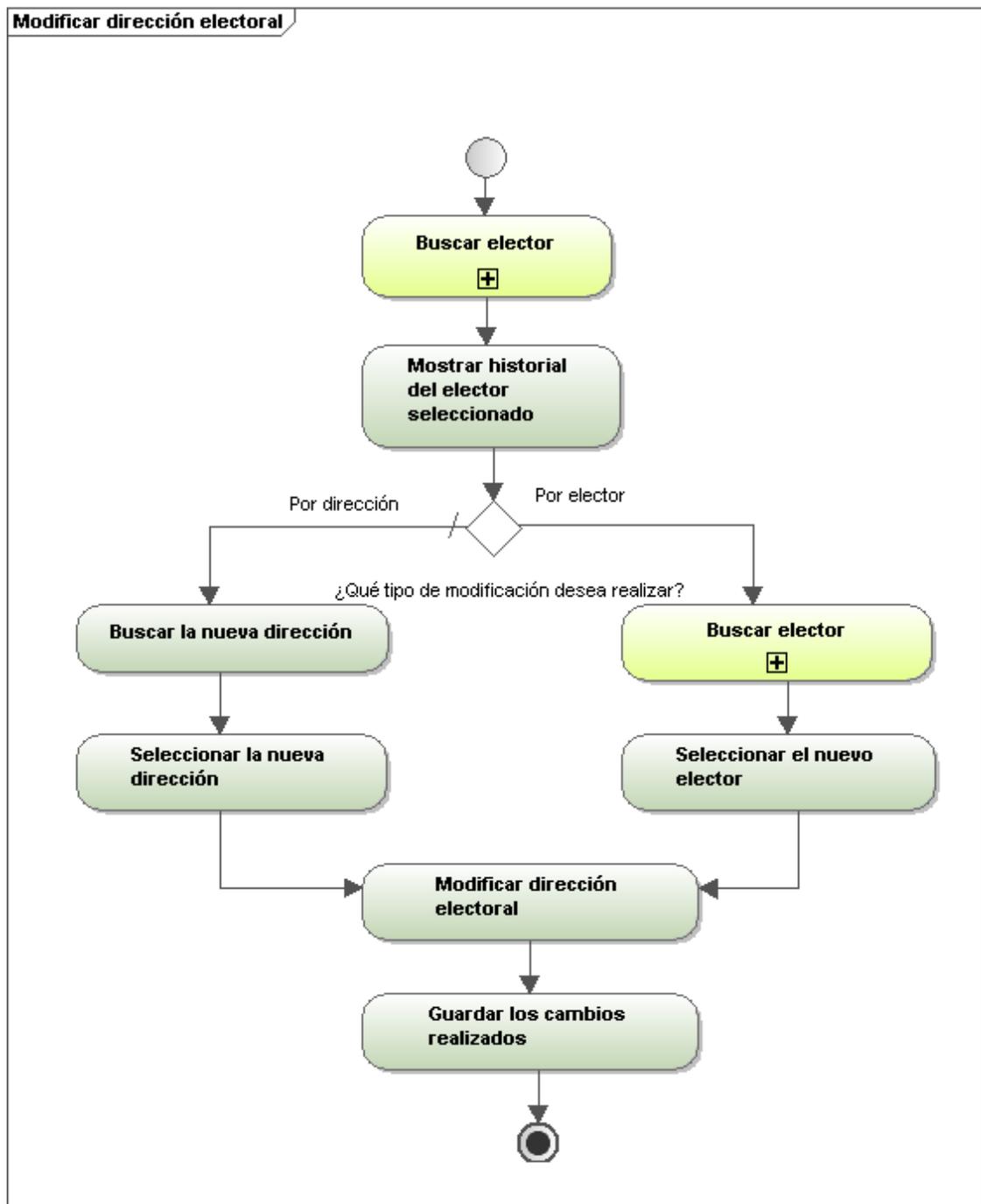
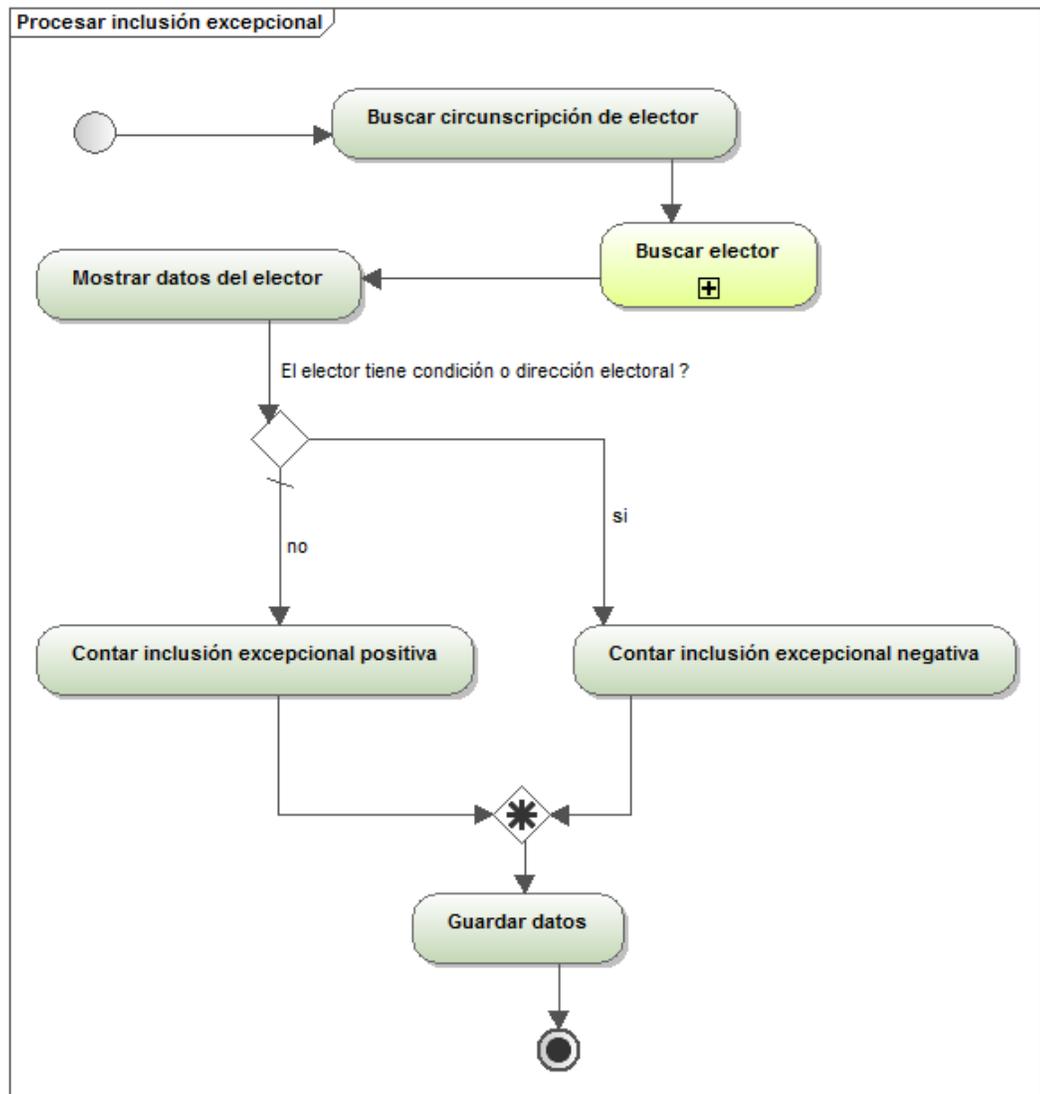


Fig.27 Diagrama “Modificar dirección electoral”



Generated by UModel

www.altova.com

Fig.28 Diagrama "Procesar inclusión excepcional"

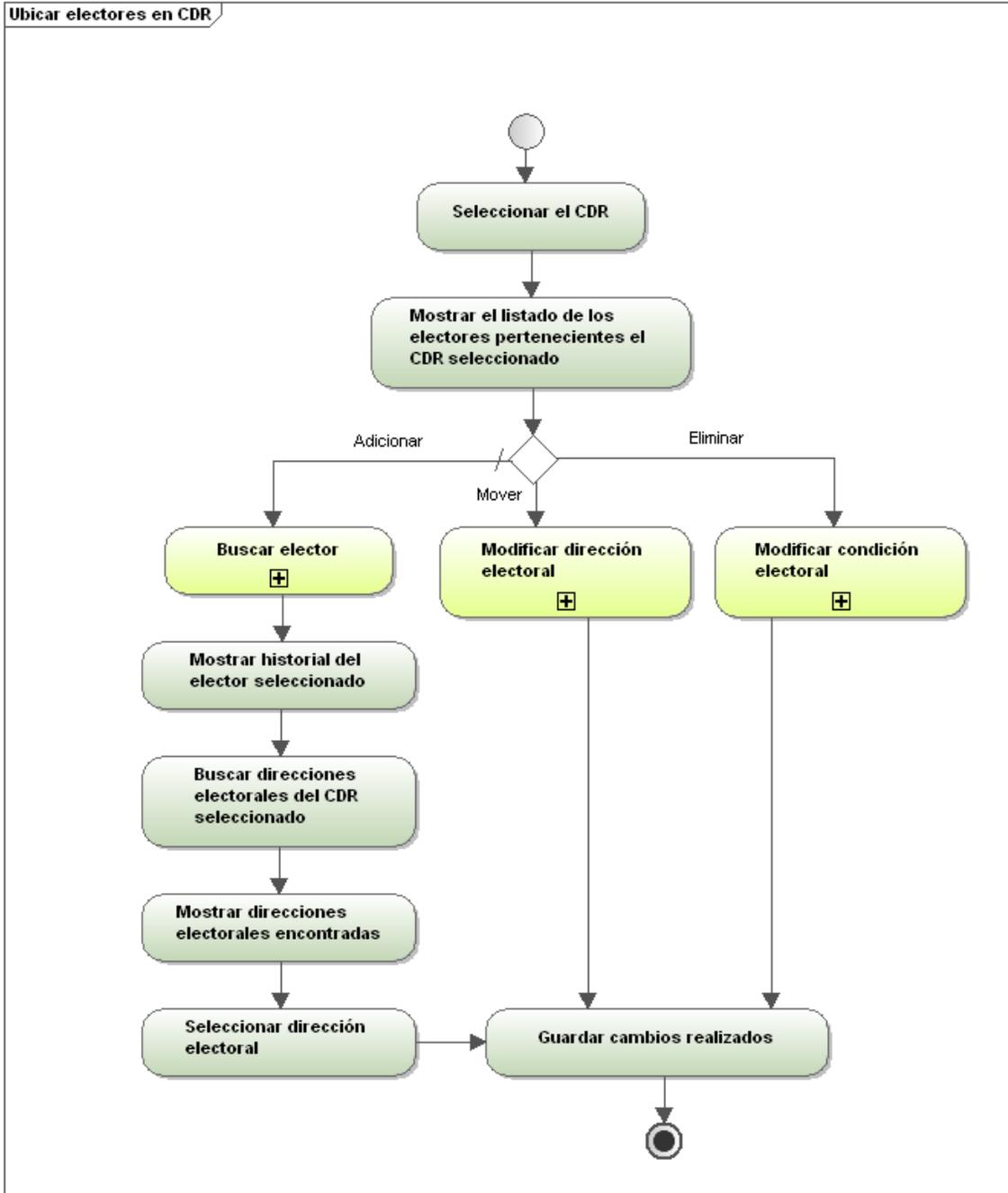
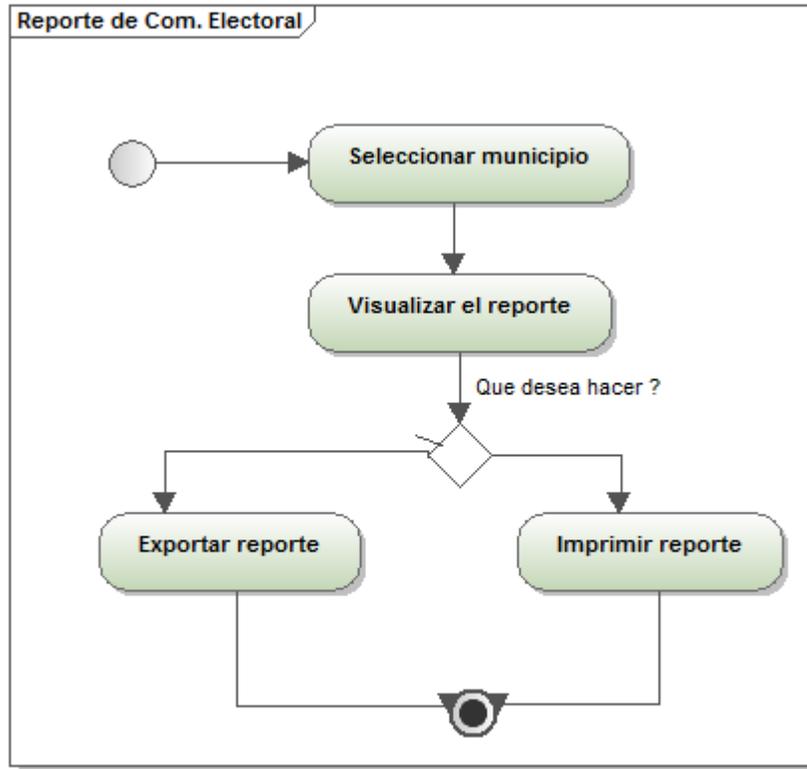


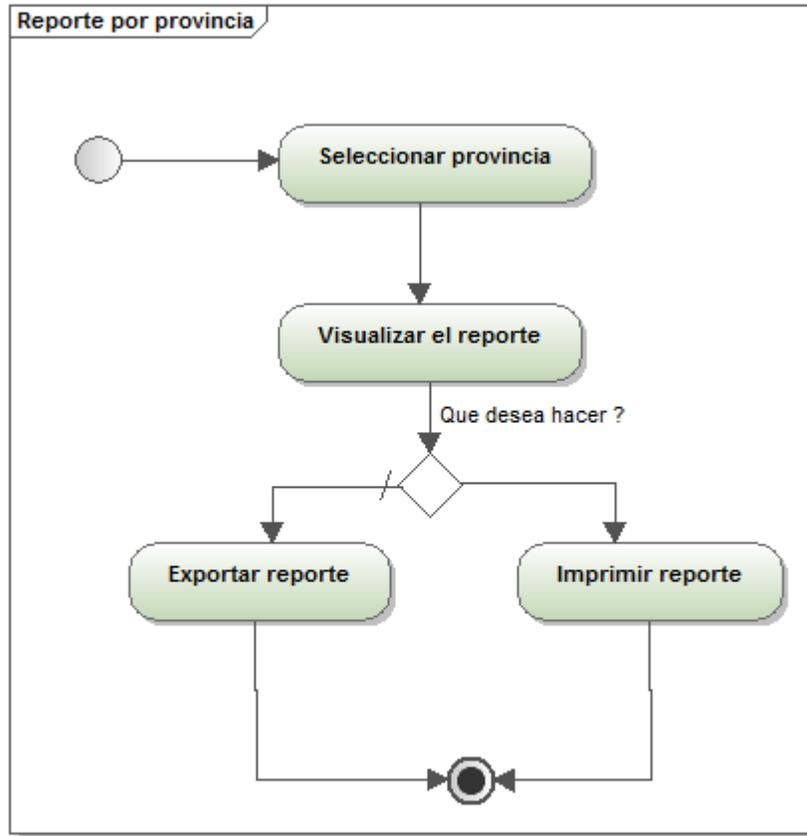
Fig.29 Diagrama "Ubicar electores en CDR"



Generated by UModel

www.altova.com

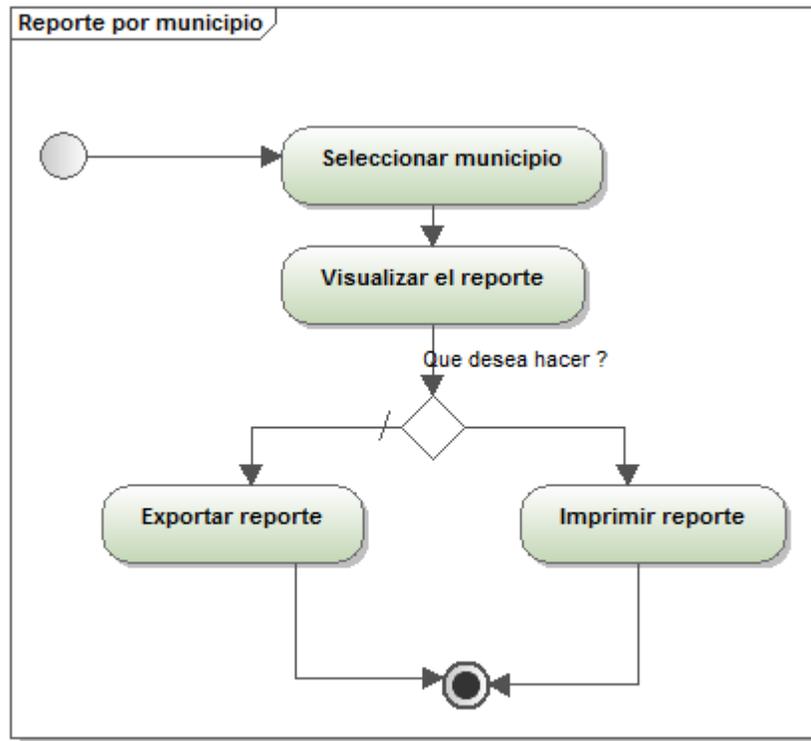
Fig.30 Diagrama “Reporte de la Comisión Electoral”



Generated by UModel

www.altova.com

Fig.31 Diagrama "Reporte por provincias"



Generated by UModel

www.altova.com

Fig.32 Diagrama “Reporte por municipios”

Descripción de requisitos funcionales

RF2. Procesar inclusión excepcional

Propósito	Permite incluir un elector	
Roles	Administrador municipal	
Precondiciones	El administrador municipal debe haberse autenticado.	
Entidades tratadas	Entidad	Atributos
	dpersona	nombre apellido dirección Electoral condición Electoral dirección Permanente
Descripción	1.2 Introducir criterios de búsqueda de circunscripción del elector: c) Provincia. d) Municipio.	

	<p>c) Circunscripción.</p> <p>1.2 Mostrar datos de la circunscripción:</p> <ul style="list-style-type: none"> g) Nombre circunscripción h) Numero circunscripción i) Consejo popular. j) Provincia. k) Municipio. l) Listado de electores <ul style="list-style-type: none"> 1) <i>Foto.</i> 2) <i>Nombres.</i> 3) <i>Apellidos.</i> 4) <i>Carné de Identidad.</i> 5) <i>Dirección electoral.</i> <ul style="list-style-type: none"> • <i>Provincia.</i> • <i>Municipio.</i> • <i>Consejo Popular.</i> • <i>Circunscripción.</i> • <i>Dirección permanente del candidato.</i> <p>1.3 Seleccionar el elector a incluir</p> <p>1.4 Mostrar datos del elector:</p> <ul style="list-style-type: none"> f) <i>Foto.</i> g) <i>Nombre.</i> h) <i>Apellido.</i> i) <i>Carné de Identidad.</i> j) <i>Dirección electoral.</i> <ul style="list-style-type: none"> • <i>Provincia.</i> • <i>Municipio.</i> • <i>Consejo Popular.</i> • <i>Circunscripción.</i> • <i>Dirección permanente del candidato.</i> <p>1.5 Clasificar la inclusión excepcional en “positiva” o “negativa”.</p> <p>1.5.1 Seleccionado “positiva”.</p> <ul style="list-style-type: none"> 1.5.1.1 Contar inclusión excepcional “positiva”. 1.5.1.2 Modificar la dirección y condición electoral. <p>1.5.2 Seleccionado “negativa”.</p> <ul style="list-style-type: none"> 1.5.2.1 Contar inclusión excepcional “negativa”. 1.5.2.2 Mantengo dirección y condición electoral.
Validaciones	-

Tabla 2. Especificación del requisito Procesar inclusión excepcional

Postcondiciones	Quedan actualizadas las listas electorales
Prototipo	

RF3. Procesar exclusión

Propósito	Permite excluir un elector
Roles	Administrador municipal

Informatización del proceso de reprocesamiento del registro electoral en el Sistema Único de Identificación Nacional (SUIN)

Prototipo Interfaz: Procesar inclusión excepcional

Precondiciones	El administrador municipal debe haberse autenticado.	
Entidades tratadas	Entidad	Atributos
	dpersona	Nombre Apellido Dirección Electoral Condición Electoral Dirección Permanente
Descripción	<p>1.1 Introducir criterios de búsqueda de circunscripción del elector:</p> <ul style="list-style-type: none"> a) Provincia. b) Municipio. c) Circunscripción. <p>1.2 Mostrar datos de la circunscripción:</p> <ul style="list-style-type: none"> a) Nombre circunscripción b) Numero circunscripción c) Consejo popular. d) Provincia. e) Municipio. f) Listado de electores <ul style="list-style-type: none"> 1) Foto. 2) Nombres. 	

	<p>3) Apellidos. 4) Carné de Identidad. 5) Dirección electoral.</p> <ul style="list-style-type: none"> • Provincia. • Municipio. • Consejo Popular. • Circunscripción. • Dirección permanente del candidato. <p>1.3 Seleccionar el elector a excluir:</p> <p>1.4 Mostrar datos del elector:</p> <p>a) Foto. b) Nombre. c) Apellido. d) Carné de Identidad. e) Dirección electoral.</p> <ul style="list-style-type: none"> • Provincia. • Municipio. • Consejo Popular. • Circunscripción. • Dirección permanente del candidato. <p>1.5 Seleccionar opción Aceptar.</p> <p>1.5.1 Contar la exclusión.</p> <p>1.5.2 Guardar el motivo de la exclusión.</p>
Validaciones	-
Postcondiciones	Quedan actualizadas las listas electorales

Prototipo Interfaz: Procesar exclusión

Tabla 3. Especificación del requisito Procesar exclusión

RF4. Generar reportes

Propósito	Obtener reportes estadísticos para la actualización de las circunscripciones que van a segunda vuelta	
Roles	Administrador provincial	
Precondiciones	El administrador provincial debe haberse autenticado.	
Entidades tratadas	Entidad	Atributos
	dpersona	nombre apellido dirElect condElect dirPerm
Descripción	1.1 Mostrar un menú para permitir seleccionar el tipo de reporte que se desea obtener:	

	<p>a)Electores registrados por municipio</p> <p>b)Electores registrados por provincia</p> <p>3.2 Permitir seleccionar el criterio de búsqueda.</p> <p>3.2.1 Si el reporte seleccionado es Electores registrados por municipio:</p> <p>3.2.1.1 Permitir seleccionar:</p> <p>Provincia (Obligatorio) Formato (Combo Box)</p> <p>Municipio (Obligatorio) Formato (Combo Box)</p> <p>3.2.2 Si el reporte seleccionado es Electores registrados por provincia:</p> <p>3.2.2.1 Permitir seleccionar:</p> <p>Provincia (Obligatorio) Formato (Combo Box)</p> <p>3.2.3 Si selecciona Aceptar: ver 3.2</p> <p>3.2.4 Si selecciona Cancelar: mostrar la página principal.</p> <p>3.2.5 Mostrar una vista previa del reporte.</p> <p>3.2.5.1 Si selecciona Imprimir: imprimir el reporte.</p> <p>3.2.5.2 Si se selecciona Guardar: guardar el archivo en un dispositivo de almacenamiento.</p> <p>3.2.5.3 Si selecciona Cancelar: mostrar la página principal.</p>
Validaciones	1. Los reportes deben tener el formato especificado.
Postcondiciones	1. Han quedado guardados o impresos los reportes deseados.

Prototipo

suin		SISTEMA ÚNICO DE IDENTIFICACIÓN NACIONAL		Resultado de reprocesamiento por provincia						08/05/2013
				Período electoral: Elecciones Generales 2012						
				Provincia: Granma						
MUN	INC	INC.NEG	TOTAL	EXC	EXC.NEG	TOTAL	INC.E	INC.E.NE	TOTAL	
Río Cauto	0	0	0	0	0	0	0	0	0	
Bartolomé Masó	0	0	0	0	0	0	0	0	0	
Yara	0	0	0	0	0	0	0	0	0	
Niquero	0	0	0	0	0	0	0	0	0	
Buey Arriba	0	0	0	0	0	0	0	0	0	
Jiguani	0	0	0	0	0	0	0	0	0	
Bayamo	0	0	0	0	0	0	0	0	0	
Guisa	0	0	0	0	0	0	0	0	0	
Cauto Cristo	0	0	0	0	0	0	0	0	0	
Manzanillo	0	0	0	0	0	0	0	0	0	
Media Luna	0	0	0	0	0	0	0	0	0	
Pilón	0	4	4	1	2	3	2	1	3	
Campechuela	0	0	0	0	0	0	0	0	0	
	0	4	4	1	2	3	2	2	3	

MUN: Municipio INC: Inclusión EXC: Exclusión INC.NEG: Inclusión Negativa
 EXC.NEG: Exclusión Negativa INC.E: Inclusión Excepcional INC.E.N: Inclusión Excepcional Negativa

suin		SISTEMA ÚNICO DE IDENTIFICACIÓN NACIONAL		Resultado de reprocesamiento por municipio						08/05/2013
				Período electoral: Elecciones Generales 2012						
				Provincia: Granma Municipio: Pílon						
CIR	INC	INC.NEG	TOTAL	EXC	EXC.NEG	TOTAL	INC.E	INC.E.NE	TOTAL	
1	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	1	0	1	
3	0	0	0	1	1	2	0	0	0	
4	0	0	0	0	0	0	0	0	0	
5	0	4	4	0	0	0	1	1	2	
6	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	1	1	0	0	0	
8	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	

Prototipo Interfaz: Generar reportes

Tabla 4. Especificación del requisito Generar reportes

Casos de prueba

Caso de prueba “Procesar Inclusiones”

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Buscar circunscripción del elector	Permite seleccionar la circunscripción a la cual se le desea incluir un elector	Muestra la interfaz para especificar los datos del elector a incluir	1.Buscar circunscripción 2.Aceptar

Informatización del proceso de reprocesamiento del registro electoral en el Sistema Único de Identificación Nacional (SUIN)

EC 1.2 Buscar elector	Permite adicionar los datos del elector a incluir	Muestra la interfaz para especificar los datos del elector a incluir	1.Buscar elector 2.Aceptar
EC 1.3 Clasificar el tipo de inclusión	Permite clasificar la inclusión en "positiva" o "negativa"	Muestra la interfaz donde están los datos del elector buscado y da la posibilidad de especificar qué tipo de inclusión será dicho elector	Si selecciona "positiva": <ul style="list-style-type: none"> Se cuenta la inclusión como positiva Se modifica la dirección y condición electoral Si selecciona "negativa": <ul style="list-style-type: none"> Se cuenta la inclusión como negativa Mantengo dirección y condición electoral.

Caso de prueba "Procesar exclusiones"

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Buscar circunscripción del elector	Permite seleccionar la circunscripción a la cual se le desea excluir un elector	Muestra la interfaz para especificar los datos del elector a excluir	1.Buscar circunscripción 2.Aceptar
EC 1.2 Buscar elector a excluir	Permite insertar los datos del elector a excluir	Muestra la interfaz para especificar los datos del elector a excluir	1.Buscar elector 2.Aceptar
EC 1.3 Clasificar el motivo de la exclusión	Permite especificar la causa y la categoría de la exclusión	Muestra la interfaz donde están los datos del elector buscado y da la posibilidad de especificar el motivo de la exclusión	Selecciona dentro del motivo de la exclusión la categoría y la causa de esta y se cuenta la exclusión

Caso de prueba "Generar reportes"

Escenario	Descripción	Provincia	Municipio	Respuesta del sistema	Flujo central
EC 1.1 Obtener electores registrados por municipio.	Permite obtener un reporte de electores registrados en un municipio dado.	V/Granma	V/Jiquaní	Se muestra el reporte.	1.Obtener electores registrados por municipio 2.Aceptar
		Vacío	Vacío	Se muestra un mensaje indicando que se deben seleccionar los campos obligatorios.	
EC 1.3 Obtener electores registrados por provincia.	Permite obtener un reporte de electores registrados en una provincia dada.	V/Granma	NA	Se muestra el reporte.	1.Obtener electores registrados por provincia 2.Aceptar
		Vacío	NA	Se muestra un mensaje indicando que se deben seleccionar los campos obligatorios.	

