

**Universidad de las Ciencias Informáticas**

**Facultad 1**



**Título: Módulo de Administración para el Sistema Único de  
Identificación Nacional de la República de Cuba**

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

**Autores:** Diana Venero Paez

Alejandro Valdés González

**Tutor:** Ing. Brisey López Bello

*Ciudad Habana, junio de 2010*

*“Año 52 de la Revolución”*

*Las producciones intelectuales serán el sustento fundamental de Cuba. La idea es convertir la informática en una de las ramas más productivas y aportadoras de recursos para la nación. . .*

*Fidel Castro*

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los días \_\_\_\_\_ del mes \_\_\_\_\_ del año \_\_\_\_\_.

---

Diana Venero Paez

---

Alejandro Valdés González

---

Brisey López Bello

## DATOS DEL CONTACTO

Tutora:

Ing. Brisey López Bello

- Ingeniera en Ciencias Informáticas, UCI 2008, Título de Oro.
- Analista principal del Proyecto “Sistema Único de Identificación Nacional de la República de Cuba”.
- Profesora del Departamento de Sistemas Digitales, Facultad 1, Universidad de las Ciencias Informáticas.
- Correo electrónico: [bbello@uci.cu](mailto:bbello@uci.cu)

Consultante:

Msc. Yudenia Ramírez Mastrapa

- Ingeniera Informática, CUJAE 2003.
- Categoría docente: Profesora Asistente.
- Máster en Gestión de Proyectos Informáticos.
- Profesora del Departamento de Sistemas Digitales, Facultad 1, Universidad de las Ciencias Informáticas.
- Jefa del Departamento de Soluciones de software del Centro de Identificación y Seguridad Digital.
- Correo electrónico: [yudenia@uci.cu](mailto:yudenia@uci.cu)

## AGRADECIMIENTOS

*Quiero agradecerle primero que todo a mi madre, porque por ella es que he logrado todo en mi vida, a mi hermano Leo por ser mi guía, mi meta, mi inspiración la persona que siempre está a mi lado apoyándome en todo gracia Leo, a mi abuelito papi que no está a mi lado físicamente pero sé que siempre está a mi lado dándome la luz que me hace falta para que las cosas me salgan bien, a mi padre que sé que está muy orgulloso de las cosas que he logrado y eso siempre me ha levantado en los peores momentos porque sé que siempre contó conmigo, a mi abuelita mima que es mi segunda madre y siempre me ayudó en todo desde mis inicios de estudios y mi otro padre Pepe que tengo mucho que agradecerle ya que fue quien me impulsó a estudiar y me hizo comprender en muchas ocasiones que el estudio sería lo mejor para mi, agradecerle a mi nene (Dule) linda que siempre ha estado conmigo en la buenas y en las malas, a mis suegros que son especiales, a mi cuñadita gracias a todos. A mi tía Hilda, Rosi a Marco por todos sus consejos, a mi tío Lino, a Hidalma y a Yasiel. A mis padrinos que se que ellos están muy orgulloso de mí y yo de ellos.*

*Agradecer a mi compañera de tesis Dianita por todo el trabajo que juntos pasamos, las noche trabajando y la dedicación que ella le brindó a nuestra tesis, aunque se creyó en algunos momentos que era mi jefa cuando en si fue una secretaria ejemplar, jajaja.*

*A mi tutora Brise que siempre estuvo con nosotros y nos dio mucho apoyo y fue muchas veces quien nos unió para lograr una buena comunicación entre nosotros gracias por toda tu ayuda y por ayudarnos a crecer profesionalmente ya que con tus exigencias nos ayudaste mucho Bri y te lo agradezco.*

*A nuestra líder Yude que siempre nos apoyó en todo y supo guiarnos como buena líder que ella es.*

*A Blanco que fue una mano amiga que siempre estuvo de nuestro lado.*

*A Humberto que fue una gran ayuda en nuestro trabajo gracias de corazón.*

*Agradecerle a todo el piquete de 202, a mis amigos Alejo el calvo, al Reisbel, Yadira, René, Anita, a Armando (el cuco) que siempre me ayudó cuantas veces me hizo falta, a todos muchas gracias porque gracias a ustedes aprendí mucho y crecí profesionalmente.*

*A mi piquete de Leoncio (Hector, Pander, Danyer) a todos ustedes que siempre han sido mis amigos y hermanos y me apoyaron siempre que los necesité.*

*A mi gente de la facultad 7, el Yoan mi hermanito que siempre tuvimos tremenda amistad y me apoyaste en todo y siempre estuvimos en las buenas y en la malas, a mi hermanita Danay (La negra) que fue mi amiga y siempre estuvo a mi lado en todo momento, al Meme, a toda mi gente de la 7 que siempre creyeron en mí, a todos gracias. A Yenisel que fue como quien dice mi madrina y siempre me apoyó y creyó en mí.*

*A todos los que pensaron que yo no podía lograrlo, ellos fueron de gran ayuda porque fueron quienes me impulsaron a derribar cada obstáculo que se presentó en el camino hasta llegar al final.*

*“Gracias a todos”.*

**Ale**

*A la Revolución, por darme la oportunidad de servirle desde el corazón y el trabajo.*

*A la UCI, por haberme hecho crecer como persona y como profesional, por formarme, por ser nuestra casa todos estos años.*

*A mi mamá por darme la vida, por su cariño, su confianza, su apoyo, su dedicación y sus consejos, te adoro mamita.*

*A mi papá por estar tan orgulloso de mí y por quererme tanto.*

*A mima, a papi y Franki por haberme educado tan bien, por haberme guiado por el camino correcto cuando a veces dudé, por haber estado siempre a mi lado y no darme la espalda.*

*A mamá porque siempre tuvo mucha fé y confianza en mí, a papá porque aunque no está presente siempre luchó porque mis sueños se hicieran realidad, y porque creyó mucho en que lo lograría, te quiero mucho papá.*

*Al pipo, a tía, a tío, a Deysi, a Lysni, Ede, Tomy, a Lis, por su cariño, porque han sido mis hermanitos todos estos años.*

*A mi hermanita, que aunque está chiquitica la quiero mucho.*

*A Orli, por quererme tanto todos estos años, por haberme apoyado y ayudado, por su comprensión, por ser tan bueno conmigo y hacerme tan feliz, y porque lo quiero mucho, a sus padres por querer lo mejor para mí, por ayudarme tanto durante tantos años, y porque siempre he sido como una hija para ellos, a su hermana y su abuelito Pempe, que me quiere mucho.*

*A Ale, por ser tan buen amigo, por ayudarme tanto, por su dedicación para que todo saliera bien, por haber sido tan buen secretario en la realización del Módulo, jejeje, gracias Ale.*

*A Bri, mi amiga y tutora, por haberme enseñado tanto, habernos ayudado para que este sueño se hiciera realidad y hoy ale y yo fuéramos tan buenos amigos, gracias michi.*

*A Yude mi segunda tutora, porque siempre nos ayudó, estuvo al tanto de todo lo que nos hacía falta, y nos dio muy buenos consejos.*

*A Blanco y Humbe, por habernos dedicado tanto tiempo, por su ayuda incondicional, muchas gracias a los dos.*

*A todos mis amigos de la 8, que durante 4 años aguantaron a esta pinareña, jejeje, a Yadi por haberme apoyado tanto, y a pesar de la distancia no se ha olvidado de mí, a Meylin por haberme aconsejado tanto, a Dayana, Elvin, Yaillet, Yudi, Grenia, a Anelis, a Yadi mi flaquita de pinar, a Lucas, a Yoli, porque siempre estuvo ahí para escucharme cuando lo necesité, a los varones de mi grupo, gracias todos.*

*A Yadi que ha sido más que una amiga para mí, por querer lo mejor para mí, por haberme enseñado a crecerme ante los problemas, a ser fuerte, a tener decisión propia y haberme querido y apoyado durante estos meses, muchas gracias yadi.*

*A Pache por ser tan bueno conmigo y ayudarme tanto, gracias de verdad.*

*A mis amigos de 202 por haberme ayudado en el proyecto, y en la tesis, a Reisbel por sus horas, por haberme enseñado tanto, a Anita, Eli por ser tan buenos amigos, a Linnet por ser tan sincera, por ser tan dura en sus consejos, por su ayuda en el documento, a Jeandy por hacerme reír tanto, a René y Ale Torres que siempre estuvimos muy unidos, para que hoy todos pudiéramos graduarnos, a Adrian, Elizabeta, a Yoan, Daniel y Ale Dueñas, gracias a todos, han sido muy buenos conmigo.*

*Gracias a todos.*

**Diana**



## DEDICATORIA

*A mami y papito.*

*A mis abuelos mima, papi, y mamá.*

*A mis tíos pipo, Franki, tía y Deysi.*

*A mis primos Ede, Lysni, Lis y Tomy.*

*A Orli.*

*A mi familia.*

*A mis amigos.*

**Diana**

*A mi madre por ser mi vida.*

*A mi hermano Leo por ser mi guía.*

*A cada uno de mis hermanos.*

*A mi papá.*

*A mi abuela Mima, a Papi y a Pepe.*

*A mi hermano el Negrón.*

*A mis queridos padrinos.*

*A mi familia y Amigos.*

**Ale**

## RESUMEN

Llevar el control estricto de la identificación de la población es una de las estrategias trazadas por el país desde el triunfo de la Revolución, con el objetivo de garantizar la seguridad ciudadana y el orden interior, proceso gestionado por la Dirección de Identificación y Registros (DIR) del Ministerio del Interior (MININT) desde el año 1997. En el presente trabajo de diploma se presenta un Módulo de Administración que tiene como objetivo la gestión de los elementos de configuración del Sistema Único de Identificación Nacional de la República de Cuba (SUIN). Es una aplicación web basada en procesos, desarrollada en un entorno orientado a las tecnologías de Microsoft en la versión 3.5 de .NET Framework. Con la implantación de este módulo común en el sistema se logrará una estandarización en los procesos de gestión de direcciones de las personas, así como la manipulación normalizada de los nomencladores<sup>1</sup> utilizados en los procesos llevados a cabo en las oficinas del Carné de Identidad y Registro de la Población (CIRP) y la asignación homogénea de las direcciones a todos los ciudadanos cubanos y extranjeros residentes en el país. La investigación abarca desde las actividades relacionadas con el modelado del negocio hasta la fase de prueba de las funcionalidades implementadas.

---

<sup>1</sup> Lista de objetos definidos.

**ÍNDICE DE CONTENIDOS**

**DECLARACIÓN DE AUTORÍA.....I**

**DATOS DEL CONTACTO..... II**

**AGRADECIMIENTOS..... III**

**DEDICATORIA..... VII**

**RESUMEN..... VIII**

**INTRODUCCIÓN..... - 1 -**

**CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....6**

**1.1 INTRODUCCIÓN.....6**

**1.2 ANTECEDENTES DE LOS PROCESOS DE ADMINISTRACIÓN DE LAS APLICACIONES EN LAS OFICINAS DEL CIRP .....6**

**1.3 CONCEPTOS FUNDAMENTALES EN EL DOMINIO DEL PROBLEMA .....6**

1.3.1 SISTEMA DE GESTIÓN DE SEGURIDAD DE LA INFORMACIÓN (SGSI)..... 6

1.3.2 ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS..... 7

1.3.3 GESTIÓN DE DIRECCIONES ..... 11

1.3.4 GESTIÓN DE LA INFORMACIÓN .....14

**1.4 ANÁLISIS DE SOLUCIONES EXISTENTES ..... 14**

**1.5 ¿POR QUÉ LA NECESIDAD DE UN MÓDULO DE ADMINISTRACIÓN PARA EL SUIN? ..... 17**

**1.6 AMBIENTE DE DESARROLLO..... 17**

1.6.1 METODOLOGÍA DE DESARROLLO .....17

1.6.2 MODELADO DE PROCESOS DE NEGOCIO.....18

1.6.3 SISTEMA GESTOR DE BASE DE DATOS (SGBD) .....19

1.6.4 HERRAMIENTAS PARA EL ENTORNO DE DESARROLLO .....20

**1.7 CONCLUSIONES PARCIALES..... 25**

**2 CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA ..... 26**

<b>2.1 INTRODUCCIÓN.....</b>	<b>26</b>
<b>2.2 MODELO DE DOMINIO .....</b>	<b>26</b>
2.2.1 DIAGRAMA DE CLASES DEL MODELO DE DOMINIO.....	27
2.2.2 DESCRIPCIÓN DEL DIAGRAMA DE CLASES DEL MODELO DE DOMINIO.....	28
2.2.3 GLOSARIO DE CONCEPTOS DEL MODELO DE DOMINIO.....	28
<b>2.3 CONCEPCIÓN DE LOS MÓDULOS DEL SISTEMA .....</b>	<b>29</b>
2.3.1 ROLES DEL MÓDULO DE ADMINISTRACIÓN.....	30
2.3.2 ESPECIFICACIÓN DE LOS REQUISITOS FUNCIONALES (ERF).....	31
2.3.3 MODELO DE PROCESO MEJORADO .....	32
2.3.4 DESCRIPCIÓN DE REQUISITOS FUNCIONALES.....	34
2.3.5 PROTOTIPO DE INTERFAZ DE USUARIO.....	37
<b>2.4 MODELO DE DATOS.....</b>	<b>37</b>
2.4.1 GLOSARIO DE ENTIDADES DEL MODELO DE DATOS.....	39
<b>2.5 ESPECIFICACIÓN DE LOS REQUISITOS NO FUNCIONALES .....</b>	<b>41</b>
<b>2.6 CONCLUSIONES PARCIALES.....</b>	<b>44</b>
<b><u>3 CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA .....</u></b>	<b><u>45</u></b>
<b>3.1 INTRODUCCIÓN.....</b>	<b>45</b>
<b>3.2 ARQUITECTURA DE LA SOLUCIÓN .....</b>	<b>45</b>
3.2.1 VISTA LÓGICA DEL MÓDULO DE ADMINISTRACIÓN.....	45
<b>3.3 PATRONES DE DISEÑO.....</b>	<b>48</b>
3.3.1 PATRONES .....	48
3.3.2 PATRONES DE WORKFLOW .....	50
<b>3.4 ESPECIFICACIÓN DE CLASES.....</b>	<b>50</b>
3.4.1 CLASES ENTIDADES .....	51
3.4.2 CLASES CONECTORAS.....	51
3.4.3 CLASES GESTORAS .....	52
3.4.1 DIAGRAMA DE CLASES DEL DISEÑO .....	53
<b>3.5 SERVICIOS DEL SISTEMA.....</b>	<b>53</b>
3.5.1 SERVICIOS DE NEGOCIO .....	54
3.5.2 SERVICIOS DE RUNTIME.....	55
<b>3.6 DISEÑO DEL WORKFLOW.....</b>	<b>56</b>
<b>3.7 MODELO DE DATOS .....</b>	<b>59</b>
3.7.1 DESCRIPCIÓN DE LAS ENTIDADES FUNDAMENTALES .....	61

<b>3.8 CONCLUSIONES PARCIALES.....</b>	<b>62</b>
<b><u>4 CAPTULO IV: IMPLEMENTACIÓN Y PRUEBA .....</u></b>	<b><u>63</u></b>
<b>4.1 INTRODUCCIÓN.....</b>	<b>63</b>
<b>4.2 ESTÁNDARES DE CODIFICACIÓN .....</b>	<b>63</b>
<b>4.3 TRATAMIENTO DE EXCEPCIONES.....</b>	<b>64</b>
<b>4.4 DIAGRAMA DE COMPONENTES .....</b>	<b>65</b>
<b>4.5 DIAGRAMA DE DESPLIEGUE.....</b>	<b>66</b>
<b>4.6 INTERFACES DEL SISTEMA.....</b>	<b>66</b>
<b>4.7 PRUEBAS .....</b>	<b>67</b>
4.7.1 DISEÑO DE CASOS DE PRUEBAS .....	67
4.7.1 PRUEBAS UNITARIAS.....	70
4.7.2 PRUEBAS DE SISTEMA.....	72
<b>4.8 CONCLUSIONES PARCIALES.....</b>	<b>74</b>
<b><u>CONCLUSIONES GENERALES.....</u></b>	<b><u>75</u></b>
<b><u>RECOMENDACIONES.....</u></b>	<b><u>76</u></b>
<b><u>REFERENCIA BIBLIOGRÁFICA.....</u></b>	<b><u>77</u></b>
<b><u>GLOSARIO.....</u></b>	<b><u>79</u></b>

ÍNDICE DE TABLAS

<b>Tabla 1</b>	Roles del Módulo de Administración .....	31
<b>Tabla 2</b>	Descripción del requisito “Crear dirección” .....	36
<b>Tabla 3</b>	Descripción de la entidad “dVivienda” .....	39
<b>Tabla 4</b>	Descripción de la entidad “nLocalidad” .....	39
<b>Tabla 5</b>	Descripción de la entidad “nTipoLocalidad” .....	39
<b>Tabla 6</b>	Descripción de la entidad “nAgrupación” .....	40
<b>Tabla 7</b>	Descripción de la entidad “nTipoAgrupación” .....	40
<b>Tabla 8</b>	Descripción de la entidad “nVía” .....	41
<b>Tabla 9</b>	Descripción de la entidad “nTipoVía” .....	41
<b>Tabla 10</b>	Descripción de la entidad “Address” .....	51
<b>Tabla 11</b>	Descripción de la interfaz “IAddressConnector” .....	51
<b>Tabla 12</b>	Descripción del conector “AddressConnector” .....	52
<b>Tabla 13</b>	Descripción del controlador “AddressController” .....	52
<b>Tabla 14</b>	Descripción de la interfaz del servicio de negocio “IAddressManagerService” .....	54
<b>Tabla 15</b>	Descripción de la clase “AddressManagerService” .....	55
<b>Tabla 16</b>	Descripción de la interfaz de servicio de <i>Runtime</i> “ICreateAddressRServices” .....	55
<b>Tabla 17</b>	Descripción de la clase “CreateAddressRServices” .....	56
<b>Tabla 18</b>	Descripción de la entidad “dDirección” .....	61
<b>Tabla 19</b>	Descripción de la entidad “nReglaLugar” .....	61
<b>Tabla 20</b>	Descripción de la entidad “dInmueble” .....	61
<b>Tabla 21</b>	Diseño de caso de prueba para el RF1 “Crear dirección” .....	70
<b>Tabla 22</b>	Resultados de las pruebas de la primera iteración del RF1 “Crear dirección” .....	73

**ÍNDICE DE FIGURAS**

**Fig. 1** Modelo de Dominio .....27

**Fig. 2** Concepción de los módulos del SUIN .....30

**Fig. 3** Modelo de proceso mejorado “Crear dirección” .....33

**Fig. 4** Prototipo de interfaz de usuario “Crear dirección” .....37

**Fig. 5** Modelo de datos .....38

**Fig. 6** Capas y sus componentes.....46

**Fig. 7** Servicio *Runtime* “Crear dirección” .....49

**Fig. 8** Diagrama de clases del diseño .....53

**Fig. 9** Modelado del Workflow (1) .....58

**Fig. 10** Modelado del Workflow (2) .....59

**Fig. 11** Modelo de datos .....60

**Fig. 12** Diagrama de componentes del Módulo de Administración.....65

**Fig. 13** Diagrama de despliegue del Módulo de Administración .....66

**Fig. 14** Interfaz de usuario “Crear dirección” .....67

**Fig. 15** Prueba Unitaria realizada al servicio “*GetPrimaryTypes*” .....71

**Fig. 16** Resultado de la prueba unitaria al servicio “*GetPrimaryTypes*” .....71

**Fig. 17** Iteraciones de las pruebas de caja negra.....74

## INTRODUCCIÓN

El país desde el triunfo de la Revolución ha trazado estrategias con el objetivo de llevar a cabo el proceso revolucionario, una de ellas es llevar el control estricto de la identificación de la población nacional o extranjera que resida temporal o permanentemente dentro del país, a fin de controlar y garantizar la seguridad ciudadana y el orden interior. Este proceso no se realiza de forma estándar, por lo que no se considera seguro y es difícil de controlar toda la información referente al mismo por parte de los funcionarios.

La necesidad de crear en Cuba un sistema de identidad seguro y su impacto en las actividades de la sociedad fueron enunciados desde inicios del siglo pasado por Fernando Ortiz<sup>2</sup>, momento que se considera como punto de partida para la aparición de diferentes regulaciones relativas a esta actividad en el territorio nacional.

Desde el año 1997 los procesos relativos a la identificación son gestionados por la Dirección de Identificación y Registros (DIR) del Ministerio del Interior (MININT). Durante el último período del 2008 y principios del 2009 se realizó un diagnóstico del estado actual de los sistemas de identificación de la población a nivel nacional e internacional, en función de proponer la modernización de los mismos para garantizar la inscripción e identificación de las personas con procesos y documentos seguros que incluyan el uso de la biometría, a partir de lo cual, se creó la Base de Datos Única de los Ciudadanos de la República de Cuba.

Una vez concluidos estos estudios se realizaron intercambios entre el MININT y la Universidad de las Ciencias Informáticas (UCI) con el objetivo de presentar los resultados obtenidos por esta última en la modernización del Sistema Único de Identificación Nacional (SUIN). Este agrupa un conjunto de módulos fundamentales que están encargados de desarrollar sus propias tareas, pero a la vez tienen procesos que debido a sus necesidades contienen mucha información y responsabilidades que son comunes entre ellos.

Actualmente en las oficinas del Carné de Identidad y Registro de la Población (CIRP) existen diferentes problemas relacionados con la gestión de las direcciones. No se cuenta con reglas definidas para validar

---

<sup>2</sup> Periodista, etnólogo, historiador e investigador de la sociedad cubana, que se dedicó a defender la identidad de todos los cubanos.



la creación de los lugares. No se encuentran definidos los tipos de lugares. Cuando se crea un lugar se invierte el orden de los datos ya que se especifica primeramente el nombre del lugar y luego su tipo, lo que afecta la legibilidad de estos, la búsqueda de un tipo de lugar específico, así como la impresión de la cadena de texto que conforma la dirección, puesto que se hace necesario invertir dicha cadena. Además no se puede evitar la repetición de direcciones ya que existe una codificación desordenada. Los datos que conforman la dirección son gestionados a nivel provincial, esto dificulta la creación de nuevas direcciones ya que no se puede llevar a cabo la búsqueda de elementos contenidos en un municipio. Otro de los problemas que existe es que los inmuebles son gestionados en las oficinas del CIRP y esto crea desorganización y réplica innecesaria de información. Otra de las dificultades es la no existencia de una forma normalizada de manipular los nomencladores<sup>3</sup>. No se controla de manera eficiente cada sesión que está siendo utilizada en las distintas oficinas del CIRP. Además no se cuenta con un mecanismo para el control del trabajo diario en las oficinas que permita evaluar el desempeño de las mismas.

A partir de la situación problemática antes expresada se define como **problema científico** a resolver: *¿cómo informatizar la administración del Sistema Único de Identificación Nacional de la República de Cuba?*

Partiendo del problema planteado se tiene que el **objeto de estudio** se enfocará a los *Procesos de Administración y Configuración en Sistemas de Gestión de Información*.

El **campo de acción** de la investigación lo construyen los *Procesos de Administración y Configuración en el Sistema Único de Identificación Nacional de la República de Cuba*.

Para resolver el problema planteado con anterioridad se propone como **objetivo general**: *Desarrollar un Módulo para la Administración del Sistema Único de Identificación Nacional de la República de Cuba*.

El citado objetivo se desglosó en **objetivos específicos**:

1. Desarrollar el marco teórico de la investigación.
2. Realizar el modelo de dominio para el Módulo de Administración del Sistema Único de Identificación Nacional de la República de Cuba.
3. Especificar los requisitos funcionales y no funcionales del Módulo de Administración del Sistema

---

<sup>3</sup> Lista de objetos definidos.

Único de Identificación Nacional de la República de Cuba.

4. Realizar el diseño del Módulo de Administración del Sistema Único de Identificación Nacional de la República de Cuba.
5. Desarrollar el Módulo de Administración del Sistema Único de Identificación Nacional de la República de Cuba.
6. Validar el Módulo de Administración del Sistema Único de Identificación Nacional de la República de Cuba.

Para dar cumplimiento a los objetivos específicos se proponen las siguientes **tareas de la investigación**:

1. Estudio bibliográfico para la definición de la metodología y herramientas a utilizar.
2. Identificación de las entidades del dominio y sus relaciones.
3. Descripción del modelo de dominio.
4. Identificación de los requisitos del Módulo de Administración del Sistema Único de Identificación Nacional de la República de Cuba.
5. Descripción de los requisitos del Módulo de Administración del Sistema Único de Identificación Nacional de la República de Cuba.
6. Realización de los prototipos de interfaz de usuario.
7. Realización de la validación de los requisitos.
8. Realización del modelo de diseño del Módulo de Administración del Sistema Único de Identificación Nacional de la República de Cuba.
9. Implementación del Módulo de Administración del Sistema Único de Identificación Nacional de la República de Cuba.
10. Aplicación de las pruebas de software al Módulo de Administración del Sistema Único de Identificación Nacional de la República de Cuba.
11. Interpretación de los resultados arrojados en las pruebas aplicadas al Módulo de Administración del Sistema Único de Identificación Nacional de la República de Cuba.

Para dar solución a la interrogante se plantea la siguiente **hipótesis**:

*Con la implementación de un módulo de Administración se logrará la gestión eficiente de los elementos de configuración del Sistema Único de Identificación Nacional de la República de Cuba.*

De la hipótesis antes mencionada se plantean las siguientes **variables de la investigación**:

**1. Variable independiente:**

- ✓ Módulo de Administración del Sistema Único de Identificación Nacional de la República de Cuba.

**2. Variable dependiente:**

- ✓ Gestión eficiente.

Los **Métodos Científicos** utilizados en esta investigación fueron:

➤ **Métodos teóricos:**

1. Análisis histórico-lógico: Permite analizar la trayectoria completa de la gestión de la información que se lleva a cabo en las oficinas del CIRP, así como el estudio histórico del mismo que permite ver deficiencias y proponer soluciones de acuerdo a las necesidades.
2. Analítico sintético: Mediante este método se analiza toda la información recopilada en los medios bibliográficos que pueda servir para desarrollar mejor el diseño del sistema, y aplicar estos conocimientos para tener una mejor preparación en el tema.

➤ **Métodos empíricos:**

1. Observación: Permite visualizar lo que ocurre realmente en las oficinas del CIRP, con el objetivo de clasificar las dificultades existentes y las funcionalidades que debe tener el sistema a implementar.
2. Entrevista: Este método se utiliza para obtener toda la información referente a los problemas de seguridad y administración de la información en las oficinas del CIRP.

**Capítulo 1:** Fundamentación Teórica: Muestra el estudio de aplicaciones que contienen un Módulo de Administración. Se hace referencia a las tendencias y tecnologías actuales que se usaron. Además de la base conceptual y teórica que son el soporte de la investigación.

**Capítulo 2:** Características del sistema: Se hace una propuesta del sistema, para ello se estudiaron los principales procesos vinculados al objeto de estudio y al campo de acción, los cuales se muestran a través de los requerimientos del sistema, los actores que intervienen, sus relaciones y una descripción de cada uno de ellos, teniendo en cuenta los requisitos funcionales y no funcionales que debe cumplir el sistema.

**Capítulo 3:** Análisis y Diseño: Se muestran los flujos de trabajo a seguir y las clases que generará el sistema propuesto, incluyéndose en este, los patrones de diseño, servicios que sustentan el software y las clases que tendrán persistencia en la base de datos.

**Capítulo 4:** Implementación y Prueba: Se muestra la solución del sistema en términos de componentes y su relación, así como los diagramas de implementación y despliegue. Se llevan a cabo las pruebas de caja negra y caja blanca con el objetivo de evaluar las funcionalidades del sistema a desplegar.

### CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

#### 1.1 Introducción

En este capítulo se hace un estudio de aplicaciones web que contienen un Módulo de Administración las cuales manejan conceptos similares a los del SUIN, ejemplo: gestión de roles y usuarios, aunque no presentan algunas características que son fundamentales en el desarrollo del mismo como es la gestión de direcciones. También se exponen las herramientas y tecnologías actuales utilizadas para el desarrollo de este módulo.

#### 1.2 Antecedentes de los procesos de administración de las aplicaciones en las oficinas del CIRP

En las oficinas del CIRP se cuenta con sistemas automatizados para llevar a cabo los procesos de confección y entrega de los documentos de identidad. Uno de estos es el Sistema Automatizado para el Carné de Identidad y Registro de la Población (SACIRP) que tiene un administrador funcional en cada provincia, el cual otorga los permisos sobre el sistema a los funcionarios de cada área en las oficinas del CIRP. Al surgir la necesidad de un sistema que integre todos los procesos llevados a cabo en estas oficinas, se hace indispensable la realización de un Módulo de Administración el cual garantice los elementos de configuración del sistema y la seguridad e integridad de los módulos que lo conformen.

#### 1.3 Conceptos fundamentales en el dominio del problema

##### 1.3.1 Sistema de Gestión de Seguridad de la Información (SGSI)

La información para una organización constituye la base fundamental del negocio de cualquier empresa. Bien por necesidades legales, por competitividad o por las garantías que se ofrecen a los clientes, la seguridad de los datos manejados debe constituir una prioridad. Los sistemas de seguridad de la información contemplan una serie de mecanismos y controles de seguridad que deben ser aplicados a todos los niveles. La aplicación de un sistema de administración de la seguridad de la información debe contener:

- El alcance de la administración de seguridad en el negocio.
- La evaluación y análisis de riesgos de la organización.
- Parametrización y métrica de los riesgos.
- Evaluación y respuesta frente a los posibles riesgos existentes.
- Definición de controles para el análisis de los riesgos.

Un **SGSI** consiste en el diseño, implantación, mantenimiento de un conjunto de procesos para gestionar eficientemente la accesibilidad de la información, buscando asegurar la confidencialidad, integridad y disponibilidad. Todo ello minimizando los riesgos de la seguridad.

La política de seguridad de los SGSI debe incluir los siguientes elementos:

- Documentación de procedimientos operativos.
- Control de cambios operacionales.
- Procedimientos de gestión de incidencias.
- Segregación de tareas. (1)

### **1.3.2 Administración de sistemas informáticos**

En un sistema informático con funcionalidades de gestión de la información de cualquier tipo existe algo bien similar llamado módulo administrativo, que tendrá varias operaciones fundamentales:

- Controlar las operaciones como parte de la seguridad.
- Definir las estructuras lógicas para tener una buena organización en el sistema.
- Gestionar las configuraciones globales o específicas de los negocios que integra.

#### **¿Qué tipo de funcionalidades debe realizar un módulo de administración?**

Por lo general diferentes tipos de aplicaciones necesitan la gestión o configuración de los usuarios y de las oficinas donde operan sus funcionarios, la gestión de proyectos, definición de servidores y bases de datos a acceder, definición de dominios, gestión de sesiones activas y estadísticas para balanceo de carga y mejoramiento del tiempo de respuesta, buscador de objetos, herramientas de colaboración entre desarrolladores, control de acceso restringido para cada proyecto, y otras funcionalidades que son comunes para cualquier sistema. Partiendo de estos conceptos se dice que el módulo administrativo de un sistema informático se encarga de llevar a cabo las funcionalidades antes mencionadas con el objetivo de perfeccionar todo el funcionamiento del mismo. (2)

Como se explica anteriormente uno de los principales elementos que se manejan en la administración de un sistema es el término usuario, así como la asignación de roles y permisos a estos para la realización de las tareas en una aplicación determinada, garantizando de esta manera la seguridad de la información.

Un **usuario** es aquella persona que utiliza un dispositivo o un ordenador y realiza múltiples operaciones con distintos propósitos. A menudo es un usuario aquel que adquiere una computadora o dispositivo electrónico y que lo emplea para comunicarse con otros usuarios, generar contenido y documentos, utilizar software de diversos tipos y muchas otras acciones posibles. (3)

El concepto de usuario es utilizado generalmente en el campo de la tecnología informática para referirse a quien utiliza determinado hardware y/o software, mediante el cual obtiene un servicio. Las cuentas de usuario son creadas para poder brindar acceso a diferentes personas con respecto a un mismo sistema informático. Generalmente un usuario posee su propia carpeta personal, permisos para ejecutar determinados programas y acceso limitado a determinada red informática (interna o externa) con todo lo que esto supone. (4)

La **gestión de los usuarios** es un área de entidad propia, en la medida en que la relación de estos con los sistemas es crucial a la hora de asegurar un correcto funcionamiento de los servicios puestos a su disposición. (5)

Las cuentas de usuario y de grupos permiten que los usuarios participen en un dominio y tengan acceso a los recursos de este, en función de los derechos y permisos que tengan asignados. Es muy importante planificar cuidadosamente la administración de las cuentas de usuario y grupos. (6)

Una parte importante de cualquier sistema de gestión es la seguridad de la información que se maneja, hay que garantizar esa seguridad a partir de la concesión de permisos para controlar qué usuarios deben tener acceso a parte o a toda la información y quiénes no. Se puede asignar permisos solo para acceder a la información o permisos para acceder a la información y modificarla. (7)

A los usuarios del sistema se le asignan roles con el objetivo de distribuir las actividades en las aplicaciones, un **rol** es una manera de agrupar usuarios acordando a qué partes de la aplicación tienen acceso. (8)

La **gestión de roles** permite, por lo tanto, disponer de una visión conjunta y la identificación de los roles técnicos ya existentes en los sistemas. Esto puede facilitar, de gran manera, la identificación de roles funcionales o anomalías en las asignaciones, pero también facilitar la operativa de la gestión de usuarios, dado que siempre es más fácil asignar a un nuevo empleado los privilegios ya existentes para un

compañero de departamento, que ir asignándolos poco a poco a medida que los vaya pidiendo. Por otra parte, si ya se dispone de herramientas de gestión de usuarios (*Provisioning*<sup>4</sup>), la gestión de roles es un complemento muy importante para perfilar usuarios, modernizar los roles disponibles y conservar su limpieza y eficacia. (9)

Muchos sistemas manejan el concepto trámite para dirigirse a procesos que tienen un inicio y un fin con el objetivo de lograr un resultado, en este tipo de sistemas el Módulo de Administración tiene la responsabilidad de gestionarlos para asignarlos a los usuarios del sistema, con el propósito de llevar el control de los mismos. Un **trámite** es el estado de un proceso administrativo por el que tiene que pasar un asunto para ser solucionado. En este caso se denomina trámite a la confección del documento de identidad. (10)

### **Seguridad del sistema**

La seguridad de la aplicación es uno de los requerimientos más importantes que deben ser cumplidos. Está basada en las restricciones más comúnmente usadas en cualquier sistema que la implemente, con el uso de los conceptos de autenticación<sup>5</sup> y autorización<sup>6</sup>, basados en requerir una cuenta de usuario válida y activa, además de un conjunto de roles que le permitirán al usuario realizar solo aquellas operaciones que se le han asignado con el uso de este Módulo de Administración. Todo este chequeo es realizado, al iniciar la aplicación y luego comprueba la autorización a las distintas funcionalidades del sistema.

De esta forma el usuario solo podrá hacer lo que se le asigne y permitirá aumentar la fiabilidad del sistema. Al ser la seguridad un requisito muy importante para este, en las primeras fases de análisis y diseño del proyecto se deben definir una serie de restricciones para el acceso a la base de datos y para el registro de las operaciones de los usuarios que interactúan con ella. Existen tantos usuarios como sean necesarios, a los cuales se les asignan roles lógicos también registrados los cuales tienen correspondencia con los roles físicos de la base de datos, y son a estos roles a los que se les otorga el permiso para la ejecución de los procedimientos almacenados que permiten el intercambio de datos. (2)

---

<sup>4</sup> Herramienta para gestionar roles, usuarios y recursos de un sistema.

<sup>5</sup> Acción del usuario que le permite identificarse unívocamente en un sistema informático.

<sup>6</sup> Es cuando se le otorgan permisos a un usuario para realizar ciertas operaciones en un sistema informático.



La **administración de sesiones** es otro elemento fundamental en la administración de sistemas, en este caso se abordará sobre este tema relacionado con las aplicaciones web.

Muchos de los programas que se utilizan en la actualidad requieren autenticar a los usuarios antes de permitirles el acceso a información sensible; tradicionalmente, la forma de reconocer los privilegios de acceso ha sido mediante el establecimiento de sesiones de usuario. Saber quién está haciendo qué en cada momento, es uno de los principios de la seguridad informática moderna; es por eso que muchas de las aplicaciones que usamos regularmente requieren la autenticación de sus usuarios antes de permitirles el acceso a cierta información o a cierto conjunto de transacciones. La forma habitual de llevar a cabo esta tarea de certificación del usuario es el establecimiento de una “sesión” la cual normalmente se mantiene vigente desde que el usuario presenta sus credenciales (nombre de usuario, contraseña, certificado digital, etc.) hasta que este cierra el enlace o la actividad cesa por un período de tiempo prolongado. (11)

### **Gestión por procesos**

Actualmente el desarrollo de software se orienta a procesos con el objetivo de permitir diseñarlos de acuerdo con el entorno y los recursos disponibles, normalizando la actualización y la transferencia de información de todas las personas que participan en los mismos, garantizando eficiencia, efectividad y calidad de los servicios. Un **proceso** es el conjunto de actividades que se encadenan de forma secuencial y ordenada para conseguir un resultado que satisfaga plenamente los requerimientos del cliente al que va dirigido”. La **gestión por procesos** busca reducir la variabilidad innecesaria que aparece habitualmente cuando se producen o prestan determinados servicios y trata de eliminar las ineficiencias asociadas a la repetitividad de las acciones o actividades, al consumo inapropiado de recursos, etc. (12)

Los sistemas de administración también se encargan de la configuración de servidores, con el objetivo de poder gestionar sus datos identificativos y asignarle las unidades que utilizarán sus servicios, en este caso se hace necesario asignarle a los servidores las oficinas del CIRP.

Un **servidor** es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos. Un servidor brinda información a los ordenadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor. (13)

Un **servidor de aplicaciones** es una tecnología básica que proporciona la infraestructura y servicios claves a las aplicaciones alojadas en un sistema. Entre los servicios habituales de un servidor de aplicaciones se incluyen los siguientes:

- Agrupación de recursos (por ejemplo: agrupación de conexiones de base de datos y agrupación de objetos).
- Administración de transacciones distribuidas.
- Comunicación asincrónica de programas normalmente a través de colas de mensajes.
- Un modelo de activación de objetos oportuno.
- Interfaces de servicios Web XML automáticas para tener acceso a objetos de empresas.
- Servicios de detección de errores y estado de las aplicaciones.
- Seguridad integrada. (13)

### 1.3.3 Gestión de direcciones

La gestión de direcciones es otro punto fundamental que puede ser tratado como parte de un sistema de administración, donde se defina un estándar para la confección de la misma y se lleve el control de los elementos que la conforman.

Se entiende por **dirección** cualquier tipo de información descriptiva que distinga a un lugar del resto. La dirección es un sencillo y fácil método para describir una ubicación. A diferencia de las coordenadas, una dirección describe como referencia una ubicación basada en elementos existentes en una *geodatabase*<sup>7</sup>. En la mayoría de los casos una dirección es fácilmente interpretable y entendible. (14)

La **geocodificación** es el proceso de asignar una ubicación, generalmente en coordenadas, a una dirección mediante la comparación de elementos distintivos de esta con otros presentes en direcciones de referencia. (14)

Uno de los conceptos tratados en la gestión de direcciones es **asentamiento humano** que es utilizado para nombrar los distintos grupos de poblaciones, a continuación se explica en qué consisten algunos de ellos.

---

<sup>7</sup> Terminología en ingles que se utiliza para hacer referencia a una base de datos geográfica.

Un **asentamiento humano o poblacional** se considera a toda manifestación o presencia de personas con residencia fija en un lugar determinado, a partir del cual desarrollan sus actividades vitales. Constituye la expresión física del poblamiento y puede ser según el nivel de concentración de las viviendas de dos tipos: concentrado y disperso.

Un **asentamiento humano o concentrado** se considera a una agrupación de 15 o más viviendas habitadas o no, de forma permanente, separadas entre sí no más de 50 metros, con un nombre que la identifique y linderos determinados que la diferencien de otra, que puede tener o no dentro de su estructura, instalaciones de servicio, producción, etc.

Un **asentamiento humano o población rural** se considera a los lugares con población residente de forma permanente que no se clasifican como urbanos, los que en función del número y distancia que separa a las viviendas que lo componen, pueden ser concentrados o dispersos.

Un **asentamiento humano o población disperso** lo constituyen todas las viviendas habitadas o agrupaciones de ellas, que no estuvieron incluidas dentro de las consideraciones anteriores. (15)

Actualmente en las oficinas del CIRP se cuenta con sistemas que registran las direcciones de las personas con sus datos básicos. A continuación se describen dos de ellos.

### **Base de Datos Nacional del CIRP**

El desarrollo de la Base de Datos Nacional del CIRP, implantada en abril de 2006, permitió obtener una base de consulta obligada para todos los municipios del país, con un almacenamiento centralizado en la DIR y un mecanismo de actualización propio, diario y confiable. Esta base es utilizada para obtener los datos identificativos de un sitio único y sin necesidad de re teclear datos, en los que pudiera incurrirse en errores mecanográficos. Utilizando esta misma base de datos funciona hoy un Servicio Web, permitiendo implementar dentro de cualquier aplicación Web la obtención directa de los datos identificativos de cualquier persona incluida en la Base de Datos Nacional del CIRP, que almacena los datos identificativos básicos y la última dirección de cada persona. (16)

### **Base de Datos Nacional del SACIRP**

Como consecuencia de la creación del SACIRP, implantado en todo el país en noviembre de 2007, tres meses después se implementó la Base de Datos Nacional del SACIRP que supera a la del CIRP. La base

precedente almacena los datos identificativos básicos (número de identidad, nombres, apellidos, padres y la última dirección de residencia de la persona), mientras que la Base de Datos Nacional del SACIRP añade a las personas nuevos campos, incorpora las fotos y considera todas las direcciones de cada persona. Se cuenta hoy con más de 15,4 millones de personas, más de 8,5 millones de fotos y más de 18,9 millones de direcciones. Además, esta implementación tiene la ventaja de contar con un mecanismo propio de actualización instantáneo, con lo cual se tiene una copia fiel de las bases provinciales con diferencias de apenas minutos de desactualización. (17)

### **Oracle Spatial y Oracle Locator**

La mayoría de las informaciones sobre negocios tienen un componente de localización, como direcciones de clientes, ventas en un determinado territorio o límites de los servicios inalámbricos. Los negocios pueden beneficiarse de su información geográfica mediante la incorporación de un análisis de localización en sus sistemas de información. Esto permite a las organizaciones tomar mejores decisiones y responder a los clientes más eficientemente.

#### **Ventajas de Oracle Locator:**

- Consultas de cercanía y otras relaciones espaciales entre geometrías.
- Ejecuta consultas de localización sobre información relacional no almacenada en el tipo de geometría Oracle Spatial.
- Soporte para transacciones largas.
- Almacena e indexa geometrías de vectores en la base de datos.
- Optimiza el rendimiento de las aplicaciones.

#### **Ventajas de Oracle Spatial:**

- Ejecuta cálculos de longitud y área en geometrías.
- Genera nuevas geometrías tales como buffers<sup>8</sup> y uniones.
- Lleva a cabo transformaciones de sistemas de coordenadas, para geometrías individuales o capas completas.
- Almacena información de medida lineal. (18)

---

<sup>8</sup> Espacio de memoria.

### 1.3.4 Gestión de la información

Todo sistema de gestión de información como su nombre lo indica maneja documentación importante que en cierto momento es necesaria obtenerla para ser utilizada por otras personas o sistemas, por lo que es preciso que cada sistema genere un resumen de la información contenida así como el historial de los procesos que se llevan a cabo internamente en este, para ello es necesario que dicha información se pueda obtener de un período determinado, así como el total de dichos procesos, a este resumen se le denomina reporte.

Un **reporte** es un resumen generado por un sistema, que presenta de manera estructurada y/o resumida datos relevantes guardados o generados por la misma aplicación de tal manera que se vuelvan útiles para los fines que se tengan. A diferencia de un formulario, los datos dentro de un reporte no pueden ser manipulados o modificados directamente, sino que tienen que ser modificados en alguna otra parte del sistema para que se reflejen los cambios una vez que el reporte sea generado nuevamente. Un reporte es generado dinámicamente, es decir, cada vez que se realiza una llamada o se invoca desde un sistema, el reporte actualiza la información a los datos más recientes que están disponibles. (19)

### 1.4 Análisis de soluciones existentes

En la actualidad gran parte de los sistemas informáticos cuentan con un Módulo de Administración con el objetivo de gestionar sus elementos fundamentales, así como el acceso y seguridad a la información contenida en estos. A continuación se analizarán algunas de estas aplicaciones existentes en Cuba y el mundo.

#### **Sistema Automatizado para el Carné de Identidad y Registro de la Población (SACIRP)**

Sistema que se utiliza en las oficinas del CIRP, realizado con el fin de facilitar la actualización de la información, el cual trabaja sobre la Base de Datos Provincial, permitiendo hacer búsquedas de personas en la Base de Datos Nacional e importar los datos de las mismas a la Base de Datos Provincial. Este sistema cuenta con un Módulo de Administración donde son gestionadas las direcciones de las personas por un administrador que existe en cada Sección Provincial perteneciente al Órgano de la Jefatura del MININT de cada provincia, excepto en Ciudad de la Habana, donde estos sistemas son manejados por la DIR. Cada administrador provincial se encarga de otorgar los permisos de administración a los funcionarios de cada área de trabajo para que realicen las distintas operaciones sobre este sistema, con el

fin de llevar a cabo cada trámite. Con esta jerarquía administrativa se logra el control y seguridad de los procesos de identificación de las personas. Una de las modificaciones hechas al CIRP es la seguridad informática a nivel de usuario, a partir de que:

- Individualiza el trabajo según el rol asignado al usuario, lo que permite conocer las acciones realizadas por el mismo.
- Crea trazas de las acciones que se efectúan en la aplicación permitiendo la recuperación de la información sin asignar un nuevo número de sistema. (20)

### **Sistema Automatizado de Atención a la Población (SAAP)**

Es un sistema informático realizado con el objetivo de informatizar los procesos de trabajo en las oficinas del CIRP con el fin de lograr un servicio de excelencia para el pueblo. Desde el punto de vista de seguridad informática el sistema garantiza la autenticación de todos sus usuarios, con administración provincial y municipal según los permisos y roles establecido. Además brinda información estadística que permite ejecutar análisis y valoraciones tanto desde el punto de vista de los indicadores del desempeño de la oficina, como individual por cada funcionario que participa en los flujos de trabajo. El sistema está diseñado para automatizar los procesos de trabajo desde la información que se le brinda a la ciudadanía, hasta la entrega del documento de identidad o satisfacer la demanda del servicio que la misma solicite, transitando en el intermedio por la captación de la información y la realización del trámite solicitado. Para ello tiene definido diferentes roles para ejecutar el trabajo en cada área o ejecutar la fiscalización de todo el proceso a diferentes niveles según la distribución de los mismos a los funcionarios, primeros funcionarios y jefes según corresponda en la cadena de mando. A continuación se muestra como se lleva a cabo la asignación de roles y permisos:

- **Administración Provincial:** Asigna los usuarios por municipio, ejecuta el mantenimiento al sistema y a su base de datos, así como tiene acceso a todos los roles del sistema.
- **Administración Municipal:** Está habilitada para el Jefe y 2do Jefe de Unidad o Funcionario responsable; accede a toda la información que brinda el sistema y asigna el rol a los usuarios en su municipio, según la distribución del trabajo que defina diariamente.
- **Recepción:** Brinda información a la ciudadanía por solicitud, valida los requisitos según el tipo de trámite, inserta los datos identificativos del ciudadano en el sistema y garantiza el flujo continuo de los mismos hacia el funcionario tramitador o área de tramitación.

- **Archivo:** Visualiza según el número de orden asignado por el sistema los datos identificativos de los ciudadanos para realizar la búsqueda del expediente en los casos que corresponda y marcando su resultado para todos los casos. Además ejecuta la búsqueda en los registros operativos y según una respuesta refleja los antecedentes policiales de interés para los sistemas de enfrentamiento.
- **Trámite:** Visualiza y llama al ciudadano por su número de orden a partir de que esté listo para tramitar, marcando esta acción, chequea que se cumplan los requisitos para el trámite según corresponda y lo ejecuta directamente en el SACIRP, cuando termina tiene que marcar su conclusión.
- **Visualizar:** Está disponible para todos los usuarios que interactúan con el sistema; visualizando el comportamiento de los procesos de trabajo del municipio seleccionado, convirtiéndose en una herramienta eficaz de gestión. (21)

### **Sistema Autónomo de Identificación, Migración y Extranjería (SAIME)**

Es un sistema realizado para la reestructuración, modernización y automatización de la Oficina Nacional de Identificación y Extranjería de Venezuela (ONIDEX). Este sistema cuenta con un Módulo de Administración que tiene como objetivos: el control de las oficinas y los funcionarios que operan con el sistema SAIME además de un conjunto de configuraciones necesarias para el control centralizado y la eficiencia, seguridad e integridad de los procesos que se llevan a cabo. Este sistema brinda las siguientes funcionalidades:

- Configuración de oficinas móviles.
- Configuración de usuarios de oficinas.
- Configuración de usuarios.
- Configuración de sesiones y puestos de trabajo.
- Configuración de plantillas de roles para los tipos de oficinas.
- Configuración de servidores.
- Generación de actualizaciones.
- Autenticación de los usuarios en el sistema.
- Gestión de usuarios por oficinas. (22)

### 1.5 ¿Por qué la necesidad de un Módulo de Administración para el SUIN?

La DIR como entidad encargada de la identificación de la población cubana necesita de un sistema completamente seguro en la gestión de los procesos de la identidad nacional por lo que se hace indispensable la creación de un sistema capaz de llevar a cabo todos los procesos antes mencionados, garantizando una completa seguridad de la información. A partir de lo antes mencionado surge la propuesta del Módulo de Administración del SUIN el cual recoge las funcionalidades principales de administración y seguridad de un sistema así como aquellas que son indispensables para llevar a cabo cada uno de los procesos en dicho sistema. El Módulo de Administración del SUIN llevará a cabo la gestión de los elementos fundamentales en la realización de cada uno de los trámites que se realicen, con el fin de garantizar el control de toda la información referente a los mismos.

### 1.6 Ambiente de desarrollo

Para llevar a cabo un proceso de desarrollo de software se hace necesario el uso de metodologías de desarrollo, herramientas para el modelado de los procesos y la implementación de la aplicación. A continuación se describirán las herramientas que se decidieron utilizar para el desarrollo del sistema propuesto, abordando las funcionalidades y ventajas que estas presentan.

#### 1.6.1 Metodología de desarrollo

Una metodología debe definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto y guías para uso de herramientas de apoyo. Para poder desarrollar un producto de software que cumpla con la calidad requerida se hace necesario utilizar una metodología que guíe el trabajo del equipo de desarrollo, indicando los procesos que se llevarán a cabo a medida de que vaya evolucionando el sistema. El uso de metodologías permite planificar y controlar el desarrollo del sistema desde sus inicios hasta obtener el resultado esperado.

#### **MSF<sup>9</sup> for CMMI<sup>10</sup> Process Improvement**

MSF for CMMI se centra en el proceso, la comprensión de la variación, y la gestión a través de la reducción de la variación, a través del uso de la garantía de calidad, riesgo, gestión de problemas y

---

<sup>9</sup> Microsoft Solution Framework.

<sup>10</sup> Capability Maturity Model Integration.



técnicas de ingeniería mejorada introducida en condiciones controladas. Esto fomenta la conducta funcional mediante un mejor trabajo en equipo, intercambio de conocimientos y sugerencias de mejora. Unido al *Team Model*<sup>11</sup> representan el ciclo completo de vida del proyecto, incluida la visión, la producción, uso y mantenimiento. Cada miembro del equipo tiene al menos una de las funciones y es responsable de promover su rol en la realización de las actividades para desarrollar el proyecto. El trabajo orientado hacia el trabajo en equipo y el consenso de los juegos de rol facilita los controles y contrapesos necesarios para garantizar la calidad global y clientes satisfechos en el marco de desarrollo.

La integración de MSF con el Team System ofrece soporte para un desarrollo rápido e interactivo que además permite aprendizaje continuo y refinamiento. Los ciclos e iteraciones permiten la disminución de los errores en la aplicación y la mejora continua de los procesos. MSF for CMMI Process Improvement define cinco fases para el ciclo de vida del proyecto las cuales encapsulan actividades y *workstream*<sup>12</sup>. Cada fase comienza cuando aún la anterior no ha llegado a su fin, permitiendo así, la continuidad o no del trabajo para lograr el avance del proceso de desarrollo. (23)

### 1.6.2 Modelado de Procesos de Negocio

Después de haber definido la metodología de desarrollo es necesario modelar los procesos de negocio con el objetivo de especificar y documentar las funcionalidades del sistema.

#### Altova UModel 2009 para el modelado de procesos

Altova UModel 2009 es una herramienta de modelado para diseñar visualmente modelos de aplicaciones en UML<sup>13</sup>. Genera código Java, C#, o Visual Basic .NET y documentación del proyecto. Esta herramienta combina la interfaz visual con funciones de usabilidad superiores para ayudar a nivelar la curva de aprendizaje de UML. A continuación se muestran sus principales características con respecto al desarrollo de software basado en las capacidades de modelado avanzado:

- Soporte para los catorce tipos de diagramas UML.
- Modelado de esquemas XML en diagramas UML.
- Diagramas de proceso de negocio (BPMN).
- Generación de código fuente en lenguajes Java, C#, y Visual Basic .NET.

---

<sup>11</sup> Modelo de Equipo.

<sup>12</sup> Secuencia de actividades de trabajo.

<sup>13</sup> Lenguaje Unificado de Modelado (Unified Modeling Language).

- Ingeniería inversa de código fuente y ficheros binarios Java, C# y VB.NET.
- Sincronizado de modelo y código a través de ingeniería de ida y vuelta.
- Crea diagramas de secuencia desde el código fuente de la ingeniería inversa.
- Generación de documentación personalizable de proyecto.
- Compartir subproyectos para colaboración o reutilización.
- Capas de diagramas con visibilidad selectiva.
- Hipervínculos ente diagramas, documentos, o páginas Web.
- Soporte para intercambio de modelos XMI 2.1.
- Integración con sistemas de control de versiones.
- API<sup>14</sup> extendida para permitir manipulaciones externas.
- Estrecha integración con Visual Studio y Eclipse. (24)

### **Business Process Modeling Notation (BPMN)**

BPMN es un estándar de modelado de procesos de negocio, que ofrece la representación gráfica de las diferentes etapas del proceso del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes. BPMN tiene un enfoque en procesos de negocio, UML se enfoca al diseño de software y por lo tanto ambas notaciones son totalmente compatibles entre sí. Esta notación fue utilizada para el modelado de los procesos llevados a cabo en el Módulo de Administración del SUIN. (25)

### **1.6.3 Sistema Gestor de Base de Datos (SGBD)**

Los sistemas de Gestión de Bases de Datos son aplicaciones que permiten a los usuarios definir, crear y mantener la base de datos y proporciona un acceso controlado a la misma. Un SGBD es la aplicación que interactúa con los usuarios de los programas de aplicación y la base de datos. Algunos de los SGBD más conocidos son: SQL, DB2, SLQ/DS, ORACLE, INGRES, INFORMIX, SYBASE, PARADOX, DBASE, ACCESS, FOXPRO, R, RM/T y RM/V2.

#### **1.6.3.1 Oracle 11g como Sistema Gestor de Bases de Datos**

Oracle Database 11g proporciona nuevas e innovadoras funcionalidades que garantizan alto rendimiento, alta escalabilidad, fiabilidad y seguridad mediante el uso de plataformas *grid*<sup>15</sup>, asegurando altos niveles

---

<sup>14</sup> Interfaz de programación de aplicaciones.

de calidad de servicio e incrementos de la flexibilidad de negocio reduciendo además los costes de explotación. Es la primera base de datos del 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. A continuación se muestran algunas de las características de esta herramienta que muestran sus ventajas:

### **Características de Oracle 11g:**

- Gestión eficiente y segura de todo tipo de datos
- Arquitectura de máxima disponibilidad
- Gestión de datos XML
- Mejoras en los pool de conexiones y en los caches de resultados
- Mejoras en el desarrollo de aplicaciones
- Mejoras en la automatización y en la auto-administración de la base de datos (26)

### **1.6.4 Herramientas para el entorno de desarrollo**

Para la realización de aplicaciones cada vez mejores y con gran calidad se deben utilizar herramientas que permitan implementar procesos coherentes en el mejor tiempo posible. A continuación se hace un estudio de las herramientas que se utilizarán en el desarrollo de la aplicación propuesta.

#### **1.6.4.1 Visual Studio Team System 2008 como herramienta de desarrollo**

Microsoft Visual Studio Team System es un sistema integrado de gestión de ciclo de vida de aplicaciones, solución basada en herramientas, procesos y orientación. El uso de esta herramienta permite:

- Colaborar y comunicarse eficazmente con otros miembros del equipo y partes interesadas del negocio, como pueden ser los clientes.
- Garantizar la calidad en todas las fases del proceso de desarrollo.
- Mejorar la visibilidad del estado del proyecto y la calidad, y utilizar esos datos para tomar decisiones en tiempo real. (27)

Ofrece una amplia gama de herramientas para todas las fases del desarrollo de software, incluidos la creación, la prueba, la implementación, la integración y la administración. Estas herramientas son:

---

<sup>15</sup> Tecnología que permite que los ordenadores compartan información, capacidad de procesamiento, aplicaciones y dispositivos totalmente heterogéneos.

- Sistema de proyectos para administrar los datos requeridos para el desarrollo de aplicaciones de su proyecto.
- Herramientas de edición de código para escribir y modificar texto y código.
- Herramientas de refactorización y depuración para mejorar el código e identificar y resolver errores lógicos.
- Herramientas de generación de informes para ayudarlo a llevar registros y para tareas de administración.
- Herramientas avanzadas para diseñar, implementar, analizar, probar y evaluar el progreso del desarrollo de aplicaciones.
- Opciones de lenguajes que incluyen JScript 8.0, Visual Basic 2008, Visual C# 2008 y Visual C++ 2008.
- Herramientas de plataformas para la escritura de aplicaciones para las tecnologías de Office, Windows CE<sup>16</sup>, .NET y Windows.(28)

### 1.6.4.2 Plataforma de desarrollo .NET

.NET Framework es la plataforma de desarrollo de código administrado de Microsoft. Está formado por una serie de herramientas y librerías con las que se pueden crear todo tipo de aplicaciones, desde las tradicionales aplicaciones de escritorio hasta aplicaciones para XBOX<sup>17</sup> (XNA<sup>18</sup>) pasando por desarrollo web (ASP.NET), desarrollo para móviles (Compact Framework), aplicaciones de servidor (WPF<sup>19</sup>, WCF<sup>20</sup>), etcétera. (29)

### .NET Framework versión 3.5

.NET Framework 3.5 agrega de forma incremental las nuevas características de .NET Framework 3.0. Por ejemplo, los conjuntos de características de WF<sup>21</sup>, WCF, WPF y Windows CardSpace. Además .NET Framework 3.5 contiene una serie de características nuevas en distintas áreas tecnológicas que se han

---

<sup>16</sup> Sistema operativo para dispositivos móviles de 32-bits inteligentes y conectados.

<sup>17</sup> Primera videoconsola de sobremesa producida por Microsoft en colaboración con Intel y Nvidia.

<sup>18</sup> Conjunto de herramientas con un entorno de ejecución administrado que facilita el desarrollo de juegos de ordenador y de gestión.

<sup>19</sup> Windows Presentation Foundation.

<sup>20</sup> Windows Communication Foundation.

<sup>21</sup> Windows Workflow Foundation.

agregado como nuevos ensamblados para evitar cambios destacados. Algunas de estas características son:

- Integración total de LINQ<sup>22</sup> y del reconocimiento de los datos. Esta nueva característica le permitirá escribir código en idiomas habilitados para LINQ para filtrar, enumerar y crear proyecciones de varios tipos de datos SQL, colecciones, XML y conjuntos de datos usando la misma sintaxis.
- ASP.NET AJAX le permite crear experiencias web más eficaces, más interactivas y con un gran índice de personalización que funcionan con los exploradores más usados.
- Nueva compatibilidad con el protocolo web para generar servicios WCF, como por ejemplo AJAX, JSON, REST, POX, RSS, ATOM y distintos estándares WS-\* nuevos.
- Compatibilidad total con las herramientas de Visual Studio 2008 para WF, WCF y WPF, incluida la nueva tecnología de servicios habilitados para flujos de trabajo. Nuevas clases en la biblioteca de clases base (BCL<sup>23</sup>) de .NET Framework 3.5 que tratan numerosas solicitudes de cliente comunes.

.NET Compact Framework versión 3.5 amplía la compatibilidad con aplicaciones móviles distribuidas al incorporar la tecnología WCF. También agrega nuevas características de lenguaje como LINQ, incluye nuevas API basadas en los comentarios de la comunidad y mejora la depuración con herramientas y características de diagnóstico actualizadas. (30)

### **Servicios duraderos**

.NET Framework 3.5 también incorpora la compatibilidad con los servicios de WCF que utilizan el modelo de persistencia de WF para almacenar la información de estado del servicio. Estos servicios duraderos conservan la información de estado en el nivel de la aplicación, de modo que si una sesión se interrumpe y se reanuda de nuevo más tarde, la información de estado de ese servicio se puede volver a cargar desde el almacén de persistencia. (31)

#### **1.6.4.3 Windows Workflow Foundation (WF)**

Windows Workflow Foundation es una plataforma que permite a los usuarios crear un flujo de trabajo en sus aplicaciones. Es básicamente una colección compuesta por un motor de ejecución, un motor de reglas, una serie de actividades, una serie de servicios de apoyo en tiempo de ejecución y un diseñador

---

<sup>22</sup> Language Integrated Query.

<sup>23</sup> Base Class Library.

que permite a los desarrolladores diseñar sus flujos de trabajo gráficos para Visual Studio, y puede ser utilizado en escenarios simples, como sería mostrar controles de interfaz de usuario, basados en entradas de usuarios, o escenarios complejos. WF 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 reutilizarlos a través de múltiples proyectos. Uno de los aspectos más interesantes de WF es la separación entre el código de procesos de negocio y la ejecución real, a menudo un requisito en la Arquitectura Orientada a Servicios (SOA). (32)

#### 1.6.4.4 Lenguaje de programación C#

Es un lenguaje orientado a objetos, diseñado para ser utilizado en la plataforma .NET.

##### Características:

- **Sencillez:** C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo:
  - El código escrito en C# es autocontenido, lo que significa que no necesita de ficheros adicionales al propio.
  - El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), lo que facilita la portabilidad del código.
  - No se incluyen elementos poco útiles de lenguajes como C++ tales como macros, herencia múltiple, la necesidad de un operador diferente del punto (.) o acceder a miembros de espacios de nombres (::).
- **Orientado a objetos:** No admite ni funciones ni variables globales sino que todo el código y datos han de especificarse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.
- **Eficiente:** En principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros. Para ello basta marcar regiones de código como inseguras y podrán usarse en ellas punteros de forma similar a como se hace en C++, lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad de procesamiento muy grandes.

- **Orientado a componentes:** La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# 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). (33)

### 1.6.4.5 Capa de Acceso a Dato “Entity Framework”

Está diseñado para permitir a los programadores crear aplicaciones de acceso a datos programando con un modelo de la aplicación conceptual en lugar de programar directamente con un esquema de almacenamiento relacional. El objetivo es reducir la cantidad de código y mantenimiento que se necesita para las aplicaciones orientadas a datos. Las aplicaciones de Entity Framework ofrecen las siguientes ventajas:

- Las aplicaciones pueden funcionar en términos de un modelo conceptual más centrado en la aplicación, que incluye tipos con herencia, miembros complejos y relaciones.
- Las aplicaciones están libres de dependencias de codificación rígida de un motor de datos o de un esquema de almacenamiento.
- 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 Language Integrated Query proporciona validación de la sintaxis en el momento de la compilación para consultas en un modelo conceptual.

Entity Framework 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. (34)

### 1.6.4.6 LINQ: lenguaje para enlazar el mundo de los datos con el mundo de los lenguajes de programación

Language Integrated Query (LINQ) es un conjunto de características en Visual Studio 2008 que agrega eficaces capacidades de consulta a la sintaxis de los lenguajes C# y Visual Basic. LINQ incluye patrones estándar y de fácil aprendizaje para consultar y actualizar datos, y su tecnología se puede extender para utilizar potencialmente cualquier tipo de almacén de datos. Visual Studio 2008 incluye ensamblados de proveedores para LINQ que habilitan el uso de LINQ con colecciones de .NET Framework, bases de datos de SQL Server, conjuntos de datos de ADO.NET y documentos XML. 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. LINQ para SQL<sup>24</sup> es una implementación de O/RM<sup>25</sup> que viene con la nueva versión del .NET Framework y permite modelar bases de datos relacionales con clases de .NET. Se puede consultar bases de datos así como actualizar, añadir y borrar datos de ellas. (35)

## 1.7 Conclusiones parciales

- En este capítulo se realizó el estudio de aplicaciones en el ámbito nacional e internacional, las cuales contienen un Módulo de Administración para lograr el control de la estructura organizativa de las mismas.
- Después del análisis de otras soluciones se realizó el estudio de las funcionalidades que cumplían con el objetivo de dar solución al proceso de administración del SUIN, se vio la necesidad de la implementación de un Módulo de Administración ya que estas soluciones no satisfacían todas las necesidades del cliente.
- Se hizo un estudio de los principales conceptos relacionados con la administración de los sistemas y su relación con los elementos que serán tratados en la solución del Módulo de Administración del SUIN.
- Se llevó a cabo el estudio de las características y ventajas de las herramientas utilizadas en los procesos de desarrollo de software.

---

<sup>24</sup> Siglas del inglés Structured Query Language.

<sup>25</sup> Siglas del inglés Object Relational Mapping (Mapeador de Objetos Relacionales).



### CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

#### 2.1 Introducción

En este capítulo se hace un estudio del proceso actual de administración y seguridad de los sistemas informáticos en las oficinas del CIRP definiendo las principales deficiencias que existen con el fin de dar solución a las mismas. Se definen y describen las principales funcionalidades que debe cumplir el sistema para llevar a cabo la gestión y configuración de los principales elementos que son manejados en los procesos de identificación de las personas.

#### 2.2 Modelo de dominio

Después de haber hecho un estudio detallado de cómo se llevan a cabo los procesos de gestión de los elementos administrativos en las unidades del CIRP, se realizó un modelo de dominio que esquematizara las entidades conceptuales presentes y que permitiera dejar claro el entorno y el alcance del negocio a automatizar. En el siguiente modelo de dominio se muestra la relación que existe entre los conceptos tratados en el análisis de la administración de los sistemas de las oficinas del carné de identidad, destacando los elementos de control y estructura que son utilizados. (Ver **Fig. 1** Modelo de Dominio)

2.2.1 Diagrama de clases del Modelo de Dominio

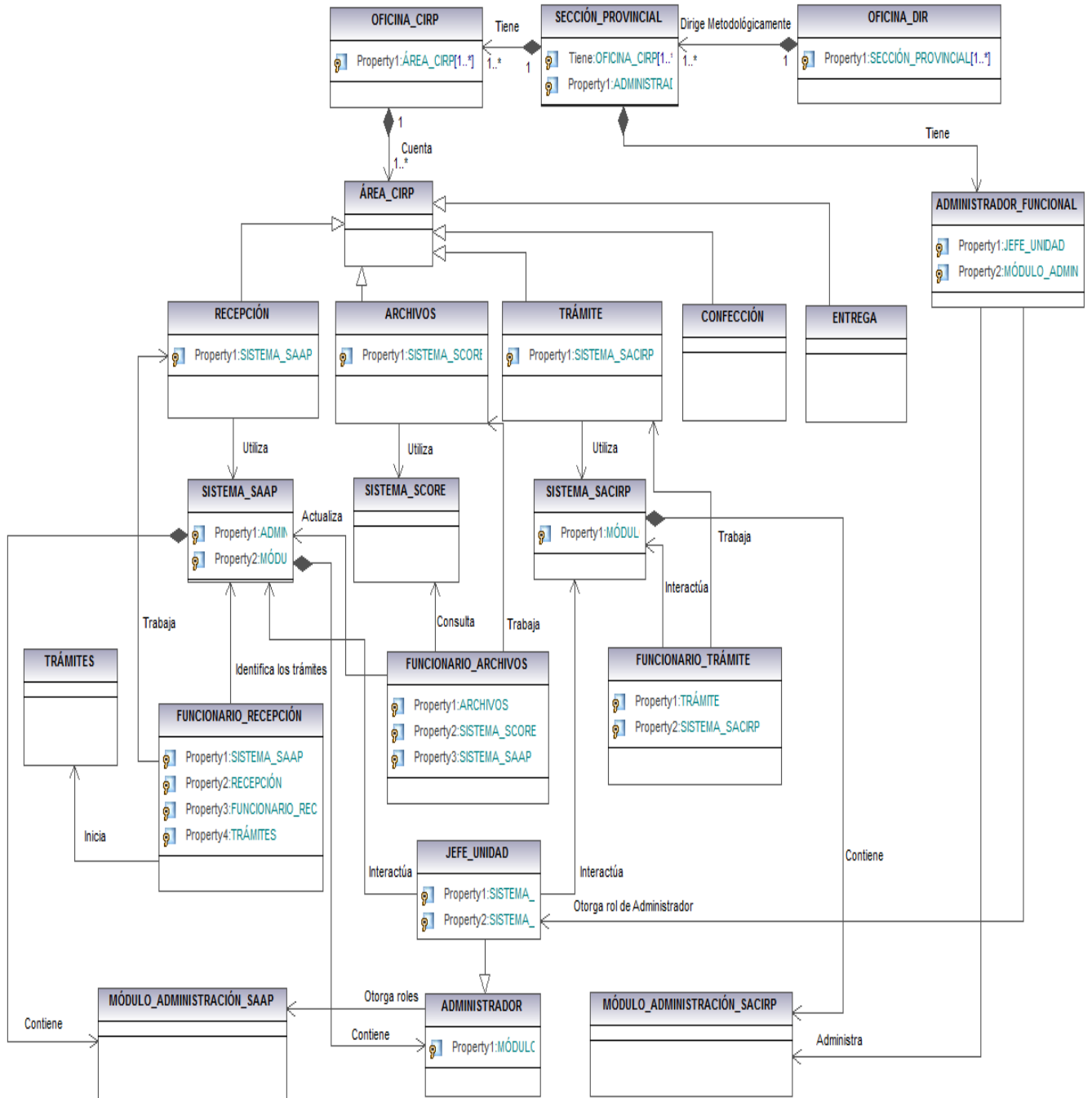


Fig. 1 Modelo de Dominio

### 2.2.2 Descripción del diagrama de clases del Modelo de dominio

Las oficinas del CIRP están estructuradas por áreas de procesos funcionalidades para llevar a cabo la tramitación de los documentos de identidad nacional, estas son: Área de Recepción, Archivos, Trámite, Confección y Entrega. En estas áreas los procesos son llevados a cabo por funcionarios los cuales tienen roles diferentes. Cuando una persona se dirige a una oficina del CIRP y solicita un trámite es atendido en la Recepción por el funcionario de dicha área. En esta área se recogen los datos de la persona en el SAAP, posteriormente el funcionario del Área de Archivos hace una búsqueda en el sistema que contiene los antecedentes penales de todas las personas (SCORE), además de buscar los datos de las personas en los archivos de la oficina. Después el funcionario del Área de Trámite con el uso del SACIRP gestiona los datos relativos al trámite que está realizando la persona. Finalmente se confecciona el documento de identidad en el área correspondiente y se le entrega a la persona. Estos sistemas cuentan con un Módulo de Administración para la gestión de los elementos administrativos. Con respecto al SAAP existe un Administrador provincial que es el encargado de otorgar el rol de administrador a cada Jefe de Unidad, el cual asigna los permisos a los funcionarios en las oficinas. Para la administración del SACIRP existe un Administrador provincial que es el encargado de gestionar las direcciones de las personas y otorgar los permisos a los funcionarios en las oficinas, excepto en Ciudad de la Habana, donde la administración se lleva a cabo directamente por la DIR.

### 2.2.3 Glosario de conceptos del Modelo de dominio

- **Oficina DIR:** Oficina de Dirección de Identificación y Registros que atiende a nivel nacional todos los procesos de identidad de las personas.
- **Oficina Provincial:** Oficina Provincial que se encarga de dirigir y controlar los procesos llevados a cabo en cada oficina del CIRP de los municipios.
- **Oficina CIRP:** Oficina del Carné de Identidad y Registro de la Población donde se realizan los trámites correspondientes a la identificación de las personas.
- **Área CIRP:** Área funcional donde se realiza un determinado grupo de procesos relacionados con los trámites en las oficinas del CIRP.
- **Recepción, Archivos, Trámite, Confección y Entrega:** Áreas que conforman las oficinas del CIRP.
- **Trámite:** Conjunto de actividades con un inicio y un fin donde el resultado en este caso es un documento de identidad.

- **Sistema SAAP:** Sistema Automatizado de Atención a la Población, donde se almacenan los datos de la persona que solicita un trámite, este sistema se encuentra en la Recepción.
- **Sistema SCORE:** Sistema donde se realiza la búsqueda de los antecedentes penales de cada persona, se encuentra en el área de Archivos.
- **Sistema SACIRP:** Sistema donde se realizan los trámites del carné de identidad, se encuentra en el área de Trámite.
- **Funcionario de Recepción:** Persona encargada de atender a la persona que solicita un trámite e inserta los datos de esta persona en el SAAP.
- **Funcionario de Archivos:** Persona encargada de buscar los antecedentes penales de la persona en el SCORE y los datos personales en los archivos.
- **Funcionario de Trámite:** Persona encargada de realizar el trámite solicitado.
- **Administrador:** Rol que asume la persona encargada de gestionar los datos en los sistemas SAAP y SACIRP.
- **Módulo de Administración del SAAP:** Módulo donde se gestionan los datos de las personas.
- **Módulo de Administración del SACIRP:** Módulo donde se gestionan las direcciones de las personas.
- **Jefe de Unidad:** Asume el rol de Administrador de los sistemas, es el encargado de gestionar los permisos sobre estos, y la información que hay en ellos.

### 2.3 Concepción de los módulos del sistema

Con el fin de automatizar y mejorar los procesos de identificación nacional que se llevan a cabo en las oficinas del CIRP actualmente se ha decidido la realización de un Sistema Único de Identificación Nacional que sea capaz de dar solución a todas las dificultades existentes con respecto a la tramitación de los documentos de identidad. El sistema se ha dividido en 7 módulos con el fin de agrupar las funcionalidades más importantes que deben ser cumplidas y así lograr una mejor estructura y organización de la secuencia de dichos procesos, estos son: Recepción, Captura de datos, Captura de documentos, Captura de imágenes, Supervisión, Personalización y Entrega. Todo sistema debe contar con un mecanismo que garantice la gestión y seguridad de la información, así como la configuración de sus elementos más importantes. El Módulo de Administración surge con la necesidad de configurar y gestionar los elementos que serán utilizados en la realización de los procesos de cada uno de los módulos antes mencionados. Para dicha gestión se han definido tres niveles que garantizan los permisos con el fin

de lograr la seguridad del sistema. Una de las funcionalidades más importantes que implementa este módulo es la Gestión de las direcciones, elemento fundamental en la identificación de la persona. También se encarga de la Gestión de nomencladores. A continuación se muestra la estructura de los módulos del SUIN. (Ver **Fig. 2** Concepción de los módulos del SUIN)



Fig. 2 Concepción de los módulos del SUIN

### 2.3.1 Roles del Módulo de Administración

El Módulo de Administración del SUIN cuenta con tres roles fundamentales que son los encargados de llevar a cabo todas las funcionalidades que este posee. Estos roles se corresponden con los tres niveles de acceso a la gestión de la información, ellos son: Administrador Nacional, Administrador Provincial y Administrador de Unidad. A continuación se muestra la descripción detallada de los mismos. (Ver **Tabla 1** Roles del Módulo de Administración)

Rol	Objetivo
Administrador Nacional	Garantizar la administración del SUIN a nivel Nacional. Específicamente puede realizar acciones tales como: crear los niveles de precedencia para conformar las reglas que se utilizarán en la confección de las direcciones y gestionar los nomencladores que son utilizados en los distintos procesos que se realizan en el sistema. También puede realizar las operaciones del Administrador a nivel Provincial y el

	Administrador a nivel de Unidad.
Administrador Provincial	Garantizar la administración del SUIN a nivel Provincial. Específicamente se encarga de ejecutar diferentes acciones en el sistema tales como: la gestión de los tipos de lugares, lugares y los inmuebles, elementos fundamentales en la confección de una dirección. Además de realizar las operaciones del Administrador a nivel de Unidad.
Administrador de Unidad	Garantizar la administración del SUIN a nivel de Unidad. Específicamente se encarga de asignarle una dirección a una persona u otro elemento.

**Tabla 1** Roles del Módulo de Administración

### 2.3.2 Especificación de los requisitos funcionales (ERF)

Para la realización del Módulo de Administración se llevó a cabo un estudio de todas las funcionalidades que este debe cumplir, con esta información se hizo la especificación de requisitos para tener un mejor entendimiento de lo que el sistema debe hacer y poder implementar un módulo con la calidad requerida. En la ERF se describen cada uno de los requisitos, que son el elemento conductor del desarrollo del sistema. A continuación se mostrará la especificación del requisito funcional “Crear Dirección. Para realizar un análisis profundo de las especificaciones de los requisitos funcionales ver **ANEXO I** “Catálogo de requisitos”.

#### **RF1.** Crear dirección

- RF1.1. Mostrar listado de las Provincias
- RF1.2. Permitir seleccionar una Provincia
- RF1.3. Mostrar listado con los Municipios de la Provincia seleccionada
- RF1.4. Permitir seleccionar un Municipio
- RF1.5. Mostrar listado con los grupos de lugares existentes
- RF1.6. Permitir seleccionar un grupo de lugar
- RF1.7. Mostrar listado con los tipos de lugares del grupo seleccionado
- RF1.8. Permitir seleccionar un tipo
- RF1.9. Mostrar una tabla con el listado de lugares existentes del tipo seleccionado

RF1.10. Permitir seleccionar un lugar

1.10.1 Mostrar un menú con las diferentes opciones que se podrán realizar a partir del lugar seleccionado.

a) Opciones a realizar:

i) Mostrar listado con los grupos de lugares contenidos en el lugar seleccionado

- Si se selecciona un grupo de lugar:
  1. Mostrar listado con los tipos de lugares del grupo seleccionado
  2. Permitir seleccionar un tipo
  3. Mostrar el listado de lugares existentes del tipo seleccionado

ii) Gestionar inmueble en el lugar seleccionado

- Mostrar listado de inmuebles existentes en el lugar seleccionado:
  1. Kilómetro
  2. Número del inmueble
  3. Código postal
  4. Tipo de inmueble
  5. Punto de referencia
- Permitir seleccionar un inmueble
- Mostrar la dirección conformada con los datos del inmueble creado y el lugar donde se encuentra
- Permitir adicionar la dirección conformada

### **2.3.3 Modelo de proceso mejorado**

Una vez que se definen los requisitos funcionales que el módulo debe satisfacer, se realiza el modelado de los procesos con el objetivo de representar las actividades llevadas a cabo por el Módulo lo más exactas posible. Cuando un proceso es modelado se puede apreciar la interrelación que existe entre las actividades, definir los puntos de contacto con otros procesos, así como identificar los subprocesos comprendidos.

Para llevar a cabo la Gestión de direcciones el primer paso es realizar la Gestión de regla de dirección, este proceso consiste en crear cada una de las estructuras que puede tener la dirección. Estas reglas son estándares utilizados para poder crear y validar una dirección. También tienen que ser gestionados los

datos de la dirección, para ello se realiza la denominada Gestión de tipos de lugares, la Gestión de lugares y la Gestión de datos del inmueble. Después que se han gestionado los elementos que conforman las direcciones se puede hacer el proceso de Gestión de dirección. Una dirección está compuesta por una Provincia, un Municipio, los datos del lugar en correspondencia con los elementos que presente, dígame Localidad, Agrupación o Vía, una dirección puede contener hasta 6 elementos ordenados de la siguiente manera: 1 Localidad, 2 Agrupaciones y 3 Vías. Una vez seleccionados estos elementos se especifican los datos que conforman un inmueble, estos son: número del inmueble, número del kilómetro si el inmueble se encuentra ubicado en alguno, un punto de referencia, el código postal y el tipo de dirección. Si la dirección creada cumple con las reglas establecidas entonces se inserta en el sistema. A continuación se muestra el diagrama de proceso mejorado del requisito “Crear dirección” (Ver **Fig. 3** Modelo de proceso mejorado “Crear dirección”). Para realizar un análisis profundo de los diagramas de proceso mejorado ver **ANEXO II** “Diagrama de proceso mejorado”.

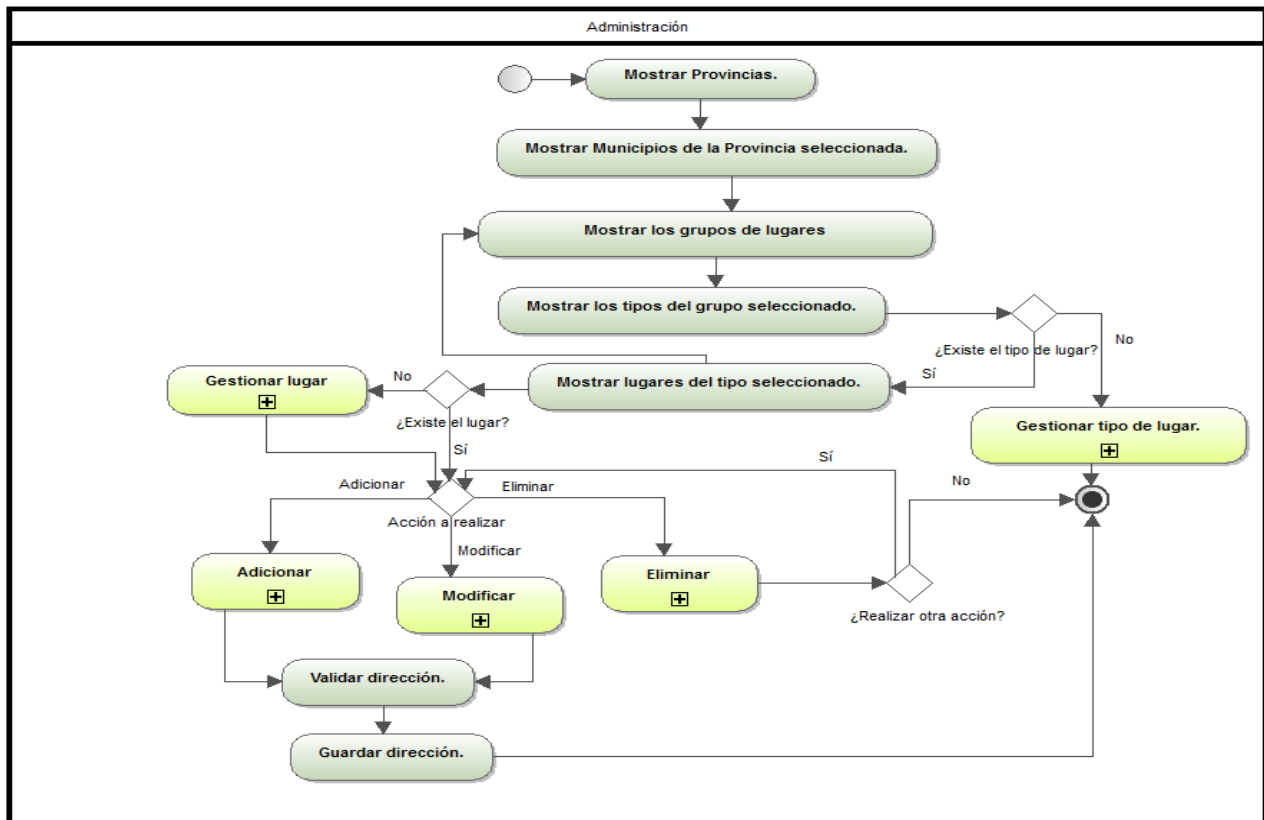


Fig. 3 Modelo de proceso mejorado “Crear dirección”



### 2.3.4 Descripción de requisitos funcionales

Una vez que los requisitos funcionales han sido identificados y se han definido los roles, entonces se describen estos requisitos. La descripción facilita la implementación del sistema ya que en ella se definen las entidades que se van a utilizar, las condiciones que debe cumplir la aplicación inicialmente, explica la secuencia de actividades entre el rol y el sistema para llevar a cabo las funcionalidades descritas, así como el resultado que se espera después de realizada cada una de las operaciones. A continuación se muestra la descripción del requisito funcional “Crear dirección” (Ver **Tabla 2** Descripción del requisito “Crear dirección”). Para realizar un análisis profundo de las descripciones de los requisitos funcionales ver **ANEXO III** “Descripción de requisitos funcionales”.

#### RF1. Crear dirección

<b>Propósito</b>	Gestionar una dirección	
<b>Roles</b>	Administrador a nivel Provincial.	
<b>Precondiciones</b>	El Administrador a nivel Provincial debe haberse autenticado con una Provincia específica. Deben existir Municipios.	
<b>Entidades tratadas</b>	<b>Entidad</b>	<b>Atributos</b>
	NGRUPOLUGAR	a) idgrupolugar b) descripcion c) fecharegistro
	NTIPOLUGAR	a) idtipolugar b) descripcion c) fecharegistro d) idgrupolugar
	NLUGARGENERAL	a) idlugargeneral b) descripcion c) sinonimo d) fecharegistro e) idlugarpadre f) idtipolugar g) idprovincia h) idmunicipio

	DINMUEBLE	<ul style="list-style-type: none"> <li>a) idlugar</li> <li>b) idtipodireccion</li> <li>c) iddireccion</li> <li>d) km</li> <li>e) codigopostal</li> <li>f) puntoreferencia</li> <li>g) numero</li> <li>h) fechaactualizacion</li> </ul>
	DDIRECCION	<ul style="list-style-type: none"> <li>a) iddireccion</li> <li>b) coordendas</li> <li>c) tira</li> </ul>
<b>Descripción</b>	<ul style="list-style-type: none"> <li>1.1 Mostrar el listado de las Provincias.</li> <li>1.2 Seleccionar una Provincia.</li> <li>1.3 Mostrar el listado de los Municipios de la Provincia seleccionada.</li> <li>1.4 Seleccionar un Municipio.</li> <li>1.5 Mostrar el listado con los grupos de lugares existentes.</li> <li>1.6 Seleccionar un grupo de lugar.</li> <li>1.7 Mostrar el listado con los tipos de lugares del grupo seleccionado en este Municipio.</li> <li>1.8 Seleccionar un tipo.</li> <li>1.9 Mostrar una tabla con el listado de lugares existentes del tipo seleccionado. <ul style="list-style-type: none"> <li>1.9.1 Mostrar un menú con las diferentes opciones que se podrán realizar a partir del lugar seleccionado. <ul style="list-style-type: none"> <li>a) Opciones a realizar: <ul style="list-style-type: none"> <li>i) Mostrar el listado con los grupos de lugares contenidos en el lugar seleccionado. <ul style="list-style-type: none"> <li>• Si se selecciona un grupo de lugar: <ul style="list-style-type: none"> <li>1. Mostrar el listado con los tipos de lugares del grupo seleccionado.</li> <li>2. Mostrar el listado de lugares existentes del tipo seleccionado.</li> </ul> </li> <li>• Si se selecciona la opción “Gestionar inmueble”: <ul style="list-style-type: none"> <li>1. Mostrar listado de inmuebles existentes en el lugar seleccionado: <ul style="list-style-type: none"> <li>a. Kilómetro</li> <li>b. Número del inmueble</li> </ul> </li> </ul> </li> </ul> </li> </ul> </li> </ul> </li> </ul> </li></ul>	

	<ul style="list-style-type: none"> <li>c. Código postal</li> <li>d. Tipo de inmueble</li> <li>e. Punto de referencia</li> </ul> <ol style="list-style-type: none"> <li>2. Seleccionar un inmueble</li> <li>3. Si se selecciona la opción “Adicionar”:             <ol style="list-style-type: none"> <li>a. Mostrar elementos que conforman un inmueble:                 <ul style="list-style-type: none"> <li>• Kilómetro</li> <li>• Número del inmueble</li> <li>• Código postal</li> <li>• Tipo de inmueble</li> <li>• Punto de referencia</li> </ul> </li> <li>b. Adicionar inmueble.</li> </ol> </li> <li>4. Si se selecciona la opción “Modificar”:             <ol style="list-style-type: none"> <li>a. Mostrar valores posibles a modificar del inmueble seleccionado.                 <ul style="list-style-type: none"> <li>• Kilómetro</li> <li>• Número del inmueble</li> <li>• Código postal</li> <li>• Tipo de inmueble</li> <li>• Punto de referencia</li> </ul> </li> <li>b. Modificar inmueble.</li> </ol> </li> <li>5. Si se selecciona la opción “Eliminar” se elimina el inmueble seleccionado.</li> <li>6. Si se selecciona la opción “Aceptar”, adicionar la dirección conformada.</li> <li>7. Si se selecciona la opción “Cancelar”, se cancela la operación de crear una nueva dirección.</li> </ol>
<b>Validaciones</b>	1. Las validaciones de las restricciones de los datos por el diccionario de datos.
<b>Postcondiciones</b>	Se crearon las direcciones.

**Tabla 2** Descripción del requisito “Crear dirección”

### 2.3.5 Prototipo de interfaz de usuario

El diseño de interfaces de usuario es una tarea fundamental en el desarrollo de un sistema, ya que es la primera vista que tenemos de la aplicación, sirve además como elemento de apoyo para la validación con el cliente. La calidad de las interfaces de usuario influye en el éxito o fracaso de una aplicación. A continuación se presenta el prototipo de interfaz de usuario como propuesta de solución al requisito funcional “Crear dirección” (Ver **Fig. 4** Prototipo de interfaz de usuario “Crear dirección”). Para realizar un estudio de los restantes prototipos de interfaz de usuario ver **ANEXO III** “Descripción de requisitos funcionales”.

Administración

Datos del inmueble

Km	Número	Código P.	Tipo	Referencia
1	5	20100	Casa	-
2	9	20200	C. Trabajo	Pto Mando
3	15	20400	Oficina	Consultorio

Adicionar    Modificar    Eliminar

Kilómetro: 91    Tipo: Escuela

Número: 37    Código P.: 20 100

Punto de referencia

Dirección

Calle 367 No.37 Km91 E/ 166 y 178 Edif:16631 Esc:2 Repa:Mulgoba Boyero Ciudad Habana C.postal:10400

Aceptar

Terminar

Fig. 4 Prototipo de interfaz de usuario “Crear dirección”

### 2.4 Modelo de datos

Una vez que se han definido y modelado las funcionalidades que debe realizar el módulo a implementar, se deben definir las entidades del negocio con sus atributos, que no son más que elementos gestionados y utilizados para realizar operaciones. Estas entidades son tratadas en la declaración de los servicios e

implementación de los métodos. En el levantamiento de requisitos se construyó una versión estable del Modelo de datos para la validación con el cliente. Este Modelo será actualizado en el Análisis y Diseño. A continuación se muestra el Modelo de datos correspondiente a la creación de direcciones y se explican sus entidades fundamentales con sus atributos. (Ver Fig. 5 Modelo de datos)

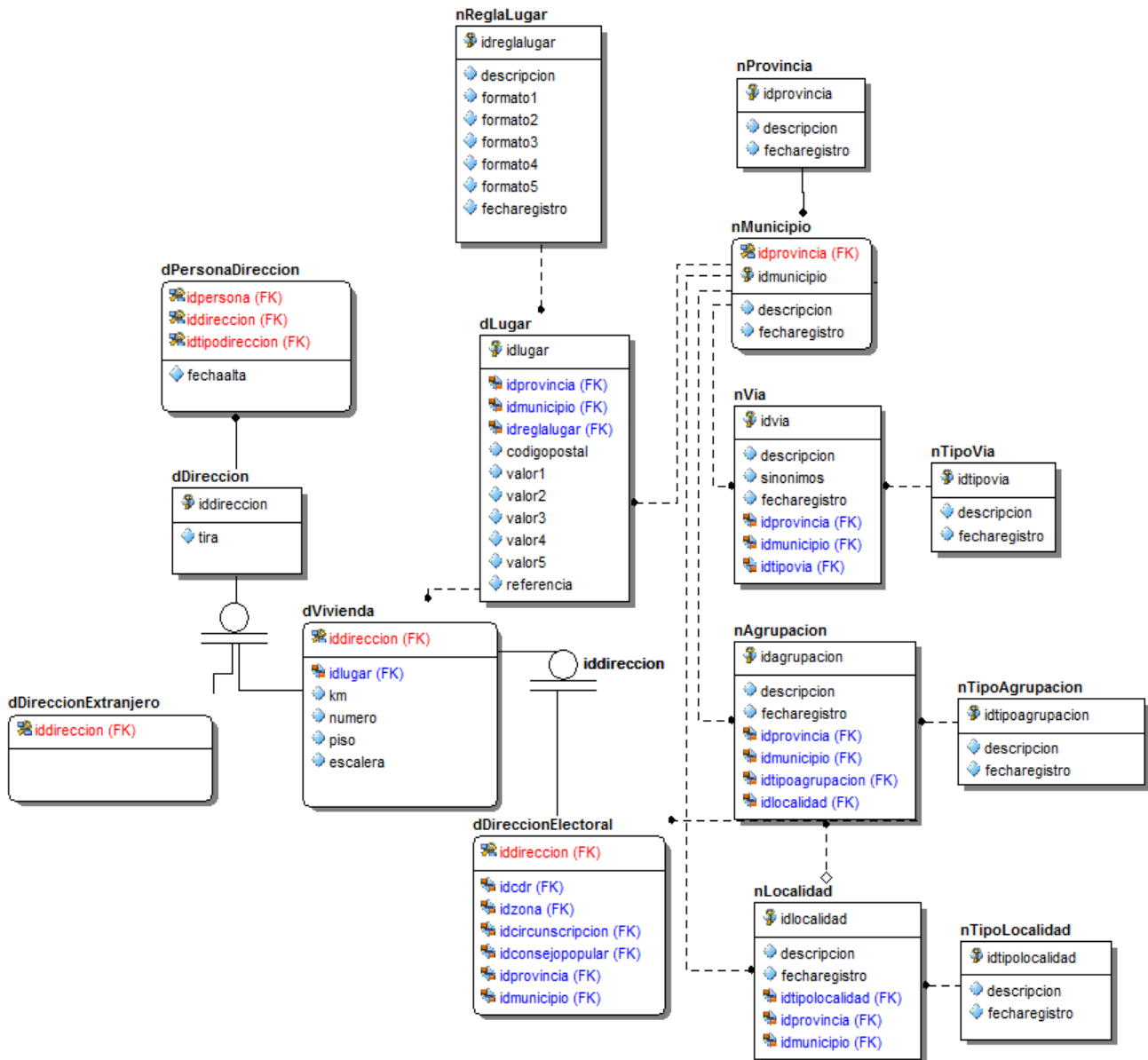


Fig. 5 Modelo de datos

**2.4.1 Glosario de entidades del Modelo de datos**

- **dVivienda:** Entidad que registra la dirección de una vivienda nacional.

Atributo	Dominio	Tipo de Datos	Nulo	Definición
iddireccion	tIdentificadorUnico	UNIQUEID	N	Identificador único de la dirección.
idlugar	tIdentificadorUnico	UNIQUEID	N	Identificador único del lugar.
km		NUMERIC(5, 0)	Y	Número de kilometraje de una carretera.
numero		CHAR(10)	Y	Número de la vivienda.
piso	tDescNomenclador	VARCHAR(100)	N	Número del piso de la vivienda.
escalera	tDescNomenclador	VARCHAR(100)	N	Escalera de la vivienda, en el caso de los edificios.

**Tabla 3** Descripción de la entidad “dVivienda”

- **nLocalidad:** Entidad que representa las localidades.

Atributo	Dominio	Tipo de Datos	Nulo	Definición
idlocalidad	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único de la localidad.
descripcion	tDescNomenclador	VARCHAR(100)	N	Descripción de la localidad.
fecharegistro		CHAR(10)	Y	Fecha en que se registra la localidad.
idtipolocalidad	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único del tipo de localidad.
idprovincia	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único de la provincia.
idmunicipio	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único del municipio.

**Tabla 4** Descripción de la entidad “nLocalidad”

- **nTipoLocalidad:** Entidad que representa los tipos de localidades.

Atributo	Dominio	Tipo de Datos	Nulo	Definición
idtipolocalidad	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único del tipo de localidad.
descripcion	tDescNomenclador	VARCHAR(100)	N	Descripción del tipo de localidad.
fecharegistro	tFechaHora	DATETIME	N	Fecha de registro del tipo de localidad.

**Tabla 5** Descripción de la entidad “nTipoLocalidad”

- **nAgrupación:** Entidad que representa las agrupaciones.

Atributo	Dominio	Tipo de Datos	Nulo	Definición
idagrupacion	tIdNomenclador	NUMERIC(4, 0)	N	Identificador de la agrupación.
descripcion	tDescNomenclador	VARCHAR(100)	N	Descripción de la agrupación.
fecharegistro	tFechaHora	DATETIME	N	Fecha en que se registra la agrupación.
idprovincia	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único de la provincia.
idmunicipio	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único del municipio.
idtipoagrupacion	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único del tipo de agrupación.
idlocalidad	tIdNomenclador	NUMERIC(4, 0)	Y	Identificador único de la localidad.

Tabla 6 Descripción de la entidad "nAgrupación"

- **nTipoAgrupacion:** Entidad que representa los tipos de agrupaciones.

Atributo	Dominio	Tipo de Datos	Nulo	Definición
idtipoagrupacion	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único del tipo de agrupación.
descripcion	tDescNomenclador	VARCHAR(100)	N	Descripción del tipo de agrupación.
fecharegistro	tFecha	DATE	N	Fecha en que se registra la agrupación.

Tabla 7 Descripción de la entidad "nTipoAgrupación"

- **nVía:** Entidad que registra las vías definidas en todo el país.

Atributo	Dominio	Tipo de Datos	Nulo	Definición
idvia	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único para la vía.
descripcion	tDescNomenclador	VARCHAR(100)	N	Descripción de la vía.
sinonimo		VARCHAR(100)	N	Otros posibles nombres que comúnmente se les da a la vía.
fecharegistro	tFechaHora	DATETIME	N	Fecha en que se registra la vía.
idprovincia	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único de la provincia.
idmunicipio	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único del municipio.

idtipovia	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único del tipo de vía.
-----------	----------------	---------------	---	--------------------------------------

**Tabla 8** Descripción de la entidad “nVía”

- **nTipoVía:** Entidad que representa el tipo de vía.

Atributo	Dominio	Tipo de Datos	Nulo	Definición
idtipovia	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único del tipo de vía.
descripcion	tDescNomenclador	VARCHAR(100)	N	Descripción del tipo de vía.
fecharegistro	tFechaHora	DATETIME	N	Fecha en que se registra el tipo de vía.

**Tabla 9** Descripción de la entidad “nTipoVía”.

## 2.5 Especificación de los requisitos no funcionales

### ➤ Usabilidad

**RnF1.** El sistema podrá ser utilizado por cualquier usuario con las siguientes características:

- Conocimientos básicos relativos al uso de una computadora.
- Conocimientos básicos del sistema operativo Windows.
- Conocimientos sólidos relativos a los procesos de negocio acorde al rol que desempeñe.

**RnF2.** El sistema será distribuido en idioma español, aunque estará preparado para soporte multilinguaje.

**RnF3.** Los términos utilizados se establecerán acorde al negocio correspondiente para facilitar la comprensión de la herramienta de trabajo.

**RnF4.** El sistema poseerá estructura y diseño homogéneos en todas sus pantallas que facilite la navegación.

- Menús laterales y desplegados que permitan el acceso rápido a la información.
- Menú de soporte que facilite el acceso a herramientas utilitarias, notificaciones del sistema y ayuda integrada.

### ➤ Seguridad

**RnF1.** El sistema estará disponible durante toda la semana en el horario laboral, según el área correspondiente.

**RnF2.** El sistema contará con diferentes niveles de acceso:



- a) De los usuarios: se realizará acorde con los roles autorizados para el uso de cada funcionalidad tanto a nivel de funciones de las aplicaciones como de información de las bases de datos.
- b) De las oficinas: se registrarán con los roles que le corresponda acorde a las funciones que le competan.
- c) De las estaciones de trabajo: se registrarán las estaciones de trabajo de forma tal que cada una accederá a las aplicaciones y funcionalidades que le corresponda según su ubicación y función en la organización.

**RnF3.** El sistema registrará todas las acciones que se realizan.

- a) Se llevará el registro del tiempo de actividad y del lugar de acceso de cada usuario y estación de trabajo para cada acción.
- b) Se podrán realizar auditorías para la comprobación de las actividades realizadas en el sistema.

**RnF4.** La conexión a los servidores de bases de datos y de aplicaciones es requerida para el correcto funcionamiento.

**RnF5.** Las réplicas de información se realizarán con la menor latencia posible entre los nodos.

**RnF6.** El tiempo medio de reparación de fallas del software no atribuibles al hardware o a la conectividad se definirá a partir de los Acuerdos de Niveles de Servicios establecidos entre el MININT y la UCI.

**RnF7.** El sistema garantizará la mayor exactitud en los datos.

### ➤ Interfaz

#### Interfaz de usuario

**RnF1.** Interfaz accesible e intuitiva, el manejo de las funcionalidades del portal debe ser lo más intuitivo posible, de manera que sean muy claras las posibles acciones a llevar a cabo y la manera de hacerlas.

**RnF2.** Interfaz consistente con las pautas de diseño definidas para los sistemas del proyecto. (36)

**RnF3.** Internacionalización de la interfaz.

**RnF4.** Consistencia de la aplicación entre los distintos navegadores. Debe visualizarse y manejarse de la misma forma en los navegadores más comunes agrupados, a continuación según su motor/intérprete de HTML, CSS y JavaScript:

- a) Trident (Internet Explorer a partir de la versión 7).
- b) WebKit (Konqueror, Safari, Google Chrome)
- c) Gecko (Mozilla Firefox, Flock)
- d) Opera

### **Interfaz de Software**

**RnF1.** Se requieren las librerías (Dll) y componentes que permitan la integración con el componente de PKI para los elementos de seguridad en las comunicaciones y gestión de usuarios.

**RnF2.** Se requieren las librerías (Dll) y componentes que permitan la integración con el componente *Device Grid Manager* para la gestión de los dispositivos de *hardware* con los que se interactúa en el sistema.

**RnF3.** Se requiere la librería para la actualización de las aplicaciones.

**RnF4.** Se requiere la librería de Oracle que permita conectarse al servidor de aplicaciones.

### **Interfaz de Comunicación**

**RnF1.** Los servicios web poseerán una interfaz que permita manejar un alto nivel de seguridad haciendo uso de SSL<sup>26</sup>.

**RnF2.** Se deben exponer servicios web que permitan la comunicación con los distintos sistemas que necesiten notificar u obtener información del sistema.

---

<sup>26</sup> Secure Socket Layer (Protocolo de Capa de Conexión Segura).

### 2.6 Conclusiones parciales

- En este capítulo se realizó un estudio de los procesos de negocio actuales con respecto a la administración de los sistemas informáticos presentes en las oficinas del CIRP mediante los Manuales de Usuario de estos y otros documentos con el objetivo de definir las deficiencias existentes, el cual permitió la concepción de un Módulo de Administración que gestione todos los elementos administrativos garantizando la seguridad de la información manejada.
- Se realizaron los diagramas de procesos para mostrar las tareas que deben realizar los desarrolladores describiendo qué hacer para darle cumplimiento a cada funcionalidad.
- Se mostró el catálogo de requisitos donde se definieron todas las funcionalidades que debe cumplir el módulo así como su descripción para un mejor entendimiento. También se mostraron los requisitos no funcionales que son las características que debe tener el sistema.
- Se expuso una versión estable del Modelo de datos presentando las principales entidades y sus atributos.

### CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

#### 3.1 Introducción

El flujo de trabajo de Análisis y Diseño es un eslabón fundamental en el proceso de desarrollo pues se crean las bases para la implementación. En este capítulo se realizará un análisis profundo de las funcionalidades descritas con el objetivo de definir las clases y sus relaciones, así como los servicios y el diseño del *Workflow*, que serán los elementos de entrada a las actividades de implementación y prueba, para ello se hace referencia a las principales entidades del negocio.

#### 3.2 Arquitectura de la solución

El Módulo de Administración del SUIN se desarrolla bajo la Arquitectura Orientada a Servicios (SOA) que es un paradigma arquitectónico que se basa en procesos de negocio y sistemas heterogéneamente distribuidos de diferente procedencia, los nodos son expuestos como servicios independientes o autónomos que contienen un conjunto de funcionalidades técnicas, cuya principal característica es que son reutilizables por cualquier sistema o componente.

##### 3.2.1 Vista lógica del Módulo de Administración

El Módulo de Administración como parte del SUIN está diseñado siguiendo un patrón de capas bien definidas y diseñadas, que se especializan en la realización de actividades específicas para evitar la dependencia de una tecnología o aplicación.

La arquitectura se encuentra representada por 5 capas lógicas (Ver **Fig. 6** Capas y sus componentes) que dan un alto nivel de encapsulamiento de las responsabilidades, permitiendo reducir el acoplamiento y aumentar la reutilización entre las mismas. Esta distribución de las capas permite que se realicen grandes cambios sin siquiera tener que realizar cambios en las demás capas. Una vez que estas estén bien definidas la comunicación entre ellas se realizará solo a nivel de interfaces que permiten trabajar de manera transparente a las instancias reales.

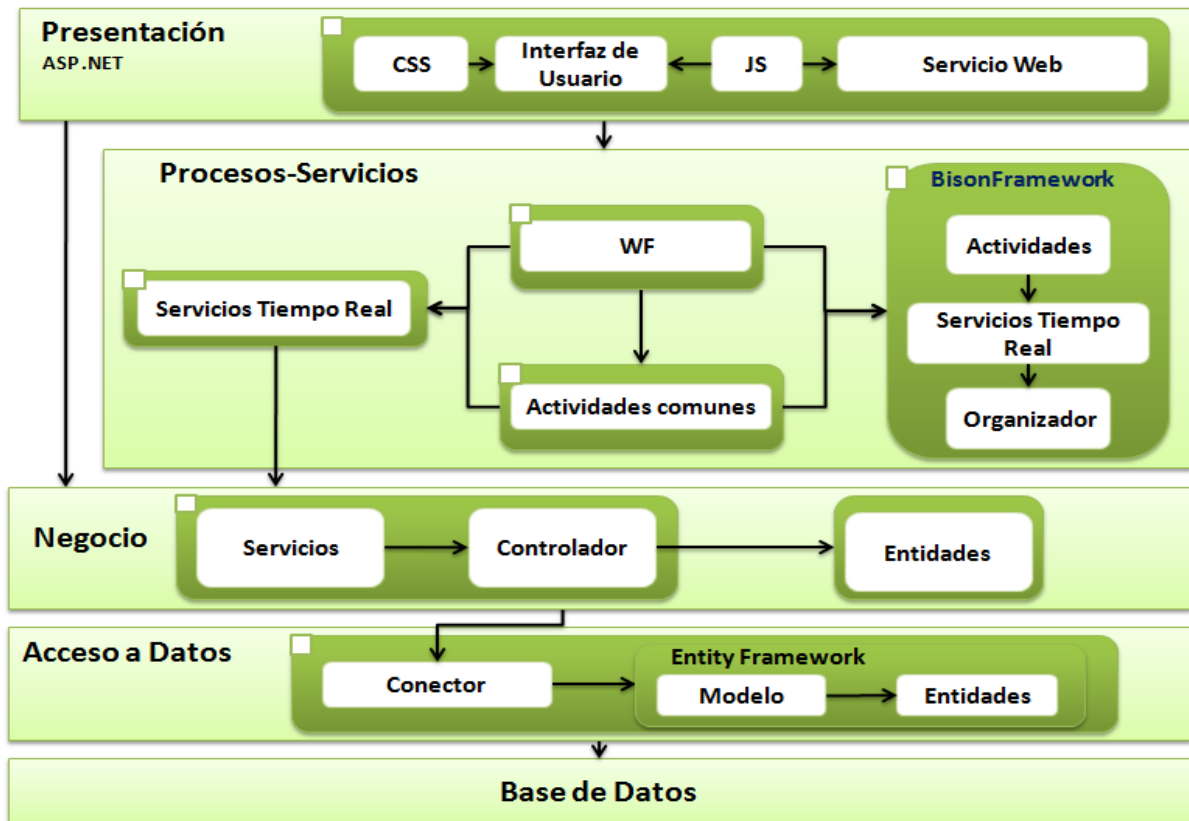


Fig. 6 Capas y sus componentes

## Descripción de las Capas de la Arquitectura:

### 1. Presentación

Tener los requerimientos y la lógica de la interfaz de usuario dentro de una capa separada permite tanto su reutilización como su independencia de la lógica de negocio. Permitiendo que se pueda modificar la interfaz sin que esto afecte al negocio y viceversa. En el sistema esta capa posee un grado de independencia mayor, ya que el flujo de las interfaces está gestionado no por esta capa sino por el motor de procesos, logrando que un cambio en la secuencia de las actividades del proceso de negocio tenga la menor cantidad de implicaciones en la capa de presentación.

Está compuesta por todas las interfaces de usuario y los componentes necesarios para su correcto funcionamiento. Estos elementos pueden ser ficheros Java Script, CSS, servicios consumidos por Java

Script, etc. Esta capa se encuentra representada por el proyecto Web de la aplicación, y tiene interacción directa con la capa de Procesos y Servicios y con la de Negocio.

### **2. Procesos y Servicios**

Tiene contenida la lógica de los procesos de negocio representada 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 *Runtime*<sup>27</sup>, 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 de dicho servicio. Al mismo nivel se encuentra definida una fábrica para los servicios de Runtime, que le permite a la capa de presentación interactuar con estos sin necesidad de que sepan la instancia concreta del servicio que utilizan. Todos estos elementos se encuentran vinculados directamente con el Bison Framework.

### **Bison Framework**

Representa el *framework*<sup>28</sup> común, y su principal objetivo es el de encapsular en un componente todas aquellas actividades y servicios que son de vital importancia y manipulan cada elemento del negocio. Es un *framework* para la orquestación de procesos de negocio con WF.

### **3. 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. 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.

### **4. Acceso a Datos**

La capa de acceso a datos permite la persistencia de los objetos de negocio, esta capa se utiliza con el objetivo de estandarizar el acceso a datos, y eliminar el acoplamiento de los componentes de negocio al gestor de base de datos. De esta forma se puede minimizar el impacto en caso de ocurrir cambios en el

---

<sup>27</sup> Tiempo de ejecución.

<sup>28</sup> De la definición en inglés de marco de trabajo.

modelo de datos. Es la única capacitada para interactuar con la capa de Base de Datos, y a su vez brinda servicios a la capa de Negocio.

### 5. Base de Datos

La capa de acceso a datos está directamente relacionada con los servicios definidos en el negocio. Para establecer esta relación hace uso de las 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 fuente de datos establecida.

### 3.3 Patrones de diseño

El desarrollo de un sistema conlleva en la mayoría de las ocasiones darle solución a problemas muy complejos que ya alguien más ha resuelto. Por esta razón uno de los pasos a tener en cuenta cuando se decide desarrollar un proyecto de software es identificar qué patrones pueden ser utilizados. Un patrón es una solución estándar para un problema común de programación.

#### 3.3.1 Patrones

En el desarrollo del Módulo de Administración se han aplicado un conjunto de patrones que permiten darle flexibilidad y no constituyen cambios grandes en el rendimiento del mismo, además se identificaron un conjunto de patrones específicos para el desarrollo de los Workflows que brindan claridad y fortaleza a los diseños de estos.

- **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 sólo posean como elementos públicos aquellos que son exclusivamente necesarios.

Ejemplo:

```
private DescriptionField<Guid> idaddress;  
public Address(string addressstrip)  
{  
this.AddressStrip = addressstrip;  
}
```

- **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 en las funciones y la información que almacenan, pueden estar diferenciadas en cierta medida. Ver **Fig. 7** Servicio Runtime “Crear dirección”.

Ejemplo:



**Fig. 7** Servicio Runtime “Crear dirección”

- **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.
- **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.
- **Singleton:** se asegura que solo se pueda crear una instancia de la clase y ofrece un punto global de acceso a esta instancia. El uso de este patrón permite que los servicios puedan ser creados solo una vez.



Ejemplo:

```
static private SuinBusinessFactory instance;
public static SuinBusinessFactory Instance
{
    get {
        if (instance == null)
        {
            Instance = new SuinBusinessFactory ();
        } return instance;
    }
}
```

- **GRASP:** el uso de este patrón está totalmente ligado a cada componente desarrollado en el sistema, donde cada uno de ellos posee solo las funcionalidades acorde a las particularidades que lo caracterizan.

### 3.3.2 Patrones de Workflow

Los patrones van desde los más simples como el patrón secuencial hasta los más complejos, por ejemplo, el patrón de sincronización. Los patrones de Workflow pueden ser clasificados en las siguientes categorías:

- **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.
- **Patrones de ramificación avanzada y sincronización:** Estos patrones superan a los patrones de control de flujo básico al permitir tipos avanzados de bifurcación y sincronización.
- **Patrones estructurales:** Estos patrones permiten terminar un subproceso cuando ya no haya nada que hacer, o permiten definir ciclos de forma arbitraria.
- **Patrones que manejan múltiples instancias:** Cuando se le da seguimiento a un caso, algunas veces es necesario que el proceso sea instanciado muchas veces.

### 3.4 Especificación de clases

Con el objetivo de dar solución a las funcionalidades que debe cumplir el módulo propuesto se realiza un análisis profundo de estas, definiéndose las clases y servicios que serán utilizados en el cumplimiento de

cada requisito funcional, las cuales se muestran a continuación. Para tener conocimiento de las demás clases y servicios ver **ANEXO IV** “Clases y servicios”.

### 3.4.1 Clases Entidades

Las clases entidades son las encargadas de modelar información que posee larga vida y que es a menudo persistente. Constituyen una representación de los datos guardados en la base de datos y permite la presencia de los mismos a lo largo del ciclo de vida del Workflow. Ofrecen soporte a los servicios y manejan actividades específicas. El Módulo de Administración hace uso de la clase “Address” que se utiliza para manejar los datos referentes a las direcciones y es el sustento de información para la confección del carné de identidad. (Ver **Tabla 10** Descripción de la entidad “Address”)

<b>Nombre de la Entidad</b>	Address	
<b>Descripción</b>	Entidad que representa la generalización de las direcciones.	
<b>Nombre del atributo</b>	<b>Tipo</b>	<b>Descripción</b>
idaddress	Guid	Identificador único de la dirección.
addressstrip	string	Cadena que guarda los datos de la dirección de la vivienda en un formato dado.

**Tabla 10** Descripción de la entidad “Address”

### 3.4.2 Clases Conectoras

Las clases conectoras son las que manejan directamente el acceso a la base de datos. Manipulan entidades de acceso a datos, y cuentan con la estructura de la lógica de acceso a datos, permitiendo una independencia total del gestor de base de datos a utilizar, el cambio en la base de datos solo genera la actualización de estas clases.

<b>Nombre</b>	IAddressConnector
<b>Descripción</b>	Es la interfaz que contiene la declaración de los métodos que son llamados por el controlador.
<b>Métodos</b>	<b>Descripción</b>
SaveAddress	Método para salvar una dirección.
UpdateAddress	Método para modificar una dirección.
DeleteAddress	Método para eliminar una dirección.
LoadAddress	Método que devuelve el listado de direcciones existentes.

**Tabla 11** Descripción de la interfaz “IAddressConnector”.

<b>Nombre</b>	AddressConnectorDAL
<b>Descripción</b>	Es la clase que contiene la implementación de los métodos de acceso a dato.
<b>Métodos</b>	<b>Descripción</b>
SaveAddress	Método para adicionar una dirección.
UpdateAddress	Método para modificar una dirección.
DeleteAddress	Método para eliminar una dirección.
LoadAddress	Método que devuelve el listado de direcciones existentes.

**Tabla 12** Descripción del conector “AddressConnector”.

### 3.4.3 Clases Gestoras

Las clases gestoras utilizan a las clases conectoras utilizando 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, además permite la realización de las pruebas a la aplicación mediante la implementación de conectores de pruebas que no accedan a datos reales.

<b>Nombre</b>	AddressController	
<b>Descripción</b>	Es la encargada de conectar los servicios de negocio con el acceso a datos.	
<b>Nombre del atributo</b>	<b>Tipo</b>	<b>Descripción</b>
AddressConnector	IAddressConnector	Permite la relación con la clase conectora.
<b>Métodos</b>	<b>Descripción</b>	
SaveAddress	Llama al método “SaveAddress” de la clase IAddressConnector para guardar una dirección.	
ModifyAddress	Llama al método “ModifyAddress” de la clase IAddressConnector para modificar una dirección.	
DeleteAddress	Llama al método “DeleteAddress” de la clase IAddressConnector para eliminar una dirección.	
LoadAddress	Llama al método “LoadAddress” de la clase IAddressConnector para cargar las direcciones existentes.	

**Tabla 13** Descripción del controlador “AddressController”.

### 3.4.1 Diagrama de clases del diseño

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones. A continuación se muestra el diagrama de clases del diseño del requisito funcional “Crear dirección”, donde se muestran las clases y sus relaciones que interactúan para dar solución a esta funcionalidad. (Ver Fig. 8 Diagrama de clases del diseño)

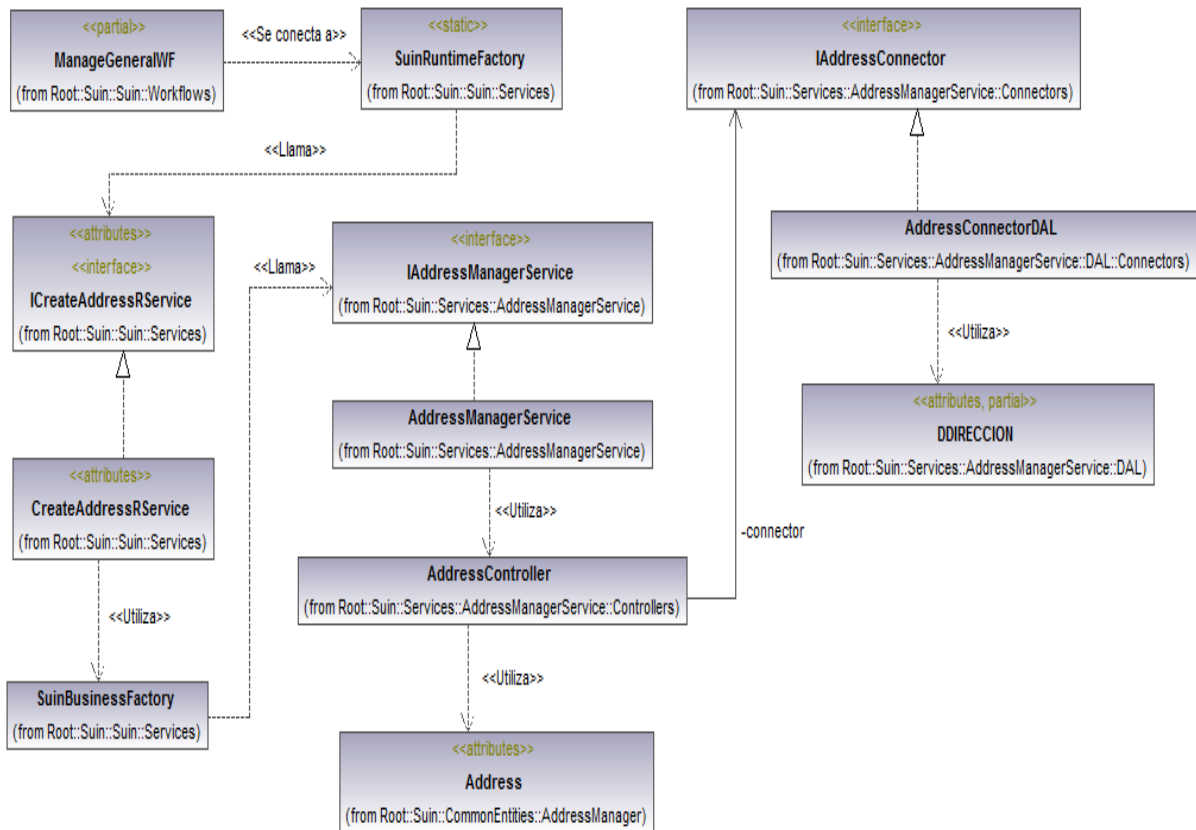


Fig. 8 Diagrama de clases del diseño

### 3.5 Servicios del sistema

Un servicio no es más que una función autocontenida, que acepta llamada(s) y devuelve respuesta(s) mediante una interfaz bien definida que lo comunica con sistemas externos a él, independientemente del lenguaje de programación o tecnología con la que se trabaje. Los servicios pueden también ejecutar acciones discretas de trabajo como serían editar y procesar una transacción. Los servicios 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. Los servicios brindan:

- Mejora en los tiempos de realización de cambios en procesos.
- Facilidad en modelos de negocios basados en colaboración con otros entes y colaboración con terceros.
- Facilidad para reemplazar elementos en los servicios sin consecuencias en el proceso de negocio.

En el Módulo de Administración los servicios han sido desglosados en dos grupos: Servicios de *Runtime* (*Runtime Services*) y Servicios de Negocio (*Bussiness Services*).

### 3.5.1 Servicios de negocio

Comunican el flujo de procesos con las capas inferiores, gestionan junto a las clases gestoras, controladoras y conectoras la información contenida en la base de datos.

<b>Nombre</b>	IAddressManagerService
<b>Descripción</b>	Es la interfaz que contiene todos los métodos que se van a utilizar para darle cumplimiento a los requisitos funcionales.
<b>Métodos</b>	<b>Descripción</b>
SaveAddress	Método que permite adicionar una dirección.
ModifyAddress	Método que permite modificar una dirección.
DeleteAddress	Método que permite eliminar una dirección.
LoadAddress	Método que permite listar las direcciones existentes.

**Tabla 14** Descripción de la interfaz del servicio de negocio "IAddressManagerService"

<b>Nombre</b>	AddressManagerService	
<b>Descripción</b>	Es donde se implementan los métodos declarados en la interfaz de servicios de negocio.	
<b>Nombre del atributo</b>	<b>Tipo</b>	<b>Descripción</b>
controller	AddressController	Permite utilizar los métodos definidos en la Interface "IAddressConnector".
<b>Métodos</b>	<b>Descripción</b>	
SaveAddress	Método que accede a la clase "AddressController" para el acceso al método "SaveAddress".	
ModifyAddress	Método que accede a la clase "AddressController" para el acceso al método "ModifyAddress".	

DeleteAddress	Método que accede a la clase "AddressController" para el acceso al método "DeleteAddress".
---------------	--

**Tabla 15** Descripción de la clase "AddressManagerService"

### 3.5.2 Servicios de Runtime

Trabajan en la captura y manejo de eventos con el usuario, es quien controla y maneja el intercambio de información con la capa de presentación. Tiene la responsabilidad de gestionar los eventos que comunican el sistema con el flujo de procesos (Capa de Procesos y Servicios).

<b>Nombre</b>	ICreateAddressRService
<b>Descripción</b>	Es donde se declaran los eventos y métodos utilizados en el Workflow.
<b>Métodos</b>	<b>Descripción</b>
OnAddAddress	Evento que se lanza cuando se va a adicionar una dirección.
OnModifyAddress	Evento que se lanza cuando se va a modificar una dirección.
OnDeleteAddress	Evento que se lanza cuando se va a eliminar una dirección.
AddAddress	Método para capturar el evento "OnAddAddress".
ModifyAddress	Método para capturar el evento "OnModifyAddress".
DeleteAddress	Método para capturar el evento "OnDeleteAddress".
MethodAddAddress	Método para adicionar una dirección.
MethodUpdateAddress	Método para modificar una dirección.
MethodDeleteAddress	Método para eliminar una dirección.

**Tabla 16** Descripción de la interfaz de servicio de *Runtime* "ICreateAddressRServices"

<b>Nombre</b>	CreateAddressRServices	
<b>Descripción</b>	Es donde se implementan los eventos y métodos utilizados por el Workflow.	
<b>Nombre del atributo</b>	<b>Tipo</b>	<b>Descripción</b>
manager	IAddressManagerService	Permite el acceso a los métodos del servicio de negocio.
<b>Métodos</b>	<b>Descripción</b>	
OnAddAddress	Evento que se lanza cuando se está insertando una dirección.	
OnModifyAddress	Evento que se lanza cuando se está modificando una dirección.	
OnDeleteAddress	Evento que se lanza cuando se está eliminando una dirección.	

AddAddress	Método para capturar el evento "OnAddAddress".
ModifyAddress	Método para capturar el evento "OnModifyAddress".
DeleteAddress	Método para capturar el evento "OnDeleteAddress".
MethodAddAddress	Método para adicionar una dirección.
MethodUpdateAddress	Método para modificar una dirección.
MethodDeleteAddress	Método para eliminar una dirección.

**Tabla 17** Descripción de la clase "CreateAddressRServices"

### 3.6 Diseño del Workflow

La tecnología de Workflow Foundation se utiliza para automatizar una secuencia de acciones o actividades para la ejecución de procesos, captura la interacción entre las entidades del mundo real, incluyendo el seguimiento del estado de cada una de ellas.

Para manejar cada una de las entidades con las que interactúa el Workflow se necesitan estructuras de control que manipulen la información contenida dentro del proceso, las que, entre otras funciones aportan herramientas similares a las usadas en distintos lenguajes y que cumplen tareas equivalentes.

#### Algunas de estas estructuras de control del flujo son:

- Condicionales (IfElseActivity, Clases Policy Activity)
- Boucles While – Until (Clases WhileActivity, ReplicatorActivity)
- Disparo y Captura de excepciones (Clases ThrowActivity, FaultHandlerActivity)
- Manejo de Eventos (HandleExternalEventActivity, EventDrivenActivity)
- Ámbito de Ejecución (Clase SincronizationScopeActivity)

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, son bloques con los que se construyen los flujos de trabajo. Las Actividades pueden ser simples o compuestas, las que son creadas por el propio desarrollador al heredar de la clase "Activity" o combinando varias actividades simples.

### Actividades de WF utilizadas en el Módulo de Administración:

#### ➤ **CallExternalMethodActivity:**

Se utiliza 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 método a invocar.

#### ➤ **HandleExternalEventActivity**

Bloquea el flujo de trabajo hasta que el evento especificado por la propiedad *InterfaceType* sea lanzado, generalmente es utilizada para la interacción con el usuario y provee la principal fuente de extracción de información con el ámbito exterior al flujo de actividades.

#### ➤ **IfElseActivity**

Permite ejecutar condicionalmente una de varias bifurcaciones. Coloca una condición en cada rama del bloque *ifElse*. Si la condición se evalúa como *true*<sup>29</sup>, 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 de parada.

#### ➤ **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 especificarle la *url* del componente representado.

#### ➤ **AssociationActivity**

Representa la asociación de un valor a una propiedad dentro del *workflow*. El valor de origen puede ser una variable de entrada o bien un valor *string* que sea basado en la propiedad *value*. Además de representar la asociación de manera gráfica también se encarga de dejar una traza de los valores que han sido modificados a lo largo del proceso.

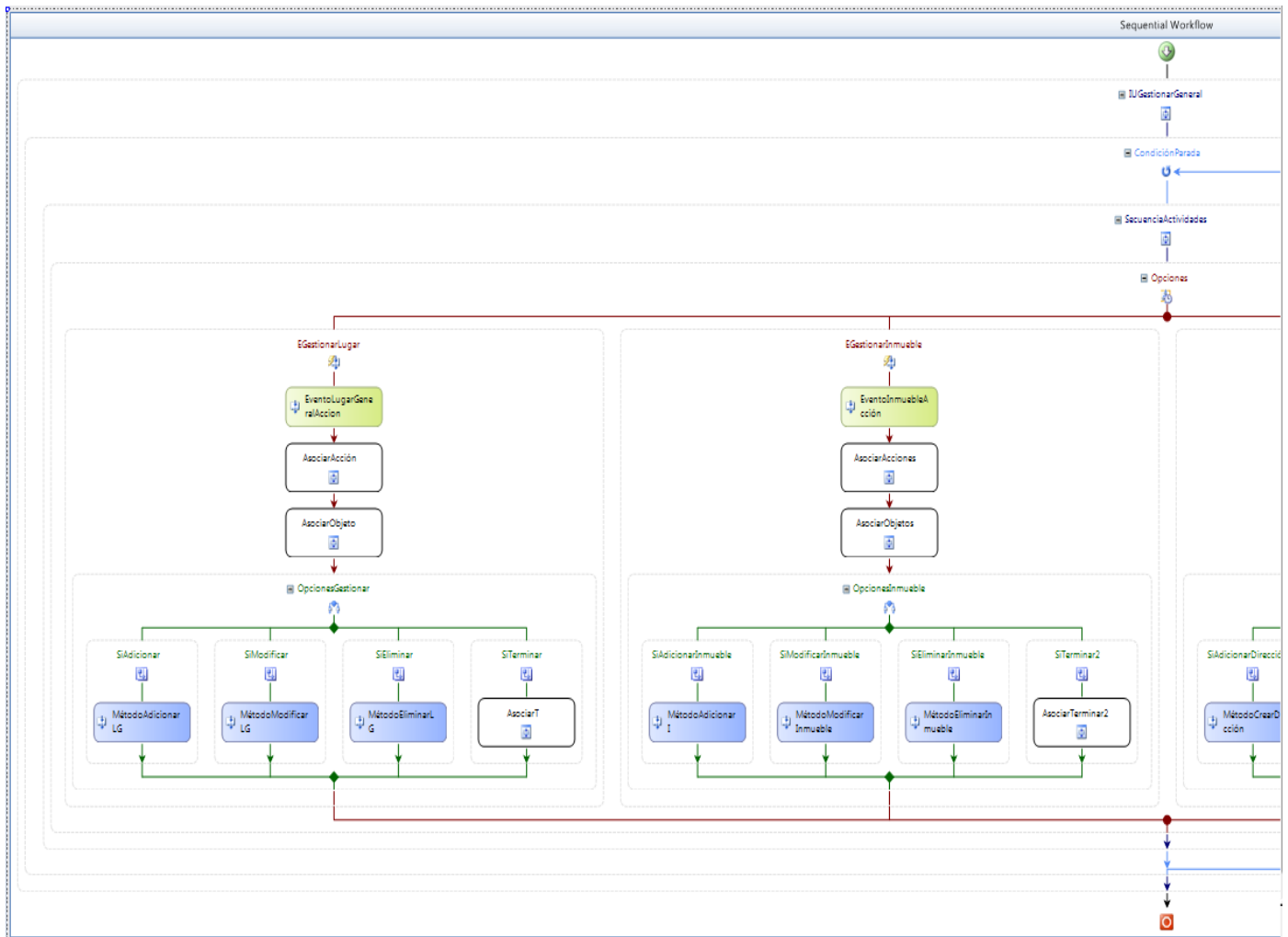
---

<sup>29</sup> Verdadero.



**Transformación del modelo de proceso mejorado a Workflow Foundation:**

El Módulo de Administración inicia uno de sus flujos de eventos cuando el funcionario de la oficina del CIRP necesita gestionar los elementos que conforman una dirección. El otro flujo de eventos que pudiera iniciarlo es la creación de una dirección. Cuando se lleva a cabo la transformación del proceso de Negocio a la modelación del Workflow cada acción del proceso de mejora pudiera traer consigo diferentes actividades en la modelación del Workflow, esto depende de la complejidad del negocio y la cantidad de entidades que se estén manejando. (Ver **Fig. 9** Modelado del Workflow (1) y **Fig. 10** Modelado del Workflow (2))



**Fig. 9** Modelado del Workflow (1)

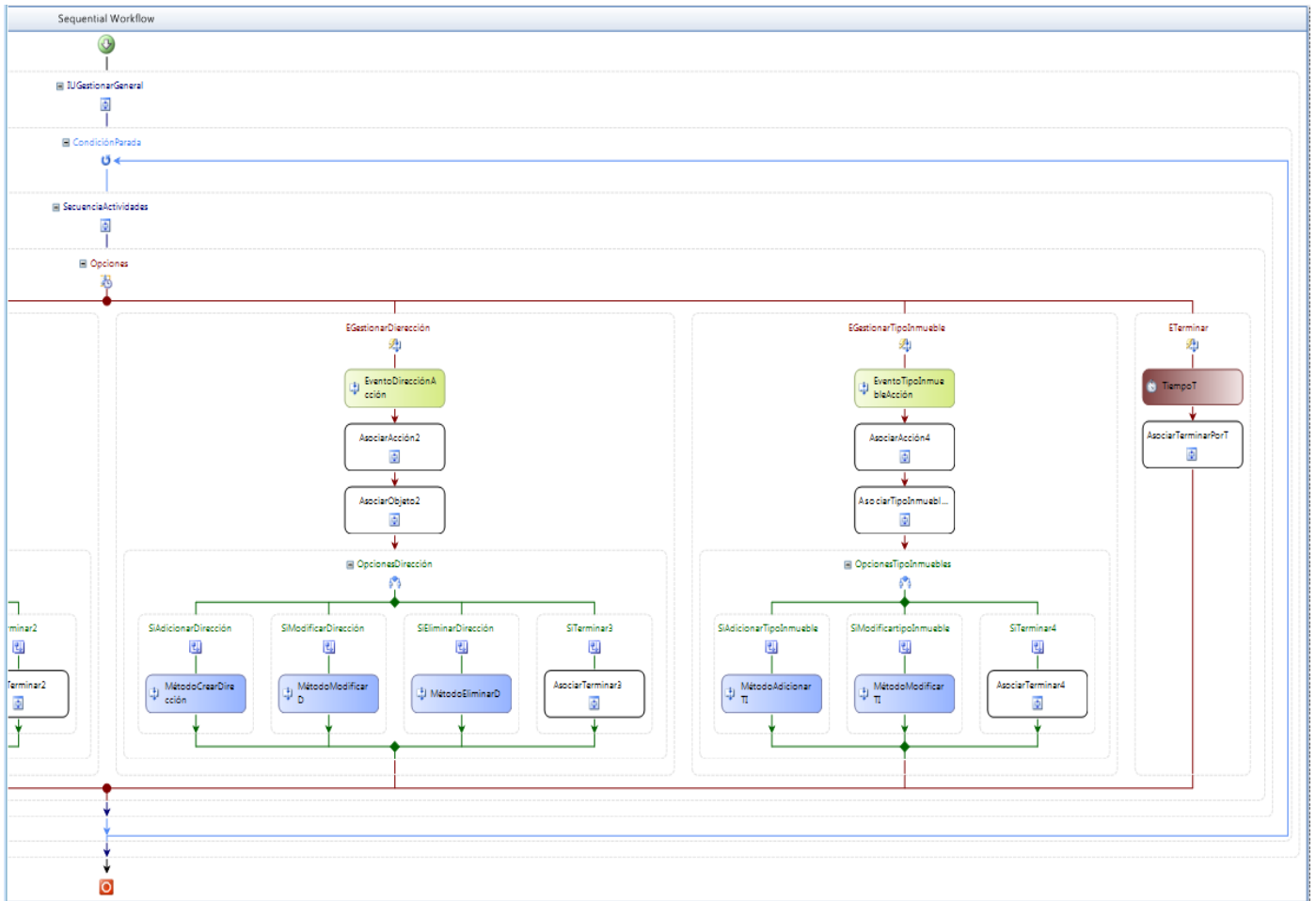


Fig. 10 Modelado del Workflow (2)

### 3.7 Modelo de Datos

Con el objetivo de obtener una especificación de requisitos completa y detallada, así como de validar con el cliente en etapas tempranas de desarrollo la visión de la persistencia de los datos asociados a la solución, se generó en la fase de levantamiento de requisitos una versión estable del modelo de datos (Ver Fig. 5 Modelo de datos). Con el avance del proceso de desarrollo de software, este modelo ha sido refinado, con la introducción de elementos que tributan o responden al cumplimiento de las funcionalidades descritas (Ver Fig. 11 Modelo de datos). Para realizar un análisis profundo de las restantes entidades del Modelo de datos ver ANEXO V “Descripción de las entidades del Modelo de datos”.

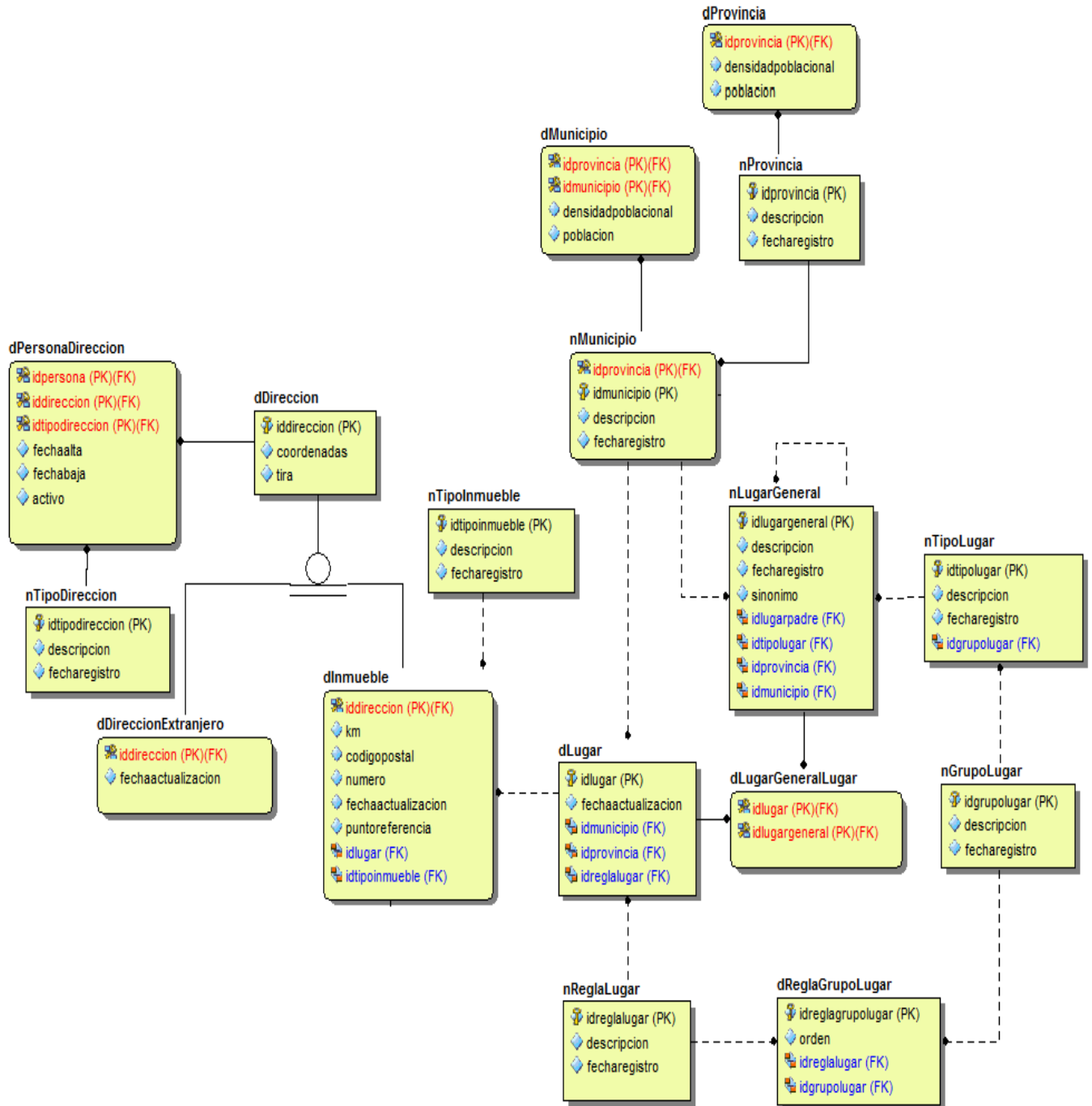


Fig. 11 Modelo de datos

**3.7.1 Descripción de las entidades fundamentales**

- **dDirección:** Entidad que representa la generalización de las direcciones.

Atributo	Dominio	Tipo de Datos	Nulo	Definición
iddireccion	tIdentificadorUnico	UNIQUEID	N	Identificador único de la dirección.
tira		CHAR(10)	Y	Cadena que guarda los datos de la dirección de la vivienda en un formato dado.
coordenadas		GEOESPACIAL	Y	Latitud y longitud de los elementos del lugar.
tipodireccion		CHAR(10)	N	Cadena que guarda el tipo de dirección.

**Tabla 18** Descripción de la entidad “dDirección”

- **nReglaLugar:** Entidad que representa las reglas para las direcciones

Atributo	Dominio	Tipo de Datos	Nulo	Definición
idreglalugar	tIdNomenclador	NUMERIC(4, 0)	N	Identificador único de cada regla.
descripcion	tDescNomenclador	VARCHAR(100)	N	Descripción de la regla.
fecharegistro	tFechaHora	DATETIME	N	Fecha en que se registra la regla.

**Tabla 19** Descripción de la entidad “nReglaLugar”

- **dInmueble:** Entidad que registra la dirección de un inmueble nacional.

Atributo	Dominio	Tipo de Datos	Nulo	Definición
iddireccion	tIdentificadorUnico	UNIQUEID	N	Identificador único de la dirección.
idlugar	tIdentificadorUnico	UNIQUEID	N	Identificador único del lugar.
km		NUMERIC(5, 0)	Y	Número de kilometraje de una carretera.
numero		CHAR(10)	Y	Número del inmueble.

**Tabla 20** Descripción de la entidad “dInmueble”

### 3.8 Conclusiones parciales

- En este capítulo se definió y describió la arquitectura del sistema, la cual se basa en un paradigma arquitectónico de orientación a servicios, lo cual posibilita que si ocurren cambios en una capa no se afecten las demás.
- Se realizó el análisis y diseño de la solución definiéndose las clases y servicios necesarios, así como la relación entre ellos, para darle solución a todas las funcionalidades descritas en el catálogo de requisitos.
- Se refinó el modelo de datos, perfeccionándose las entidades de negocio, especificándose atributos y sus relaciones para un mejor desarrollo del módulo propuesto.

## CAPTULO IV: IMPLEMENTACIÓN Y PRUEBA

### 4.1 Introducción

En el presente capítulo se muestra cómo ha sido implementada la aplicación basándose en los diagramas de componentes y el diagrama de despliegue del sistema, conformándose así un modelo de implementación donde se describe la organización y dependencia entre los nodos físicos que conforman los componentes necesarios para lograr el funcionamiento de la aplicación. Además se realizan las Pruebas Unitarias y de Sistema a todos los requerimientos, lo que permite encontrar y dar solución a los defectos encontrados para finalmente validar con el cliente.

### 4.2 Estándares de codificación

Con el objetivo de lograr una estructura y organización homogénea para la implementación de la aplicación se definen estándares de codificación que ayudan a un mejor entendimiento de la solución. A continuación se exponen los diferentes estándares definidos en la arquitectura del sistema.

#### ➤ Estilos para la capitalización

Se podrán utilizar tres convenios para la capitalización de los identificadores:

- **Pascal**

La primera letra en el identificador y la primera letra de cada subsiguiente palabra concatenada se capitalizan. Puede utilizar los identificadores de *Pascal Case* en caso de tres o más caracteres. Ejemplo:

*BackColor*

- **Camello**

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

*backColor*

- **Mayúscula**

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

*System.IO*

*System.Web.UI*

### ➤ **Sensibilidad a mayúsculas**

Para evitar confusiones y garantizar la interoperabilidad entre lenguajes, se siguieron las siguientes reglas sobre el uso de mayúsculas y minúsculas:

- No utilizar nombres o identificadores que requieran ser *case sensitivity*.
- No 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 crear *namespaces* con nombres de clases que se diferencien solo en el uso de las mayúsculas.
- No crear clases con propiedades que se diferencien solo en el uso de las mayúsculas.
- No crear clases con métodos que se diferencien solo en el uso de las mayúsculas.

### ➤ **Evitando confusión de nombre y tipo**

Se deben utilizar nombres que describan a sus identificadores en vez de nombres que describan el tipo de identificador.

## **4.3 Tratamiento de excepciones**

El diseño de un sistema no solo debe tener en cuenta lo que debe ocurrir, sino que además debe realizar un análisis profundo de las diferentes situaciones que se puedan presentar y que constituyen algún tipo de violación o de situación en particular que provocaría un error dentro del sistema.

Las excepciones son parte del diseño cuando aún la gran mayoría de ellas son identificadas en el proceso de implementación, no obstante es necesario definir con antelación un mecanismo efectivo para su tratamiento. En el caso de este sistema es necesario llevar una traza de todas aquellas excepciones que se lancen, y en muchos casos es necesario que el usuario que está interactuando con la aplicación en ese momento especifique alguna información que tribute a la futura validación o corrección de la excepción lanzada.

Para el tratamiento de excepciones se definirá una interfaz base que permita dejar una traza clara de todas las excepciones lanzadas dentro del sistema. Además se pueden considerar las mismas como parte del flujo de negocio en muchos procesos, por lo cual es necesario lograr identificar cada tipo de excepción

para darle un tratamiento a cada una de ellas, para ello se debe definir una clase en particular para aquellas excepciones que sean de interés para el negocio.

#### 4.4 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, etc. (37)

El diagrama de componentes del Módulo de Administración se ha estructurado de manera tal que se pueda apreciar la relación que existe entre cada uno de los componentes con los demás que son utilizados para implementar las funcionalidades del sistema. Se muestran los artefactos que construyen los servicios web, las entidades, elementos de presentación y acceso a datos, y cómo se referencian entre ellos. (Ver Fig. 12 Diagrama de componentes del Módulo de Administración)

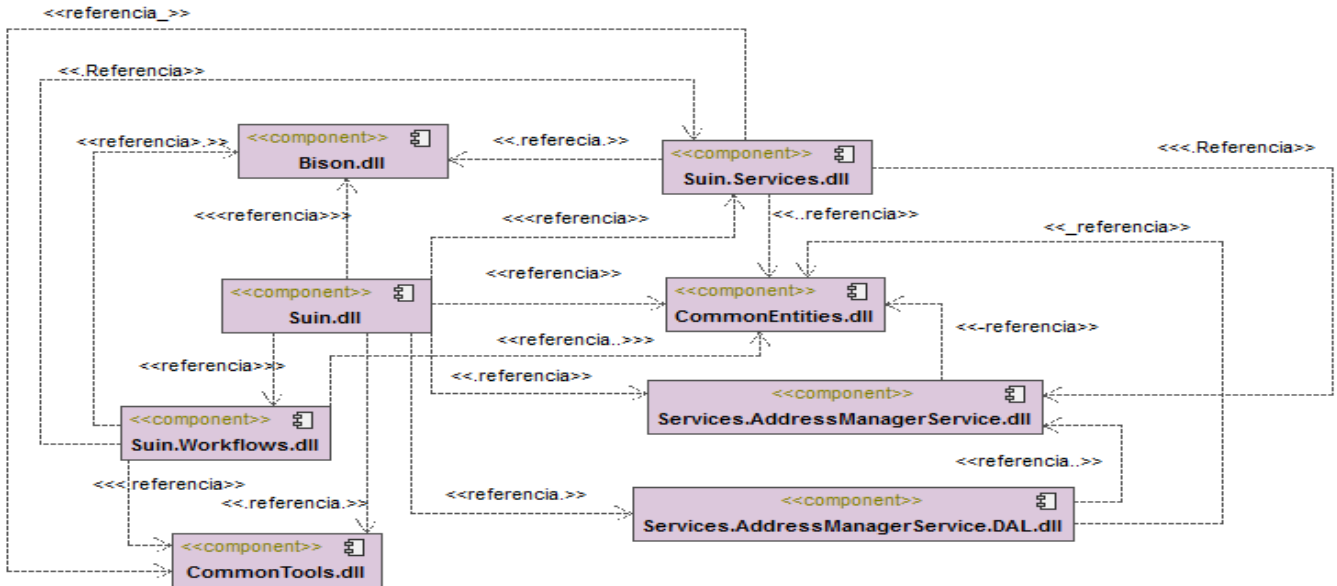


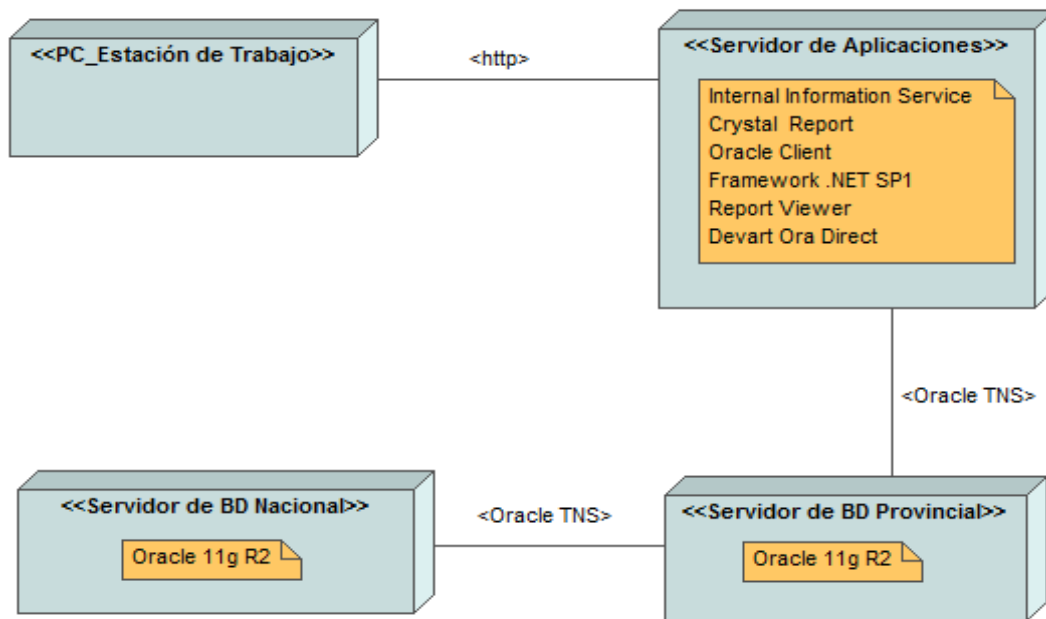
Fig. 12 Diagrama de componentes del Módulo de Administración



#### 4.5 Diagrama de despliegue

Un diagrama de Despliegue muestra cómo y dónde se desplegará el sistema. Las máquinas físicas y los procesadores se representan como nodos, y la construcción interna puede ser representada por nodos o artefactos embebidos. Debido a que los artefactos se ubican en los nodos para modelar el despliegue del sistema, la ubicación es guiada por el uso de las especificaciones de despliegue. (38)

A continuación se muestra el diseño del diagrama de despliegue del Módulo de Administración. (Ver **Fig. 13** Diagrama de despliegue del Módulo de Administración)



**Fig. 13** Diagrama de despliegue del Módulo de Administración

#### 4.6 Interfaces del sistema

La interfaz de usuario es el vínculo entre el usuario y el programa de computadora. Está compuesta por un conjunto de comandos o menús a través de los cuales el usuario se comunica con un programa. (39)

Una vez modelados los procesos de negocio y realizados los prototipos de interfaz de usuario, se desarrollaron las interfaces del Módulo de Administración, las cuales responden a las especificaciones del cliente. A continuación se muestra la interfaz de usuario donde son gestionadas las direcciones de las personas (Ver **Fig. 14** Interfaz de usuario “Crear dirección”).

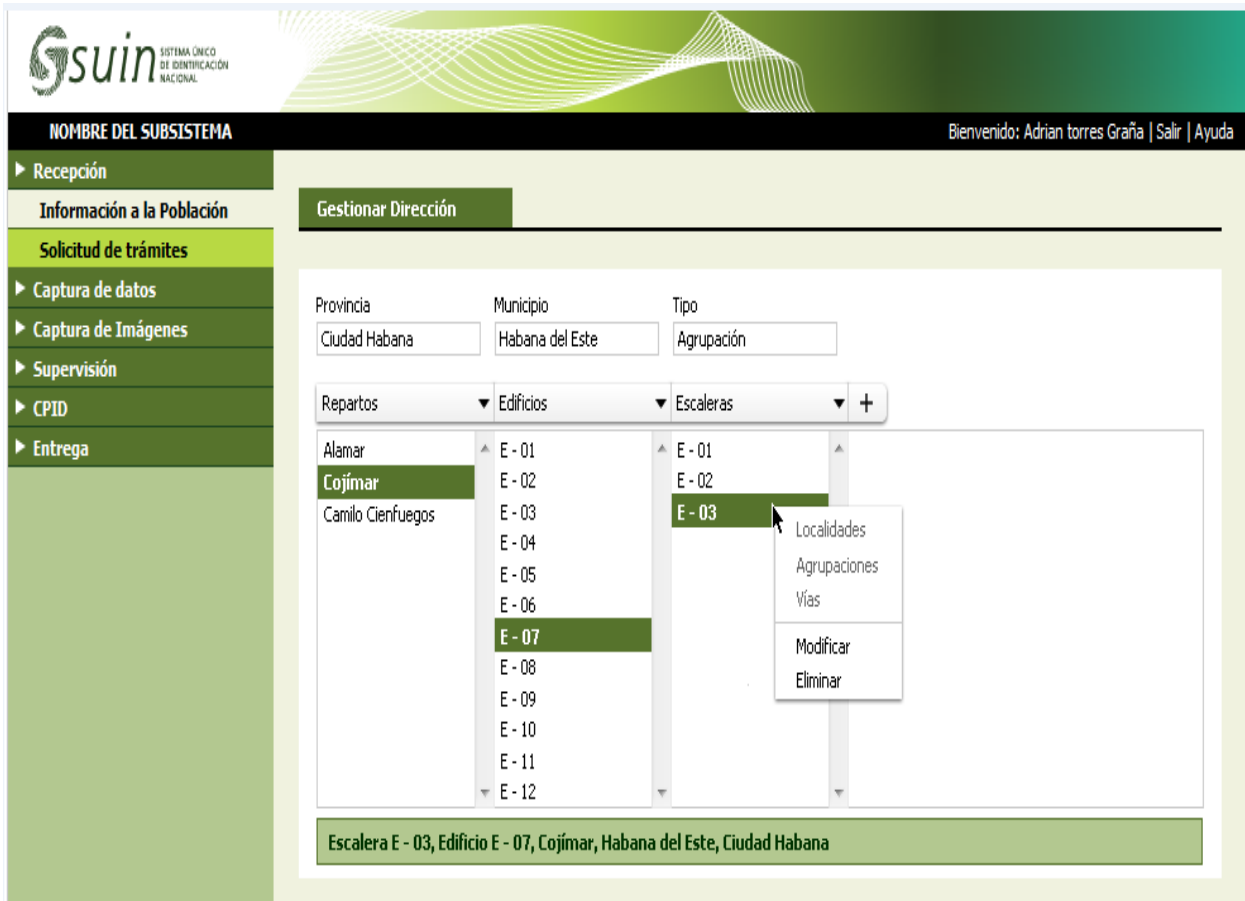


Fig. 14 Interfaz de usuario "Crear dirección"

## 4.7 Pruebas

Las pruebas constituyen una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos con el objetivo de encontrar posibles dificultades a las funcionalidades implementadas y solucionarlas.

### 4.7.1 Diseño de casos de pruebas

Los diseños de casos de prueba se llevan a cabo con el objetivo de medir los flujos de eventos de los requerimientos y obtener a partir de estas los fundamentos que sustenten la calidad del software. La generación de cada caso de prueba debe ir acompañada del resultado que ha de producir el software al

ejecutarse para detectar un posible fallo en el programa. Los casos de prueba determinan un conjunto de entradas, condiciones de ejecución y resultados esperados para un objetivo particular.

Para realizar las pruebas de caja blanca a las funcionalidades del Módulo de Administración se realizaron los diseños de casos de prueba para cada uno de los requisitos funcionales. A continuación se muestra el diseño de caso de prueba del requisito funcional “Crear dirección” (Ver **Tabla 21** Diseño de caso de prueba para el RF1 “Crear dirección”).

**Caso de prueba del RF1 “Crear dirección”:**

Condiciones de Ejecución:

1. El Administrador de la Unidad debe haberse autenticado en el sistema.
2. Debe existir el listado de las Provincias y Municipios.

**RF1. “Crear dirección”**

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
1. Crear dirección.	1.1. Crear dirección.	Permite crear una dirección que luego será asignada a personas u otros elementos que puedan ser ubicados geográficamente.	RF1.1. Se muestra el listado de las Provincias. RF1.2. Se selecciona una Provincia. RF1.3. Se muestra el listado de los Municipios de la Provincia seleccionada. RF1.4. Se selecciona un Municipio. RF1.5. Se muestra el listado con los grupos de lugares existentes en el Municipio seleccionado. RF1.6. Seleccionar un grupo de lugar RF1.7. Se muestra el listado con los tipos de lugares del grupo seleccionado. RF1.8. Se selecciona un tipo. RF1.9. Se muestra una tabla con el listado de lugares existentes del tipo seleccionado. 1.9.1 Se muestra un menú con las diferentes opciones que se podrán realizar a partir del lugar

			<p>seleccionado.</p> <p>a) Opciones a realizar:</p> <p>i) Se muestra el listado con los grupos de lugares contenidos en el lugar seleccionado.</p> <ul style="list-style-type: none"> <li>• Si se selecciona un grupo de lugar:             <ol style="list-style-type: none"> <li>1. Se muestra el listado con los tipos de lugares del grupo seleccionado.</li> <li>2. Se muestra el listado de lugares existentes del tipo seleccionado.</li> </ol> </li> </ul> <p>b) Si se selecciona la opción "Gestionar inmueble":</p> <p>i) Mostrar el listado de inmuebles existentes en el lugar seleccionado:</p> <ul style="list-style-type: none"> <li>• Kilómetro</li> <li>• Número del inmueble</li> <li>• Código postal</li> <li>• Tipo de inmueble</li> <li>• Punto de referencia</li> </ul> <p>ii) Si se selecciona un inmueble:</p> <ul style="list-style-type: none"> <li>• Se muestra la dirección conformada con los datos del inmueble creado y el lugar donde se encuentra.</li> </ul>
--	--	--	--

2. Oprimir el botón Aceptar	2.1. Oprimir el botón Aceptar	Permite guardar la dirección conformada.	1. Si se selecciona la opción "Aceptar" se adiciona la dirección conformada.
3. Oprimir el botón Cancelar	3.1. Oprimir el botón Cancelar	Permite cancelar la operación de crear una nueva dirección.	1. Si se selecciona la opción "Cancelar" se cancela esta operación.

Tabla 21 Diseño de caso de prueba para el RF1 "Crear dirección"

#### 4.7.1 Pruebas unitarias

Las pruebas unitarias son una herramienta muy importante para el desarrollo del software con calidad. Aunque el tamaño de la "unidad" que se prueba puede variar, la Herramienta para pruebas Team System genera pruebas unitarias específicas para métodos, incluidos los métodos privados.

#### Prueba de Caja Blanca

La prueba de caja blanca se centra en la estructura interna del programa para elegir los casos de prueba. Esta prueba, también llamada caja de cristal, ocupa un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para realizar sus casos de prueba. (40)

Con el objetivo de realizar las pruebas de caja blanca a las funcionalidades del Módulo de Administración se implementaron las Pruebas Unitarias con todos los servicios para comprobar la validez del código y el acceso a datos. A continuación se muestran las pruebas realizadas al servicio "GetPrimaryTypes" (Ver Fig. 15 Prueba Unitaria realizada al servicio "GetPrimaryTypes") en la cual se comprobó la efectividad de este servicio (Ver Fig. 16 Resultado de la prueba unitaria al servicio "GetPrimaryTypes").

```

public void GetPrimaryTypesTest()
{
    AddressManagerComponent target = new AddressManagerComponent();
    string municipality = "BAHIA HONDA";
    string group = "LOCALIDAD";
    List<CompositeObject> expected = new List<CompositeObject> {new CompositeObject( "POBLADO",4)};
    List<CompositeObject> actual;
    actual = target.GetPrimaryTypes(municipality, group);
    Assert.AreEqual(expected.Count, actual.Count);
    for (int j = 0; j < actual.Count(); j++)
    {
        Assert.AreEqual(expected[j].Name, actual[j].Name);
        Assert.AreEqual(expected[j].Value, actual[j].Value);
    }
}
}

```

Fig. 15 Prueba Unitaria realizada al servicio "GetPrimaryTypes"

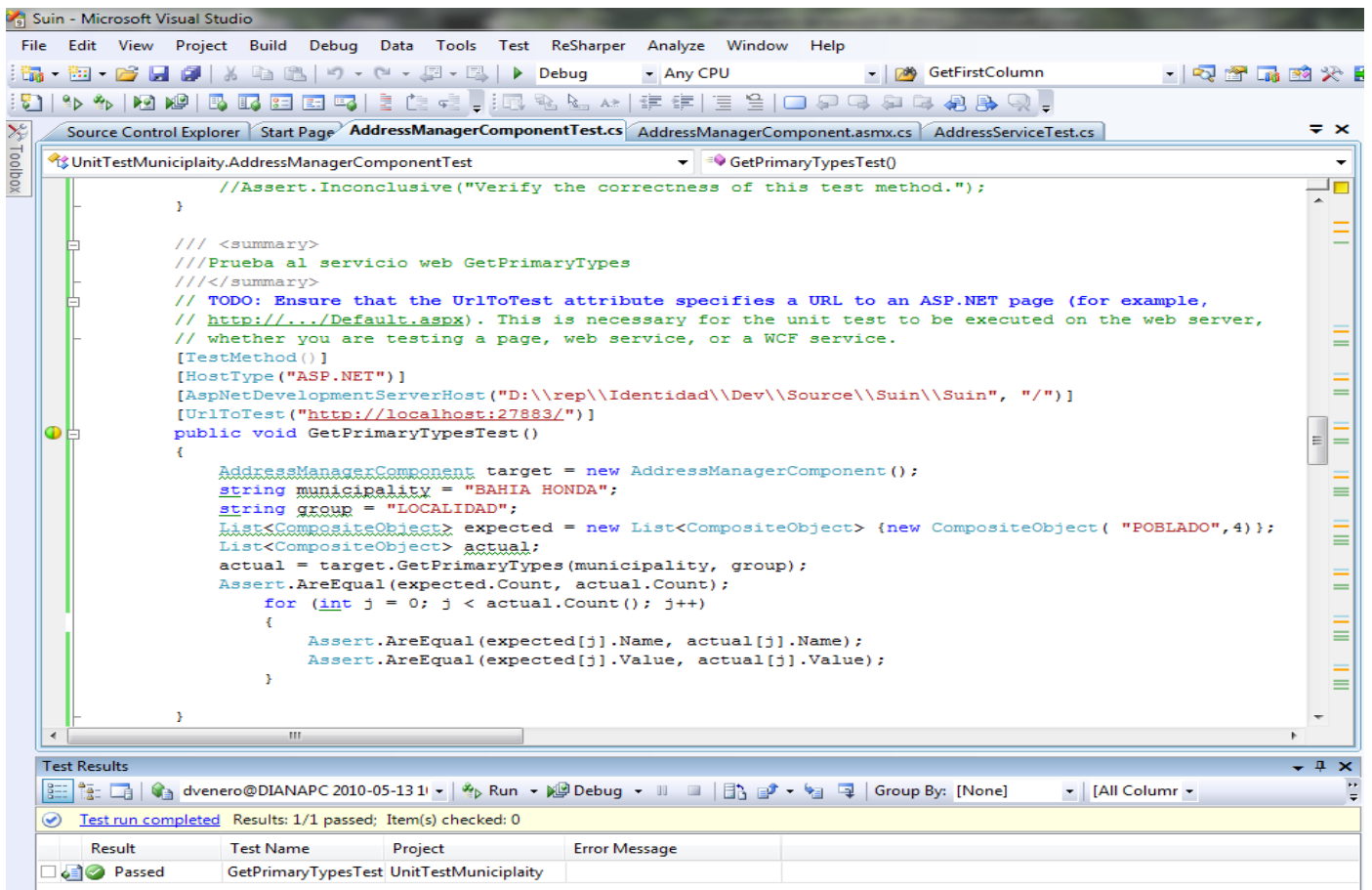


Fig. 16 Resultado de la prueba unitaria al servicio "GetPrimaryTypes"

**4.7.2 Pruebas de sistema**

Son las pruebas que se hacen cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados. Las pruebas de sistema principalmente se centran en verificar la interacción de los usuarios con el sistema, por lo que a menudo los casos de pruebas se obtienen a partir de las descripciones de los requisitos.

**Prueba de Caja Negra:**

Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida.

Con el propósito de realizarle las pruebas de caja negra al Módulo de Administración se utilizaron un conjunto de valores válidos e inválidos para encontrar posibles dificultades en las funcionalidades implementadas. (Ver **Tabla 22** Resultados de las pruebas de la primera iteración del RF1 “Crear dirección”)

**4.7.2.1 Resultados de las pruebas**

ID de Escenario	Tipos de Clases	Clases	Resultado Esperado	Resultado de la Prueba	Observaciones
1.1 Crear dirección	Clases Válidas	idlugar, UNIQUEID, no nulo	El sistema introduce idlugar.	Se introdujo el lugar seleccionado.	
		km, NUMERIC(5, 0), nulo	El sistema introduce el km correspondiente.	Se introdujo el Km correctamente.	
		codigopostal, NUMERIC(5, 0), nulo	El sistema introduce el código postal.	Se introdujo el código postal correctamente.	

		puntoreferencia, VARCHAR(100), nulo	El sistema introduce el punto de referencia.	Se introdujo el punto de referencia correctamente.	
		numero, CHAR(10), nulo	El sistema introduce el número de inmueble.	Se introdujo el número de Inmueble.	
		tipoinmueble, VARCHAR(20), no nulo	El sistema introduce el tipo de inmueble.	Se introdujo el tipo de inmueble.	
1.1	<b>Clases Inválidas</b>	km, VARCHAR(100), nulo	El sistema muestra un mensaje de error.	Muestra un mensaje de error.	
		codigopostal, VARCHAR(100), nulo	El sistema muestra un mensaje de error.	Muestra un mensaje de error.	
		puntoreferencia, NUMERIC(5,0), nulo	El sistema muestra un mensaje de error.	Muestra un mensaje de error.	
		numero, VARCHAR(100), nulo	El sistema muestra un mensaje de error.	Muestra un mensaje de error.	
		tipoinmueble, NUMERIC(5,0), no nulo	El sistema muestra un mensaje de error.	Muestra un mensaje de error.	

**Tabla 22** Resultados de las pruebas de la primera iteración del RF1 "Crear dirección"

Durante el proceso de pruebas a cada una de las interfaces del Módulo de Administración se desarrollaron tres iteraciones. En cada una de ellas el equipo de calidad del proyecto detectó las no conformidades existentes, las cuales fueron entregadas al equipo de desarrollo para ser solucionadas. (Ver **Fig. 17** Iteraciones de las pruebas de caja negra)



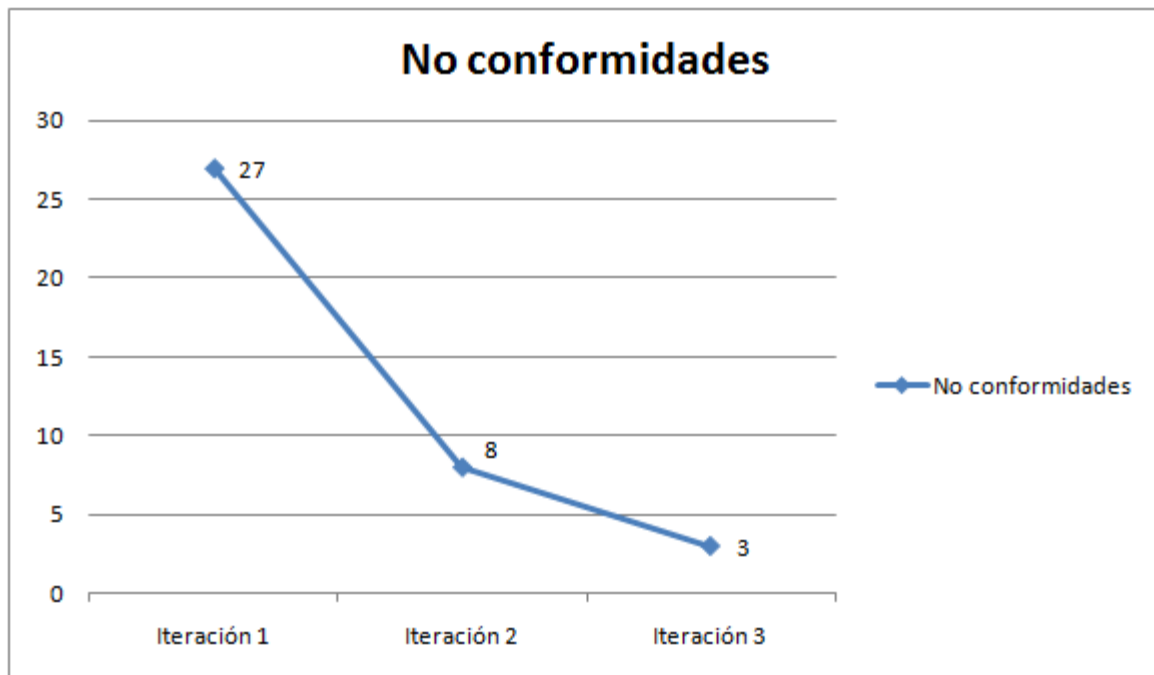


Fig. 17 Iteraciones de las pruebas de caja negra

#### 4.8 Conclusiones parciales

- En el presente capítulo se trataron temas referentes a los estándares de codificación y tratamiento de excepciones utilizados por el equipo de desarrollo del proyecto para el desarrollo de la aplicación con la calidad requerida.
- Se desarrolló una primera versión del Módulo de Administración, definiéndose los componentes que intervienen en la solución final de la aplicación así como la relación de los nodos físicos que los componen.
- Se mostraron las interfaces de la aplicación con las que interactúa el usuario final, a través de las cuales se desarrollaron las necesidades del cliente.
- Se realizaron las pruebas a la aplicación basándose en los diseños de casos de prueba definidos, donde se mostró la calidad del código y de la aplicación.

### CONCLUSIONES GENERALES

- Se realizó un estudio de aplicaciones que implementan un Módulo de Administración analizando conceptos fundamentales aplicados en el desarrollo del sistema como es la gestión de direcciones.
- Se definieron las herramientas utilizadas en el proceso de desarrollo de software después de un estudio de sus características y ventajas.
- Se llevó a cabo el análisis de los procesos de negocio actuales referente a la gestión de dirección en las oficinas del CIRP, viéndose la necesidad de la concepción de un Módulo de Administración que gestione los elementos de configuración.
- Se definieron los requerimientos funcionales que el sistema debe cumplir, así como la modelación de los procesos.
- Se generó una versión estable del modelo de datos, donde se define la persistencia de los datos asociados a la solución, con el objetivo de lograr una especificación completa y detallada de los requerimientos y una validación con el cliente.
- Se especificó y describió la arquitectura del sistema, la cual se basa en un paradigma arquitectónico de orientación a servicios, lo cual posibilita que si ocurren cambios en una capa no se afecten las demás.
- Se realizó el análisis y diseño de la solución definiéndose las clases y servicios necesarios, así como la relación entre ellos, para darle solución a todas las funcionalidades descritas.
- Se definieron los estándares de codificación y el tratamiento de excepciones a utilizar para el desarrollo de la aplicación con la calidad requerida.
- Se implementaron y probaron las funcionalidades planificadas para la primera iteración de desarrollo.

### RECOMENDACIONES

- Luego de haber desarrollado una primera versión del Módulo de Administración se recomienda hacer uso de la propuesta implementada, que integrada al SUIN, asegurará la gestión eficiente de los elementos de configuración.
- Continuar el desarrollo de la aplicación dándole seguimiento a los requerimientos funcionales que no han sido tratados en esta primera iteración.
- Definir mecanismos para la gestión y configuración de los elementos del SUIN que provean de mayor seguridad a los procesos de identificación.
- Incorporar nuevos elementos de georeferencia en la aplicación.
- Proponer la extensión de la solución de direcciones a otros órganos del MININT y del país una vez concluida la implementación y validada totalmente en la práctica.

## REFERENCIA BIBLIOGRÁFICA

1. **Juan Luis García Rambla.** microsoft.com. [En línea] [Citado el: 12 de Febrero de 2010.] <http://social.msdn.microsoft.com/search/es-es/?query=sistema de gestion de informacion>.
2. **José Alfredo Hernández Pérez.** Geothesis. [En línea] 1 de Agosto de 2008. [Citado el: 22 de Febrero de 2010.] [http://www.geothesis.com/index.php?option=com\\_content&view=article&id=479-mo-de-administracin-un-sistema-informco&catid=21artulos&Itemid=100.php.htm](http://www.geothesis.com/index.php?option=com_content&view=article&id=479-mo-de-administracin-un-sistema-informco&catid=21artulos&Itemid=100.php.htm).
3. **Universidad Autónoma de Colombia, Instituto Superior de Pedagogía.** [En línea] 2004. [Citado el: 20 de Enero de 2010.] <http://www.isp.fuac.edu.co.php.htm>.
4. **Marcos Guglielmetti.** MasterMagazine Archivo. [En línea] 2004. <http://www.mastermagazine.info/termino/7056.php>.
5. [En línea] <http://www.ibermatica.com/ibermatica/sit/situsuarios>.
6. ABCdatos. [En línea] 12 de 07 de 2005. [Citado el: 15 de Febrero de 2010.] <http://www.abcdatos.com/tutoriales/tutorial/16783.html>.
7. Iguanait. [En línea] [Citado el: 10 de Febrero de 2010.] <http://www.iguanait.com/tecnologia/iguana-java-framework/gestion-de-usuarios-y-permisos-de-acceso-al-sistema>.
8. Nobainfo. [En línea] Noba Informatica, S.L. [Citado el: 11 de Febrero de 2010.] <http://blog.nobainfo.com/22-gestion-de-roles-y-usuarios-en-openbravo/>.
9. Siainternacional. [En línea] Sistemas Informáticos Abiertos, S.A. . [Citado el: 28 de Enero de 2010.] [http://www.siainternacional.com/articles/06\\_S.htm](http://www.siainternacional.com/articles/06_S.htm).
10. *Diccionario Manual de la Lengua Española.* Cuba : Larousse Editorial, S.L., 2007.
11. [En línea] 30 de Mayo de 2006. [http://www.xombra.com/go\\_news.php?articulo=2371](http://www.xombra.com/go_news.php?articulo=2371).
12. **José Joaquín Mira, José M<sup>a</sup> Gómez, Inma Blaya, Alejandro García.** *La Gestión por Procesos.*
13. Microsoft. [En línea] 3 de Junio de 2006. [Citado el: 25 de Febrero de 2010.] [http://technet.microsoft.com/es-es/library/cc728199\(WS.10\).aspx](http://technet.microsoft.com/es-es/library/cc728199(WS.10).aspx).
14. **Proyecto infraestructura de datos especiales, Grupo Desarrollo de Software.** *Geocodificación en ARCGIS.*
15. **Dirección de estadísticas sociales, Oficina Nacional de estadísticas.** *Asentamientos humanos urbanos y rurales concentrados.* 2006.
16. **Martínez, Adrian H. Bravo.** *Propuesta de la Base Nacional del CIRP.* 2006.
17. —. *Propuesta de la Base Nacional del SACIRP.* 2006.
18. *oracle.com.* [En línea] [Citado el: 20 de Abril de 2010.] [http://www.oracle.com/global/es/database/docs/oracle\\_spatial\\_gis.pdf](http://www.oracle.com/global/es/database/docs/oracle_spatial_gis.pdf).
19. [En línea] <http://sipec.sep.gob.mx/WebHelp/reportes/reporte.htm>.
20. **DIR.** *Manual de Usuario del Sistema Automatizado de Certificación de Identidad.*
21. —. *Manual de Usuario del Sistema Automatizado de Atención a la Población.*
22. **SA., ALBET.** *Manual de Usuario de Administración del Sistema SAIME.* 2007. Versión 1.0.
23. **desarrollo, Grupo de.** *Documentos del Proyecto Identidad Cuba.* 2009.
24. [En línea] <http://danysoft.com/Altova-UModel>.

25. Milestone. [En línea] <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>.
26. **Johan Reymer Durand**. [En línea] 24 de Octubre de 2009. [Citado el: 24 de Febrero de 2010.] <http://jpmrd.wordpress.com/2009/10/24/oracle-database/>.
27. [En línea] Noviembre de 2007. [Citado el: 10 de Febrero de 2010.] <http://www.microsoft.com/teamsystem..>
28. Microsoft. [En línea] <http://www.microsoft.com/business/smb/es-mx/servidores-y-herramientas/visual-studio.msp>.
29. *Microsoft*. [En línea] [Citado el: 16 de Enero de 2010.] <http://msdn.microsoft.com/es-es/netframework/default.aspx>.
30. Microsoft. [En línea] [Citado el: 15 de Enero de 2010.] <http://www.microsoft.com/downloads/details.aspx>.
31. Microsoft. [En línea] [Citado el: 22 de Febrero de 2010.] <http://msdn.microsoft.com/es-es/library>.
32. [En línea] [Citado el: 10 de Febrero de 2010.] <http://www.windowworkflowoundation.eu/>.
33. **José A. Gonzáles Seco**. Programación. [En línea] <http://www.programacion.com/tutorial/csharp/3/>.
34. msdn.com. [En línea] <http://msdn.microsoft.com/es-es/library/bb399567.aspx>.
35. msdn.com. [En línea] Noviembre de 2007. [http://msdn.microsoft.com/es-es/library/bb397926\(VS.90\).aspx](http://msdn.microsoft.com/es-es/library/bb397926(VS.90).aspx).
36. Pautas de Interfaz . ID-REQ102-2009.
37. [En línea] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.
38. [En línea] <http://www.sparxsystems.com.ar/download/ayuda/index.html?deploymentdiagram.htm>.
39. [En línea] <http://www.fismat.umich.mx/~crivera/tesis/node6.html>.
40. **LEON, F**. *Ingeniería de Software*. 2006.

**GLOSARIO****A**

API: Una interfaz de programación de aplicaciones., 19

Autenticación: Acción del usuario que le permite identificarse unívocamente en un sistema informático., 9

Autorización: Es cuando se le otorgan permisos a un usuario para realizar ciertas operaciones en un sistema informático., 9

**B**

BCL: Es una biblioteca de clases de programación informática que constituye una parte fundamental de la Microsoft . NET Framework., 22

Buffer: Es un espacio de memoria, en el que se almacenan datos para evitar que el programa o recurso que los requiere, ya sea hardware o software, se quede en algún momento sin datos., 14

**C**

C#: Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET., 19

C++: Es un lenguaje de programación diseñado a mediados de los años 1980., 24

Compact Framework: Es una versión del .NET Framework que está diseñado para funcionar en Windows CE basado en móviles/dispositivos embebidos tales como PDAs, teléfonos móviles, los controladores de fábrica, set-top boxes, entre otros., 22

**G**

Geodatabase: Modelo que permite el almacenamiento físico de la información geográfica, ya sea en archivos dentro de un sistema de ficheros o en una colección de tablas en un Sistema Gestor de Base de Datos., 11

**J**

Java: Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90., 19

JScript: Es un lenguaje de secuencias de comandos moderno con una gran variedad de aplicaciones., 21

**P**

Password: Clave para poder acceder a los sistemas informáticos., 10

Plataformas grid: Tecnología que permite que los ordenadores compartan información, capacidad de procesamiento, aplicaciones y dispositivos totalmente heterogéneos., 20

Provisioning: Provisioning Manager utiliza políticas definidas por el usuario para seleccionar siempre los recursos adecuados para cada actividad de aprovisionamiento. Gracias a esto, los administradores se evitan el problema que supone buscar espacio disponible para el aprovisionamiento, lo que les permite dedicar más tiempo a cuestiones de mayor valor estratégico. Asimismo, el uso de una consola de administración central permite a los administradores supervisar el estado de los recursos de almacenamiento aprovisionados., 9

### U

UML: Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema., 19

### V

Visual Basic .NET: Es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre el framework .NET., 19

### W

WCF: Plataforma de mensajería, creada con el fin de permitir una programación rápida de sistemas distribuidos y el desarrollo de aplicaciones basadas en arquitecturas orientadas a servicios., 22

Windows CardSpace: es una tecnología que sale junto con el .NET Framework 3.0, provee a los usuarios la habilidad de manejar sus identidades digitales., 22

Workstream: Secuencia de actividades de trabajo., 18

WPF: Proporciona a los programadores un modelo de programación unificado con el que generar experiencias de cliente inteligentes de Windows, en las que se incorpora la interfaz de usuario, multimedia y documentos., 22

### X

XBOX: Xbox es la primera videoconsola de sobremesa producida por Microsoft en colaboración con Intel y Nvidia., 22

XNA: Conjunto de herramientas con un entorno de ejecución administrado que facilita el desarrollo de juegos de ordenador y de gestión., 22