

**Universidad de las Ciencias Informáticas**



**CommonCore: Paquete de servicios comunes para el Sistema de  
Identificación, Inmigración y Extranjería de Cuba.**

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

**Autor(es):**

Daniel Ramón Torres Miyares

Yanet Silva Fernández

**Tutor(es):**

MsC. Erik de la Vega García

Ing. Reisbel Machado Rosabal

**Ciudad de la Habana**

**Junio de 2011**

*Dios no te hubiera dado la capacidad de soñar sin darte también la posibilidad de convertir tus sueños en realidad.*

*Héctor Tassinari*

# Declaración de Autoría



## DECLARACIÓN DE AUTORÍA

Declaramos ser los únicos autores del trabajo titulado “CommonCore: Paquete de servicios comunes para el Sistema de Identificación, Inmigración y Extranjería de Cuba” y autorizamos a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo.

Y para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ de 2011.

\_\_\_\_\_  
Firma del Autor

**Daniel Ramón Torres Miyares**

\_\_\_\_\_  
Firma del Autor

**Yanet Silva Fernández**

\_\_\_\_\_  
Firma del Tutor

**MsC. Erik de la Vega García**

\_\_\_\_\_  
Firma del Tutor

**Ing. Reisbel Machado Rosabal**

## DATOS DE CONTACTO

**Daniel Ramón Torres Miyares**

Correo: [drtorres@estudiantes.uci.cu](mailto:drtorres@estudiantes.uci.cu)

Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba

**Yanet Silva Fernández**

Correo: [ysfernandez@estudiantes.uci.cu](mailto:ysfernandez@estudiantes.uci.cu)

Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba

**MsC. Erik de la Vega García**

Correo: [edelavega@uci.cu](mailto:edelavega@uci.cu)

Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba

**Ing. Reisbel Machado Rosabal**

Correo: [rmachado@uci.cu](mailto:rmachado@uci.cu)

Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba

## DEDICATORIA

### **Yanet:**

Cuando me pongo a pensar cómo debo comenzar estas líneas no encuentro las palabras justas, pero espero que les llegue a quienes más me importan, que aunque no sean de una escritora, me salen del alma.

Siempre tuve en mente graduarme y ser ingeniera, pero luego de alcanzar la meta te pones a pensar en todo lo que le ha costado a la familia. Estos cinco años he podido contar con mi mamita querida, con mi papá a quien quiero con el alma, con el pesado de mi hermanito, con mis abuelos y con el novio maravilloso que tengo, pero de quienes más me acuerdo son de aquellos que ya no están con uno, y quisiera que supieran en donde quiera que estén que los extraño mucho y que les dedico todo lo que soy: a mi abuela Hayde, mi abuelo Margarito y a mis bisabuelos mima y pipo. A mi mamita y a mi papito que me han apoyado en cada una de las decisiones que he tomado, a mi hermanito, a mis abuelos que me han dado todo el amor que tienen, a mi novio que es lo más lindo que me ha pasado en estos cinco años les dedico este trabajo que tanto sacrificio les ha costado, pues han sufrido cuando triste estaba, han reído cuando alegre me sentía, me han regañado cuando lo necesitaba y sobre todo me han apoyado siempre.

### **Daniel:**

Quisiera dedicar el presente trabajo a mi familia por haberme dado todo ese amor y apoyo en mi vida, y en especial:

A mi hermana y mi novia mis ángeles.

A mi madre y mi padre que todo lo que soy, se los debo a ellos.

## AGRADECIMIENTOS

### Yanet

Luego de cinco años estudiando, saber que culmina el camino me hace pensar en todos los que me han ayudado en la carrera y han dado su granito de arena para que me pudiera graduar con todo los conocimientos que tengo. Agradezco:

A la FEU, que me enseñó que bien puedes trabajar hasta las 7 de la mañana y tener una prueba ese mismo día, a los compañeros con los que trabaje por enseñarme que se puede trabajar y hacer amigos con los que puedes contar para siempre: Vismar, Ander, Aleli, Alexander, Randy, Yarismay, Yendry, Yidian, José Augusto, Yamile, Nilo, Carlo Carlo, Abelito, Carlos Arce, Jorgito, Yenier, Yoel e Idanis;

A los amigos que te hacen reír cuando más lo necesitas: Michel el gordo, Julio el punto y Denier el feo; A las nuevas amistades: Alejandro, Yirian, Ludmary, Dianelis, Idia, Daniel y Grethel;

A las viejas amistades: Lianet, Loida, Ibis, Elisandra y Editó;

A los que trabajan conmigo en el proyecto día a día: Dayana, Dairon, el Calvo, René, Arnaldo, Eduardo, Humberto, Reisbel, Manuel, Mavilio, Yani, Brayan, Arianna y Yuli;

A los profesores y no profesores que me han enseñado todo lo que sé: Erik, Boloy, Blanco, Adrian, Yudenia, Joel y Choni;

A Diosito por escucharme siempre que necesitaba desahogarme y por darnos salud y bienestar a mi familia y a mí;

A Fidel y al Che que con su ejemplo me han sabido guiar hacia el camino de un hombre nuevo, o en este caso una mujer nueva;

A los amigos del barrio que son familia: Lupe, Osmaida, Roberto, Yeni y Oreste;

Y sobre todo a mi familia querida: A mi mamita, a mi papito, a mi hermanito, a Norkis mi novio, a mis abuelos (Caridad, Hayde, Manuel y Margarito), a mis bisabuelos (Mima y Pipo); a mis suegros (Ramona y Norkis), mi cuñado Norbis, mis primos (Danay, Ismelis, Yanelis, Dani, Yuri, Luis Daniel, Raulito, Claudia) y a mis tíos (Lázaro, Mayda, Odalis, Mayra, Isac, Tita y Raúl).

## **Daniel:**

A los tutores, Reisbel por las habilidades que me ha transmitido como desarrollador, Erik quien con su perfeccionismo me ha enseñado a como ser un profesional, a Blanco y Adrian por sus consejos profesionales y su interés en nuestro trabajo y a Yude que nos ha ayudado en todo momento.

A mi compañera de tesis Yanet quien le da un toque especial a nuestra tesis.

A Hector y Mai que brindaron un apoyo fundamental en el desarrollo de esta tesis.

Al grupo de trabajo del proyecto que de una forma u otra han aportado en el desarrollo de esta tesis y se han comportado como una familia para mí.

A todos los profesores que han puesto empeño en mi superación profesional y personal.

A mi otra familia, mis suegros Armando e Idalmis por aceptarme como un hijo y a mi sobrinito coco.

A mi ángel Dunet por estar junto a mí brindándome apoyo, cariño y amor durante estos casi cuatro años.

Y por último a mi familia que les debo quien soy, mis abuelos Cesar y Esther que aunque no estén presentes siempre tendrán un pedazo de mi corazón, Mimi y Mario que siempre están cuando los necesito, a Maira por cuidar de mi padre, a Maylin y Albertico mis nuevos hermanos, a Alberto que me ha tratado como su hijo y no para de aportarme conocimientos, a mi tía Diana y a mi primita Yeni que están lejos pero es como si estuvieran cerca, a mi hermana Elena mi angelito, a mi padre por enseñarme que es la vida y como vivirla y a mi madre mi tutora incansable, mi guía, mi amor.

## RESUMEN

En los últimos años el Ministerio del Interior de la República de Cuba ha venido desarrollando sistemas para el buen funcionamiento de las Direcciones de Inmigración y Extranjería y de Identificación y Registro. Por ello, y en aras de una mejora tecnológica, el Ministerio del Interior en conjunto con la Universidad de las Ciencias Informáticas decide realizar el Sistema de Identificación, Inmigración y Extranjería de Cuba, el cual permite llevar a cabo las funciones de estas direcciones con la calidad requerida. Este sistema está compuesto por dos subsistemas: Sistema Único de Identificación Nacional y Sistema de Inmigración, Extranjería y Ciudadanía, los cuales cuentan con funcionalidades comunes.

El presente trabajo de diploma tiene como objetivo desarrollar un paquete de servicios que exponga funcionalidades comunes y reutilizables para el Sistema de Identificación, Inmigración y Extranjería de Cuba teniendo en cuenta su Arquitectura Orientada a Servicios. Para la realización del paquete se hizo un estudio de las Arquitecturas Orientada a Servicios, así como las tecnologías para implementar esta, de donde se determinó utilizar *Windows Communication Foundation* para exponer los servicios comunes del sistema y *Entity Framework* para el mapeo de la base de datos.

Este paquete de servicios posibilita el aumento del grado de reutilización de funcionalidades comunes mediante el encapsulamiento, permite a las capas superiores del sistema abstraerse de todo el trabajo de acceso a datos y de funcionalidades esenciales, facilitando que cualquier cambio en la implementación de los servicios sea transparente para las capas superiores.

### Palabras Claves:

SOA, *Windows Communication Foundation*, Paquete de Servicios Comunes, Reutilización.



## ÍNDICE

Datos de Contacto .....	I
Dedicatoria.....	II
Agradecimientos .....	III
Resumen .....	V
Índice .....	VI
Índice de Tablas .....	IX
Índice de Ilustraciones .....	XII
Introducción .....	1
Capítulo 1 Fundamentación Teórica .....	6
1.1. Introducción .....	6
1.2. Arquitectura Orientada a Servicios.....	6
1.2.1. Ventajas de la Arquitectura Orientada a Servicios.....	7
1.2.2. Facilitadores Tecnológicos claves de la Arquitectura Orientada a Servicios.....	9
1.3. Arquitectura definida para el SIIEC.....	11
1.4. Windows Communication Foundation (WCF).....	12
1.4.1. Características de WCF .....	14
1.4.2. Integración de WCF con otras Tecnologías de Microsoft.....	15
1.4.3. Consumo de Servicios Web .....	16
1.4.4. Seguridad en WCF.....	16
1.4.5. Ventajas de la Seguridad de WCF .....	17
1.4.6. Áreas funcionales de la seguridad de WCF.....	18
1.5. Arquitecturas de los Servicios.....	19
1.5.1. Arquitectura por Capas .....	20
1.5.2. Patrón de Arquitectura Modelo-Vista-Controlador .....	21
1.6. Capa de Acceso a Datos .....	22
1.6.1. TierDeveloper 6.1.....	22
1.6.2. NHibernate 3.0.....	23
1.6.3. ADO.NET Entity Framework.....	23
1.7. Ambiente de desarrollo .....	24

1.7.1.	Metodología de Desarrollo de Software.....	24
1.7.2.	Herramientas de Modelado .....	25
1.7.3.	Sistema Gestor de Base Datos .....	25
1.7.4.	Entorno de Desarrollo Integrado.....	27
1.7.5.	Control de Versiones .....	28
1.7.6.	Framework .....	28
1.8.	Conclusiones .....	30
Capítulo 2	Propuesta de Solución .....	31
2.1.	Introducción .....	31
2.2.	Presentación de la solución .....	31
2.3.	Modelo de dominio .....	32
2.4.	Descripción del Modelo de Dominio.....	33
2.5.	Requerimientos de los Servicios .....	34
2.5.1.	Descripción de los Roles .....	34
2.5.2.	Definición de Requisitos Funcionales .....	34
2.5.3.	Especificación de los requisitos funcionales .....	35
2.5.4.	Especificación de Requisitos no Funcionales .....	38
2.6.	Descripción de los Servicios Propuestos .....	40
2.7.	Conclusiones .....	41
Capítulo 3	Análisis y Diseño del Sistema .....	43
3.1.	Introducción .....	43
3.2.	Arquitectura de la solución.....	43
3.3.	Estructura .....	45
3.4.	Patrones utilizados .....	46
3.5.	Diagramas de las clases .....	48
3.5.1.	Descripción de las clases entidades.....	49
3.6.	Modelo de datos .....	49
3.7.	Descripción de entidades.....	50
3.8.	Conclusiones .....	51
Capítulo 4	Implementación y Prueba .....	52

4.1.	Introducción .....	52
4.2.	Estándares de Codificación .....	52
4.3.	Tratamiento de Errores .....	53
4.4.	Diagrama de despliegue .....	54
4.5.	Implementación .....	54
4.5.1.	Diagrama de componente .....	55
4.6.	Pruebas .....	56
4.6.1.	Pruebas unitarias .....	56
4.6.1.1.	Diseño de Casos de Pruebas .....	56
4.6.1.2.	Resultados de las pruebas .....	58
4.6.1.3.	Conclusiones Pruebas Unitarias .....	61
4.6.2.	Pruebas de Carga al Sistema.....	62
4.6.2.1.	Conclusiones Pruebas de Carga al Sistema.....	65
4.6.3.	Aceptación de los Servicios.....	66
4.7.	Reutilización de Servicios .....	66
4.8.	Beneficios del Paquete .....	68
4.9.	Conclusiones .....	68
	Conclusiones .....	69
	Recomendaciones .....	70
	Bibliografía.....	71
	Glosario de Términos.....	75
	Anexos.....	76
Anexo I.	Catálogo de Requisitos Funcionales.....	76
Anexo II.	Descripción de Requisitos Funcionales .....	78
Anexo III.	Fragmento del Modelo de Datos. ....	139
Anexo IV.	Descripción de Entidades de EncoderServices .....	142
Anexo V.	Diseño de Casos de Pruebas.....	147
Anexo VI.	Pruebas Unitarias.....	155
Anexo VII.	Pruebas de Aceptación .....	166

## ÍNDICE DE TABLAS

Tabla 1 Descripción de los roles .....	34
Tabla 2 Especificación del Servicio de Nomencladores. ....	38
Tabla 3 Descripción de la entidad NTIPOPROCESO.....	51
Tabla 4 Descripción de la Entidad NTIPOTRAMITE .....	51
Tabla 5 Diseño de caso de prueba para el servicio EncoderService .....	58
Tabla 6 Pruebas de Requisitos por Iteración del servicio EncoderService .....	61
Tabla 7 Pruebas realizadas a cada requisito funcional de los servicios por cada iteración del desarrollo ..	62
Tabla 8 Descripción del hardware de las máquinas de prueba. ....	63
Tabla 9 Resultados de la prueba de carga.....	64
Tabla 10 Resultados prueba de carga. ....	64
Tabla 11 Reutilización de servicios por módulos.....	67
Tabla 12 Mostrar los nomencladores existentes dado una clasificación de nomenclador. ....	81
Tabla 13 Contar los nomencladores existentes dado una clasificación de nomenclador.....	84
Tabla 14 Mostrar los nomencladores existentes dado una clasificación de nomenclador, un número de inicio y una cantidad de total.....	87
Tabla 15 Mostrar los datos de un nomenclador similar según parámetros relacionales. ....	90
Tabla 16 Contar los datos de un nomenclador similar a los parámetros de la búsqueda realizada. ....	93
Tabla 17 Obtener los datos de un nomenclador similar según parámetros, un número de inicio y una cantidad de total.....	97
Tabla 18 Insertar Persona Nacional.....	98
Tabla 19 Insertar Básico Persona Nacional. ....	99
Tabla 20 Insertar Persona Extranjera.....	100
Tabla 21 Insertar Básico Persona Extranjera. ....	101
Tabla 22 Insertar Persona no Registrada.....	103
Tabla 23 Insertar Básico Persona no Registrada. ....	104
Tabla 24 Modificar Persona Nacional.....	105
Tabla 25 Modificar Persona Extranjera. ....	106
Tabla 26 Modificar Persona no Registrada. ....	108
Tabla 27 Modificar Estado Fallecido de la Persona.....	108

Tabla 28 Modificar Dirección Permanente de la Persona.....	109
Tabla 29 Búsqueda fonética de una persona en los registros operativos según datos biográficos.....	111
Tabla 30 Buscar una persona en los registros operativos dado su número de identidad. ....	112
Tabla 31 Buscar una persona en los registros operativos dado un tipo de información.....	114
Tabla 32 Buscar listado de personas dada una persona semejante.....	115
Tabla 33 Buscar una persona según el id de la persona.....	115
Tabla 34 Buscar una persona según el número de identidad de la persona. ....	116
Tabla 35 Buscar personas nacionales similares a una persona dada. ....	118
Tabla 36 Buscar personas nacionales que tengan una dirección dada. ....	120
Tabla 37 Buscar persona nacional dado el identificador de persona.....	122
Tabla 38 Buscar persona nacional dado el número de identidad. ....	123
Tabla 39 Buscar personas extranjeras, dada una persona similar. ....	125
Tabla 40 Buscar persona extranjera dado un id de persona. ....	127
Tabla 41 Buscar persona extranjera dados el documento de identidad y una persona semejante.....	129
Tabla 42 Buscar personas no registradas dada una persona similar. ....	131
Tabla 43 Buscar persona no registrada dado el identificador de una persona. ....	132
Tabla 44 Buscar persona no registrada dado el número de identidad.....	134
Tabla 45 Obtener foto de una persona dado el identificador de la foto.....	134
Tabla 46 Comprobar si se puede realizar un proceso a una persona dada.....	135
Tabla 47 Adicionar un proceso dado a una persona. ....	136
Tabla 48 Buscar listado de direcciones dado el identificador de una persona.....	137
Tabla 49 Buscar listado de direcciones dado el ci de una persona. ....	138
Tabla 50 Buscar listado de direcciones dada una tira de dirección. ....	138
Tabla 51 Buscar listado de direcciones por parámetros.....	139
Tabla 52 Entidad DDIRECCION. ....	141
Tabla 53 Entidad DPROCESO.....	141
Tabla 54 Entidad DPERSONA.....	142
Tabla 55 Descripción de la entidad de EncoderServices NAREAAPLICACION .....	142
Tabla 56 Descripción de la entidad de EncoderServices NATRIBUTODPCUMENTO .....	143
Tabla 57 Descripción de la entidad de EncoderServices NCATEGORIAEXTRANJERO.....	143

Tabla 58 Descripción de la entidad de EncoderServices NCAUSACANCELARDENEGAR .....	144
Tabla 59 Descripción de la entidad de EncoderServices NREQUISITOTIPOTRAMITE .....	144
Tabla 60 Descripción de la entidad de EncoderServices NREQUISITOTRAMITE .....	144
Tabla 61 Descripción de la entidad de EncoderServices NTIPOAPLICACION .....	145
Tabla 62 Descripción de la entidad de EncoderServices NTIPODATO .....	145
Tabla 63 Descripción de la entidad de EncoderServices NTIPODOCIDENTIFICACION.....	146
Tabla 64 Descripción de la entidad de EncoderServices NTIPODOCUMENTO.....	146
Tabla 65 Descripción de la entidad de EncoderServices NTIPOMEDIDACNI .....	146
Tabla 66 Descripción de la entidad de EncoderServices NTIPOPROCESOBLOQUEO.....	147
Tabla 67 Descripción de la entidad de EncoderServices NTIPOPROCESOTIPOTRAMITE .....	147
Tabla 68 Diseño de caso de prueba para el servicio PersonManagerService .....	149
Tabla 69 Diseño de caso de prueba para el servicio SearchPersonService.....	152
Tabla 70 Diseño de caso de prueba para el servicio SearchAddressService .....	154
Tabla 71 Diseño de caso de prueba para el servicio OperativeRecordService .....	155
Tabla 72 Diseño de caso de prueba para el servicio ProcessRequestService .....	155
Tabla 73 Pruebas de Requisitos por Iteración del servicio SearchAddressService .....	156
Tabla 74 Pruebas de Requisitos por Iteración del servicio SearchPersonService .....	159
Tabla 75 Pruebas de Requisitos por Iteración del servicio PersonManagerService .....	161
Tabla 76 Pruebas de Requisitos por Iteración del servicio OperativeRecordService.....	163
Tabla 77 Pruebas de Requisitos por Iteración del servicio ProcessRequestService.....	164

## ÍNDICE DE ILUSTRACIONES

Ilustración 1 Arquitectura Orientada a Servicios desde el punto de vista de la tecnología.....	9
Ilustración 2 Vista lógica de la arquitectura del software. ....	11
Ilustración 3 Abstracción del sistema. ....	31
Ilustración 4 Modelo de dominio.....	33
Ilustración 5 Arquitectura de los servicios propuesta.....	44
Ilustración 6 Estructura de los servicios. ....	44
Ilustración 7 Diagrama de Paquetes para los servicios. ....	45
Ilustración 8 Diagrama de Clases. ....	48
Ilustración 9 Fragmento modelo de datos del EncoderServices. ....	50
Ilustración 10 Diagrama de Despliegue.....	54
Ilustración 11 Diagrama de Componentes ....	55
Ilustración 12 Prueba unitaria del EncoderService.....	59
Ilustración 13 Resultado final de las pruebas a EncoderService. ....	60
Ilustración 14 Gráfica de las pruebas de Requisitos por Iteración del servicio EncoderService.....	61
Ilustración 15 Gráfica de las pruebas realizadas a cada requisito funcional de los servicios por cada iteración del desarrollo.....	62
Ilustración 16 Leyenda gráficos.....	63
Ilustración 17 Gráfico de tiempo de respuesta de las funcionalidades respecto al tiempo de pruebas total. ....	64
Ilustración 18 Gráfico de tiempo de respuesta de las funcionalidades respecto al tiempo de pruebas total. ....	65
Ilustración 19 Gráfica del resultado de las Pruebas de Carga al Sistema ....	65
Ilustración 20 Gráfica de reutilización de servicios por módulos.....	67
Ilustración 21 Fragmento del Modelo de Datos del Paquete de Servicios. ....	140
Ilustración 22 Gráfica de las pruebas de Requisitos por Iteración del servicio SearchAddressService.....	156
Ilustración 23 Resultado final de las pruebas a SearchAddressService ....	157
Ilustración 24 Gráfica de las pruebas de Requisitos por Iteración del servicio SearchPersonService .....	159
Ilustración 25 Resultado final de las pruebas a SearchPersonService .....	160
Ilustración 26 Gráfica de las pruebas de Requisitos por Iteración del servicio PersonManagerService....	162

Ilustración 27 Resultado final de las pruebas a PersonManagerService .....	163
Ilustración 28 Gráfica de las pruebas de Requisitos por Iteración del servicio OperativeRecordService ..	164
Ilustración 29 Resultado final de las pruebas a OperativeRecordService.....	164
Ilustración 30 Gráfica de las pruebas de Requisitos por Iteración del servicio ProccessRequestService .	165
Ilustración 31 Resultado final de las pruebas a ProccessRequestService.....	165
Ilustración 32 Acta de liberación del subsistema SPDI.....	166



## INTRODUCCIÓN

La capacidad para responder rápidamente ante los cambios y optimizar los procesos de negocio es un factor clave para la competitividad y el crecimiento de las organizaciones. La agilidad de estas puede verse cuestionada si se apoya en entornos de Tecnologías de Información y Comunicación (TICs) que no pueden responder de forma flexible a los cambios que afectan a la actividad de negocio. Explotar el potencial que poseen las aplicaciones y recursos de las TICs y hacerlo disponible de forma general a toda la organización facilita la optimización de procesos y mejora la agilidad empresarial. (1)

Cuba apuesta hoy al desarrollo de la informática con el énfasis puesto en la comunidad y en el aumento de la producción estatal, en función de esto el Ministerio del Interior (MININT), en aras de lograr un mejor servicio, se dispone realizar la automatización de las áreas de la Dirección de Identificación y Registro (DIR) y la Dirección de Inmigración y Extranjería (DIE), las cuales tienen como principios llevar a cabo un control estricto de la identificación de la población nacional, así como controlar las normativas legales relacionadas con la migración, extranjería y ciudadanía en Cuba. Por la relevancia de estas direcciones se hace necesario incorporar una mayor rapidez y facilidad en la realización de los procesos que se realizan. (2) (3)

Con estos propósitos el Centro de Identificación y Seguridad Digital (CISED) de la Universidad de las Ciencias Informáticas (UCI) en conjunto con el MININT acuerdan crear el Sistema de Identificación, Inmigración y Extranjería de Cuba (SIIEC).

Por la importancia de este sistema, el impacto que tiene en la sociedad y en las decisiones del país en función de sus ciudadanos, se hace imprescindible desarrollar una solución basada en guías que permitan realizar una solución escalable, flexible, eficiente, segura y fácil de administrar, a estas guías se les ha denominado Arquitectura de Software, porque a semejanza de los planos de un edificio o construcción, indican la estructura, funcionamiento e interacción entre las partes del software. En el libro "*An introduction to Software Architecture*", David Garlan y Mary Shaw definen que la arquitectura es un nivel de diseño que hace foco en aspectos "más allá de los algoritmos y estructuras de datos de la computación; el diseño y especificación de la estructura global del sistema es un nuevo tipo de problema". (4)

La Arquitectura Orientada a Servicios (SOA, *Service Oriented Architecture*) es una filosofía de diseño que permite un mejor alineamiento de las TIC con las necesidades de las empresas, permitiendo a empleados,

clientes y socios comerciales responder de forma más rápida y adaptarse adecuadamente a las presiones del mercado. (1)

El sistema deberá ser desplegado en todas las provincias del país. El flujo de información que existe entre las más de 200 oficinas es muy grande pues se realizan aproximadamente 6000 trámites diarios, que pueden implicar la modificación de más de 6000 personas. Además, la solución que se propone debe soportar el tráfico de imágenes, las cuales consumen un tamaño considerable del ancho de banda disponible. Por las características del sistema y de la información que es gestionada es necesario que el sistema pueda ser accesible en todo momento y en caso de fallas estas deben ser corregidas en el menor tiempo posible. El tiempo de respuesta además debe ser rápido por ser un sistema de vital importancia para el país en aeropuertos y otras entidades que realicen trámites, esto provoca que se generen altos niveles de concurrencia debido a la cantidad de información que es gestionada por el sistema en tiempo real. (5)

Dadas estas necesidades se propuso una solución cliente-servidor, con una Arquitectura Orientada a Servicios representada por 5 capas lógicas: Presentación, Proceso-Servicios, Negocio, Acceso a Datos y Base de Datos. Estas capas cuentan con una independencia lógica entre ellas facilitando la reutilización de la aplicación y disminuyendo en tiempo y esfuerzo los posibles cambios a realizarle a una solución basada en la misma.

Debido a la magnitud de los procesos, el alto nivel de escalabilidad requerido, el volumen de funcionalidades comunes, la necesidad de que cualquier cambio tenga el menor impacto posible en el sistema, la poca experiencia de los desarrolladores para implementar una Arquitectura Orientada a Servicios y la necesidad de facilitar el desarrollo del sistema, se plantea el siguiente **problema científico**: ¿Cómo desarrollar el intercambio de información de los procesos comunes existentes en el Sistema de Identificación, Inmigración y Extranjería de Cuba de manera que permita su reutilización y teniendo en cuenta la Arquitectura Orientada a Servicios definida?

A partir de este problema se definió como **objeto de estudio** Las Arquitecturas Orientadas a Servicios y como **campo de acción** El intercambio de información de los procesos comunes en la Arquitectura Orientada a Servicios del Sistema de Identificación, Inmigración y Extranjería de Cuba.

Como **objetivo general** se propone: Desarrollar un paquete de servicios que exponga funcionalidades comunes y reutilizables para el Sistema de Identificación, Inmigración y Extranjería de Cuba teniendo en

cuenta su Arquitectura Orientada a Servicios. Para dar cumplimiento a dicho objetivo general se tienen los siguientes **objetivos específicos**:

1. Realizar un estudio sobre el marco teórico de la investigación.
2. Obtener los requisitos funcionales y no funcionales comunes para los servicios en cada una de las áreas.
3. Analizar y diseñar la arquitectura del núcleo del paquete de servicios.
4. Realizar la implementación del paquete de servicios comunes reutilizables para el Sistema de Identificación, Inmigración y Extranjería de Cuba.
5. Probar el paquete de servicios comunes para el Sistema de Identificación, Inmigración y Extranjería de Cuba.

Se propone como **hipótesis**: Si se desarrolla un paquete de servicios que exponga las funcionalidades comunes dentro de la Arquitectura Orientada a Servicios definida por el Sistema de Identificación, Inmigración y Extranjería de Cuba se logrará su uso reutilizable en los procesos existentes.

## Tareas de investigación

- Realizar un estudio bibliográfico de las Arquitecturas Orientadas a Servicios.
- Describir las herramientas y metodología a utilizar para el desarrollo de los servicios definidas por el proyecto.
- Definir los requisitos funcionales y no funcionales de los servicios.
- Desarrollar el análisis y diseño de cada uno de los servicios.
- Implementar el paquete de servicios comunes para el Sistema de Identificación, Inmigración y Extranjería de Cuba.
- Realizar pruebas unitarias y del sistema a los servicios comunes.

Los métodos de investigación que se han aplicado en la realización de este trabajo son:

## Métodos teóricos:

- **Histórico-Lógico:** Se empleó para hacer un análisis de la utilización de las arquitecturas basadas en servicios y el impacto de estas en las organizaciones.
- **Analítico-Sintético:** Se analiza toda la información recopilada a través de los diferentes medios bibliográficos que puedan servir para desarrollar mejor el diseño del núcleo de servicios.
- **Hipotético-Deductivo:** Se utiliza para luego de plantear la hipótesis y siguiendo la lógica de deducción llegar a nuevos conocimientos y predicciones, que son sometidos a verificaciones.
- **Modelación:** se presenta creando abstracciones de los servicios con el objetivo de explicar la realidad. El modelo como sustituto del objeto de investigación es semejante a él, existiendo una correspondencia objetiva entre el modelo y el objeto.

## Métodos empíricos:

- **Entrevista:** Se manifiesta a través de la realización de entrevistas a los proveedores de requisitos, además es necesario el intercambio con el personal capacitado para obtener información referente al tema, criterios y corregir malas concepciones para un mayor entendimiento del problema a resolver y determinar las necesidades del cliente.

El presente documento consta de cuatro capítulos, desarrollados a partir del estudio realizado de la información bibliográfica e institucional existente. La descripción de los mismos se presenta a continuación:

- **Capítulo 1: Fundamentación teórica:** cuenta con el respaldo teórico de los temas tratados en el trabajo de investigación, necesarios para el correcto entendimiento de la solución planteada. Se describen los conceptos fundamentales asociados al dominio del problema y al objeto de estudio realizándose un análisis de la situación actual. Se presenta el ambiente de desarrollo necesario para el análisis y la implementación de los servicios comunes.
- **Capítulo 2: Propuesta de solución:** se explican los conceptos existentes a través del modelo de dominio, se presenta la estructura de la arquitectura de los servicios y se exponen los requisitos funcionales y no funcionales de los servicios a desarrollar.
- **Capítulo 3: Diseño del sistema:** se realiza el modelo de diseño del sistema que se va a implementar, se definen y se especifican el conjunto de clases a utilizar. Además se describe la

arquitectura a utilizar, así como los patrones para el desarrollo del sistema y se presentan los servicios del mismo y el modelo de datos usado.

- **Capítulo 4: Implementación y Pruebas:** se realiza el desarrollo del paquete de servicios y para comprobar su correcto funcionamiento se le realizan las pruebas de caja blanca, unitarias y las de sistema, que permiten verificar la calidad de los servicios comunes. Se realiza un estudio para validar el nivel de reutilización del paquete.

## Capítulo 1 FUNDAMENTACIÓN TEÓRICA

### 1.1. INTRODUCCIÓN

En este capítulo se realiza un estudio sobre la Arquitectura Orientada a Servicios. Se profundiza en la arquitectura interna de los servicios, las tecnologías de acceso a datos; partiendo de los principales conceptos para lograr su entendimiento. Se analiza el ambiente de desarrollo propuesto caracterizando las herramientas y tecnologías que se analizaron para ser utilizadas en la confección de la solución.

### 1.2. ARQUITECTURA ORIENTADA A SERVICIOS

La Arquitectura Orientada a Servicios establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante Servicios Web, una tecnología basada en estándares e independiente de la plataforma, con la que la Arquitectura Orientada a Servicios puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular. (6)

¿Qué es un servicio exactamente? Un servicio es una funcionalidad concreta que puede ser descubierta en la red y que describe tanto lo que puede hacer como el modo de interactuar con ella. (7)

Desde la perspectiva de la empresa, un servicio realiza una tarea concreta: puede corresponder a un proceso de negocio tan sencillo como introducir o extraer un dato como “Carné de Identidad de una persona”. Pero también los servicios pueden acoplarse dentro de una aplicación completa que proporcione servicios de alto nivel, con un grado de complejidad muy superior, por ejemplo “Introducir datos de una persona”, un proceso que, desde que comienza hasta que termina, puede involucrar varias aplicaciones de negocio. (8)

La estrategia de orientación a servicios permite la creación de servicios y aplicaciones compuestas que pueden existir con independencia de las tecnologías subyacentes. En lugar de exigir que todos los datos y lógica de negocio residan en un mismo ordenador, el modelo de servicios facilita el acceso y consumo de los recursos informáticos a través de la red. Puesto que los servicios están diseñados para ser independientes, autónomos y para interconectarse adecuadamente, pueden combinarse y recombinarse con suma facilidad en aplicaciones complejas que respondan a las necesidades de cada momento en el

seno de una organización. Las aplicaciones compuestas (también llamadas “dinámicas”) son las que permiten a las empresas mejorar y automatizar sus procesos manuales, disponer de una visión consistente de sus clientes y socios comerciales y orquestar sus procesos de negocio para que cumplan con las regulaciones legales y políticas internas. El resultado final es que las organizaciones que adoptan la orientación a servicios pueden crear y reutilizar servicios y aplicaciones, y adaptarlos ante los cambios evolutivos que se producen dentro y fuera de ellas, y con ello adquirir la agilidad necesaria para ganar ventaja competitiva. (8)

## 1.2.1. VENTAJAS DE LA ARQUITECTURA ORIENTADA A SERVICIOS

### Arquitectura Orientada a Servicios desde el punto de vista del negocio

La Arquitectura Orientada a Servicios, desde el punto de vista del negocio, ayuda a resolver los siguientes requerimientos, largamente reclamados por el área de negocio:

- Mejorar la flexibilidad y agilidad de los sistemas.
- Proporcionar una visión integrada de los distintos pilares de la organización.
- Mejorar la cobertura de las necesidades del negocio.
- Reducir el impacto de la evolución de la tecnología en las aplicaciones de negocio. (6)

### Arquitectura Orientada a Servicios desde el punto de vista de la tecnología

La Arquitectura Orientada a Servicios, desde el punto de vista tecnológico, es el resultado de la constante evolución hacia un mayor desacoplamiento de las capas de una aplicación (presentación, orquestación de procesos y servicios de negocio) y a un mayor nivel de estandarización – interoperabilidad de cada una de estas capas.

Con un total desacoplamiento de las capas de las aplicaciones y un muy alto nivel de interoperabilidad, los beneficios desde el punto de vista tecnológico son claros: (9)

#### Favorece la reutilización:

- Aumenta el grado de reutilización al desacoplar las capas de una aplicación.
- Permite reutilizar las aplicaciones existentes mediante la encapsulación en servicios.
- Permite la utilización de servicios de terceros.

- Permite reaprovechar las plataformas existentes.

## **Aumenta la flexibilidad:**

- Simplifica la adaptación de los sistemas existentes.
- Evita el desarrollo de interfaces punto a punto entre los sistemas.
- Aumenta la interoperabilidad entre sistemas, permitiendo tanto la externalización como la prestación de servicios.

## **Mejora la productividad de los procesos:**

- Aumenta el nivel de automatización de los procesos, reduciendo el número de actividades manuales.
- Permite monitorizar la actividad del negocio (cuadros de mando).
- Permite realizar un análisis estadístico de los flujos de negocio reales en base a indicadores clave de negocio, permitiendo la identificación de puntos de mejora a optimizar.
- Permite evaluar el impacto y beneficio de variantes en los procesos mediante simulación.

## **Mejora el proceso de construcción de software:**

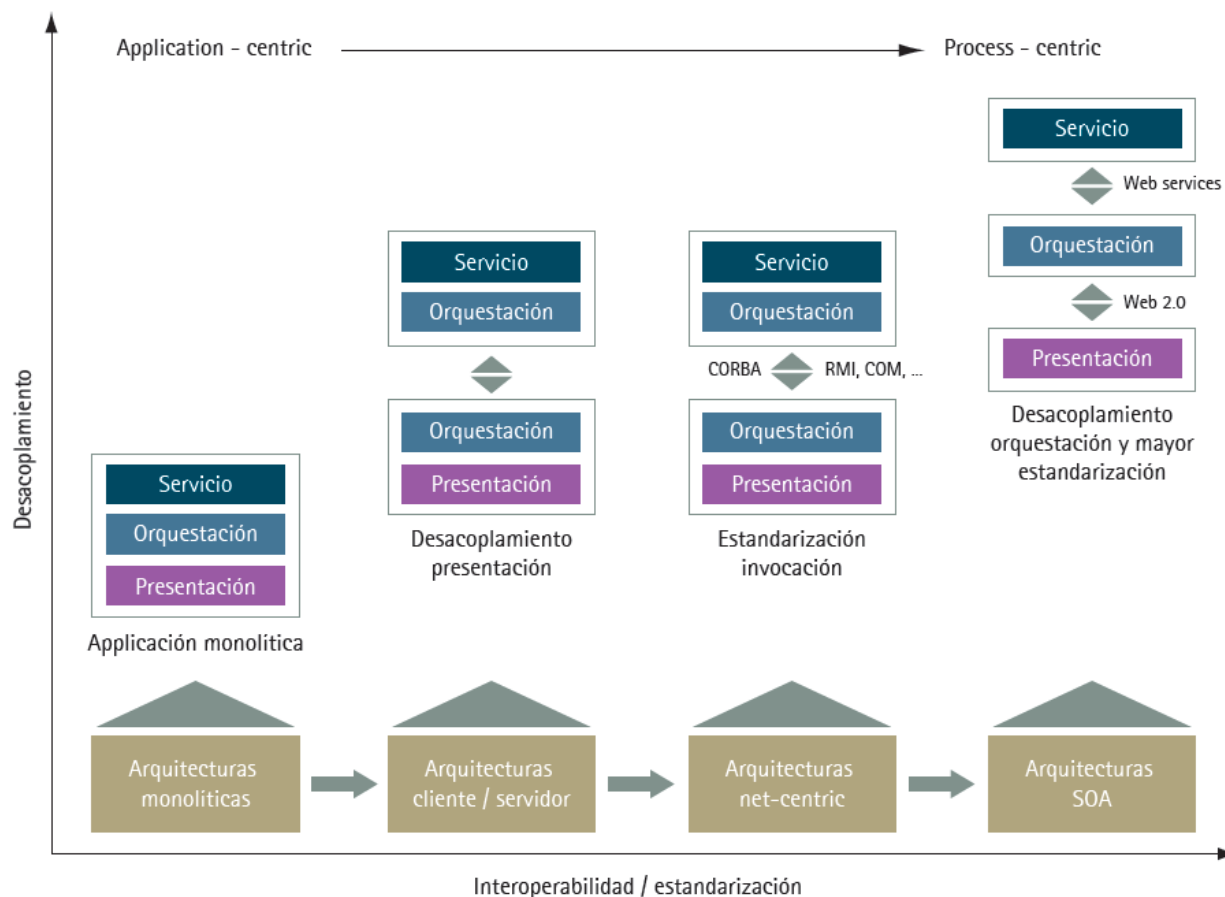
- Favorece la industrialización.
- Mejora la especificación de los requerimientos de negocio.
- Proporciona una filosofía de desarrollo común a todos los negocios y canales.
- Mejora la calidad.
- Desacopla el desarrollo de servicios y de procesos.
- Mejora el mantenimiento (procesos autodocumentados).

## **Mejora la usabilidad de las aplicaciones:**

- Permite presentar al usuario la información dispersa en distintos sistemas y de forma integrada.
- Permite alcanzar un mayor nivel de automatismo en las aplicaciones en procesos complejos de *workflow*.



- Permite utilizar tecnologías de presentación avanzadas como Web 2.0. (6)



**Ilustración 1 Arquitectura Orientada a Servicios desde el punto de vista de la tecnología. (6)**

## 1.2.2. FACILITADORES TECNOLÓGICOS CLAVES DE LA ARQUITECTURA ORIENTADA A SERVICIOS

Los facilitadores tecnológicos principales que permiten, desde el punto de vista tecnológico, la implantación de la Arquitectura Orientada a Servicios. No es imprescindible el uso de todos los facilitadores, pero cada uno de ellos es importante para alcanzar plenamente todos los beneficios esperados. (6)

### **Business Process Management (BPM)**

Gestión de Procesos de Negocio, es una disciplina de gestión que combina una visión centrada en procesos y de integración de funcionalidades que pretende mejorar la efectividad de las organizaciones.

Una solución BPM dispone de los medios necesarios para la realización efectiva de estos procesos así como las funcionalidades necesarias para que los gestores de las empresas puedan controlar y modificar los flujos de trabajo (“*workflows*”) tanto manuales como automáticos. (10)

## **La tecnología de Servicios Web (WS)**

Los Servicios Web son aplicaciones que utilizan estándares para el transporte, codificación y protocolo de intercambio de información. Los Servicios Web permiten la intercomunicación entre sistemas de cualquier plataforma y se utilizan en una gran variedad de escenarios de integración, tanto dentro de las organizaciones como con socios de negocios.

Los Servicios Web se basan en un conjunto de estándares de comunicación, como son XML<sup>1</sup> (*Extended Markup Language*) para la representación de datos, SOAP<sup>2</sup> (*Simple Object Access Protocol*) el intercambio de datos y el lenguaje WSDL<sup>3</sup> (*Web Services Description Language*) para describir las funcionalidades de un servicio web. Existen más especificaciones, a las que se denomina genéricamente como la arquitectura WS-\*, que definen distintas funcionalidades para el descubrimiento de Servicios Web, gestión de eventos, archivos adjuntos, seguridad, gestión y fiabilidad en el intercambio de mensajes y transacciones. (6)

## **Enterprise Service Bus (ESB)**

Un *Enterprise Service Bus* se encuentra basado en estándares que permite la comunicación síncrona o asíncrona entre aplicaciones heterogéneas. Un ESB es considerado la columna vertebral de una Arquitectura Orientada a Servicios, ya que conecta aplicaciones, recursos y componentes, garantizando características claves en este tipo de arquitectura como la confiabilidad y seguridad.

Un ESB o Bus de servicios empresariales, es un software que actúa como intermediario, entre la comunicación de aplicaciones de diferentes sistemas. Es decir es un punto central donde se registran todos los servicios expuestos por todas las aplicaciones de un entorno empresarial (sin importar las plataformas que lo componen) y sobre el cual se puede construir aplicaciones que re-aprovechen todas estas funcionalidades. (11)

---

<sup>1</sup> En español Lenguaje de Marcas Extensible.

<sup>2</sup> Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML (44)

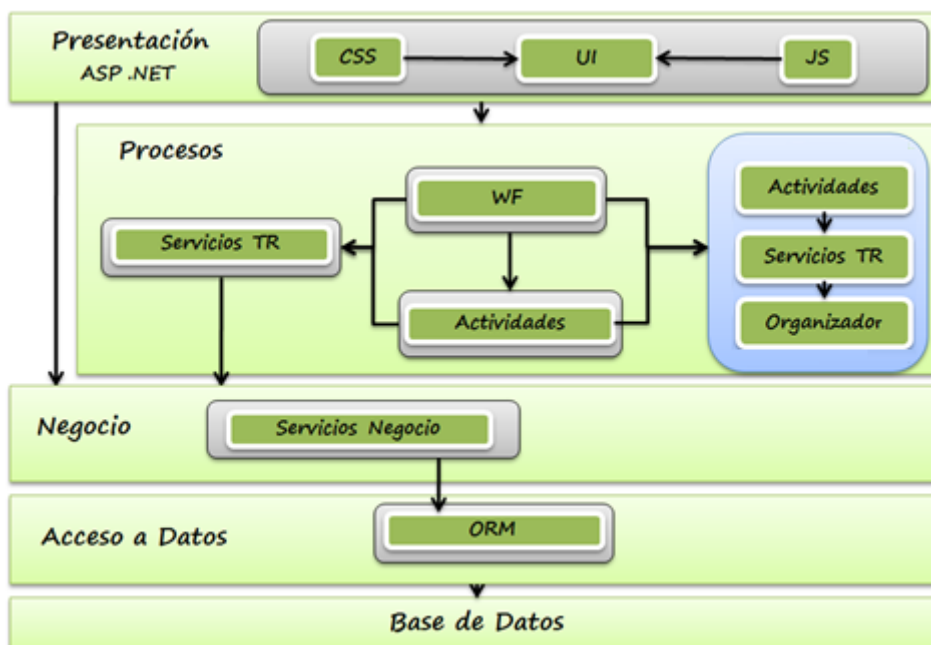
<sup>3</sup> En español Lenguaje de Descripción de Servicios Web.

## **Business Activity Monitoring (BAM)**

El objetivo del *Business Activity Monitoring* es proporcionar información en tiempo real sobre el estado y los resultados de varias operaciones, procesos y transacciones. Los principales beneficios del BAM se centran en ayudar a tomar mejores decisiones de negocios informadas, de forma rápida abordar las áreas problemáticas, y las organizaciones de la posición de volver a sacar el máximo provecho de las oportunidades emergentes. (12) (13)

### **1.3. ARQUITECTURA DEFINIDA PARA EL SIIEC**

La arquitectura del sistema está conformada por cinco capas fundamentales, orientadas al desarrollo de soluciones sobre la arquitectura cliente-servidor. Las capas que la conforman cuentan con una independencia lógica entre ellas facilitando la reutilización de la aplicación y disminuyendo en tiempo y esfuerzo los posibles cambios a realizarle a una solución basada en la misma. (5)



**Ilustración 2 Vista lógica de la arquitectura del software. (5)**

**Capa de Presentación (*Presentation Layer*):** Es la interfaz de comunicación de la aplicación con un usuario determinado. 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 Web, Interfaces de Usuario, etc. Tiene interacción directa con la capas de Procesos y con la de Negocio.

**Capa de Procesos (*Process Layer*):** Tiene contenida la lógica de los procesos de negocio, las actividades que por su nivel de reutilización o importancia lógica se encapsulan en una actividad propia del proyecto y los servicios de tiempo real, los cuales son los encargados de la interacción con los *workflows* definidos, estos a su vez interactúan con los servicios de la capa de negocio haciendo uso para ello de una fábrica de servicios que le devuelve la instancia requerida. Al mismo nivel se encuentra definida una fábrica para los servicios de tiempo real, que le permite a la capa de Presentación interactuar con estos sin necesidad de que se sepa la instancia del servicio que utilizan.

**Capa de Negocio (*Business Layer*):** Está conformado por un conjunto de servicios de negocio que realizan las acciones de negocio que no pueden ser definidas en la capa de procesos. Además mantienen separadas las acciones atómicas del negocio de la definición del proceso. Los principales servicios son la base fundamental de todo el negocio del sistema, controlan y gestionan la información, dan soporte, seguridad de acceso, flexibilidad de cambio, etc.

**Capa de Acceso a Datos (*Data Access Layer*):** Es el componente que da soporte a las funcionalidades de la capa de Negocio que se encuentran relacionadas con una fuente de datos. Se encuentra dividida en varias capas en sí misma para lograr una separación entre la gestión de las instancias de proceso y las entidades de negocio.

**Capa de Base de Datos (*Data Base Layer*):** Está constituida por todo el conjunto de tablas y procedimientos que permiten el almacenamiento de la información recolectada y procesada por los procesos. Se encuentra dividida en varias vistas que separan la gestión de los procesos de los datos de negocio de la persona y la información de soporte. La única capa encargada de interactuar con esta es la de Acceso a Datos.

## 1.4. WINDOWS COMMUNICATION FOUNDATION (WCF)

*Windows Communication Foundation* es la tecnología de Servicios Web de nueva generación de *Microsoft*, que facilita la interconexión entre sistemas y aplicaciones dentro de la organización y a lo largo de infraestructuras geográficamente dispersas. Es el primer modelo de programación creado de principio a fin para facilitar el desarrollo de aplicaciones orientadas a servicios. WCF eleva los Servicios Web a un

nivel superior, al ofrecer al desarrollador un entorno de alta productividad para la creación de aplicaciones basadas en servicios Web seguros y fiables, capaces de interactuar entre plataformas. Los desarrolladores ya pueden centrar sus esfuerzos en la lógica de negocio y delegar todos los aspectos de bajo nivel en WCF. Además de proporcionar un modelo de programación para el desarrollo de Servicios Web, WCF incluye una serie de herramientas y funcionalidades de gestión que facilitan a los profesionales de informática y tecnología la labor de crear, poner en marcha, configurar y monitorizar los Servicios Web. (14) (15)

Con WCF, es posible enviar datos como mensajes asincrónicos de un extremo de servicio a otro. Un extremo de servicio puede formar parte de un servicio disponible hospedado por IIS (*Internet Information Services*), o puede ser un servicio hospedado en una aplicación. Un extremo puede ser un cliente de un servicio que solicita datos de un extremo de servicio. Los mensajes pueden ser tan simples como un carácter o una palabra que se envía como XML, o tan complejos como una secuencia de datos binarios. A continuación se indican algunos escenarios de ejemplo: (14)

- Un servicio seguro para procesar transacciones comerciales.
- Un servicio que proporciona datos actualizados a otras personas, como un informe sobre tráfico u otro servicio de supervisión.
- Un servicio de chat que permite a dos personas comunicarse o intercambiar datos en tiempo real.
- Una aplicación de panel que sondea los datos de uno o varios servicios y los muestra en una presentación lógica.
- Exponer un flujo de trabajo implementado utilizando *Windows Workflow Foundation* como un servicio WCF.
- Una aplicación de Silverlight<sup>4</sup> para sondear un servicio en busca de las fuentes de datos más recientes.

En resumen, WCF se ha diseñado para ofrecer un enfoque manejable para la creación de Servicios Web y clientes de estos. (16)

---

<sup>4</sup> **Microsoft Silverlight** es una estructura para aplicaciones web que agrega nuevas funciones multimedia como la reproducción de vídeos, gráficos vectoriales, animaciones e interactividad, en forma similar a lo que hace Adobe Flash.

## 1.4.1. CARACTERÍSTICAS DE WCF

WCF incluye el siguiente conjunto de características:

- **Varios modelos de mensajes:** Los mensajes se intercambian mediante uno de los distintos modelos. El más común es el de solicitud-respuesta, en que un extremo solicita datos de otro extremo y el otro extremo responde. Existen otros modelos, como un mensaje unidireccional, en que un único extremo envía un mensaje sin esperar ninguna respuesta. Un modelo más complejo es el modelo de intercambio de mensajes dúplex, en donde dos extremos establecen una conexión y se envían datos entre sí, como ocurre con un programa de mensajería instantánea. (16)
- **Metadatos de servicios:** WCF admite la publicación de metadatos de servicios utilizando los formatos especificados en los estándares de la industria, como WSDL, Esquemas XML y *WS – Policy*. Estos metadatos pueden utilizarse para generar y configurar automáticamente clientes para el acceso a los servicios de WCF. Los metadatos se pueden publicar sobre HTTP<sup>5</sup> (*HyperText Transfer Protocol*) y HTTPS<sup>6</sup> (*HyperText Transfer Protocol Secure*), o utilizando el estándar Intercambio de metadatos de Servicios Web. (16)
- **Contratos de datos:** Dado que WCF se basa en *.NET framework*, también incluye métodos con código sencillo para proporcionar los contratos que desea aplicar. Uno de los tipos de contrato universales es el contrato de datos. Básicamente, mientras se escribe el código del servicio utilizando *Visual C#* o *Visual Basic*, la forma más sencilla de controlar los datos consiste en crear clases que representan una entidad de datos con propiedades que pertenecen a la misma. WCF incluye un completo sistema para trabajar con los datos de esta manera tan sencilla. Cuando se han creado las clases que representan los datos, el servicio genera automáticamente los metadatos que permiten a los clientes ajustarse a los tipos de datos que se han diseñado. (16)
- **Varios transportes y codificaciones:** Los mensajes pueden enviarse con cualquiera de los protocolos y codificaciones integrados. La combinación más frecuente de protocolo y codificación consiste en enviar mensajes SOAP codificados de texto utilizando HTTP usado en WWW (*World Wide Web*). WCF también le permite enviar mensajes sobre TCP, canalizaciones con nombre o

<sup>5</sup> En español Protocolo de Transferencia de Hipertexto.

<sup>6</sup> En español Protocolo Seguro de Transferencia de Hipertexto.

MSMQ<sup>7</sup> (*Message Queue de Microsoft*). Estos mensajes pueden codificarse como texto o utilizando un formato binario optimizado. Los datos binarios pueden enviarse de manera eficaz utilizando el estándar MTOM<sup>8</sup> (*Message Transmission Optimization Mechanism*). Si ninguno de los transportes o codificaciones proporcionados satisface sus necesidades, puede crear uno personalizado. (16)

- **Mensajes confiables y en cola:** WCF admite el intercambio de mensajes confiable utilizando sesiones confiables implementadas sobre *WS – Reliable Messaging* y utilizando MSMQ (16)
- **Mensajes duraderos:** Un mensaje duradero es aquel que nunca se pierde debido a una interrupción de la comunicación. Los mensajes que forman parte de un modelo de mensajes duraderos siempre se guardan en una base de datos. Si se produce una interrupción, la base de datos le permite reanudar el intercambio de mensajes cuando se restablezca la conexión. También puede crear un mensaje duradero utilizando *Windows Workflow Foundation (WF)*. (16)
- **Transacciones:** WCF también admite las transacciones que usan uno de los tres modelos de transacción: las transacciones *WS – Atomic*, las API del espacio de nombres *System.Transactions* y *Microsoft DTC* (Coordinador de transacciones distribuidas). (16)
- **Compatibilidad con AJAX y REST:** REST es un ejemplo de una tecnología de web 2.0 en evolución. WCF se puede configurar para procesar datos XML "sin formato" que no se ajustan en un sobre SOAP. WCF también se puede extender para admitir formatos XML concretos, como ATOM (un estándar popular de RSS) e incluso formatos que no sean XML, como *JavaScript Object Notation (JSON)*. (16)
- **Extensibilidad:** La arquitectura de WCF tiene varios puntos de extensibilidad. Si se necesita una función adicional, existen una serie de puntos que le permiten personalizar el comportamiento de un servicio. (16)

## 1.4.2. INTEGRACIÓN DE WCF CON OTRAS TECNOLOGÍAS DE MICROSOFT

WCF es una plataforma flexible, debido a esta flexibilidad extrema también se usa en otros productos de *Microsoft*. La primera tecnología en adaptarse a WCF fue *Windows Workflow Foundation (WF)*. Los flujos

---

<sup>7</sup> Herramienta de Microsoft para realizar el manejo de colas de mensajes. (43)

<sup>8</sup> En español Mecanismo de Optimización de Transmisión del Mensaje.

de trabajo simplifican el desarrollo de aplicaciones encapsulando los pasos del flujo de trabajo como “actividades”. En la primera versión de *Windows Workflow Foundation*, un desarrollador tenía que crear un host para el flujo de trabajo. La versión siguiente de *Windows Workflow Foundation* se integró con WCF. Esto permitió hospedar cualquier flujo de trabajo fácilmente en un servicio de WCF; puede hacer esto si elige automáticamente el tipo de proyecto WF – WCF en *Visual Studio 2010*. (16)

### 1.4.3. CONSUMO DE SERVICIOS WEB

Además de ofrecer una plataforma de desarrollo de primer nivel para la creación de Servicios Web, *Microsoft* ha introducido el soporte para Servicios Web en toda su línea de productos, permitiendo que sus principales funcionalidades puedan exponerse en forma de Servicios Web. Tanto los productos de *Microsoft* como de otros proveedores pueden consumir estos servicios para permitir una mayor conectividad y mejores experiencias de usuario.

### 1.4.4. SEGURIDAD EN WCF

*Windows Communication Foundation* es una plataforma de programación distribuida basada en mensajes SOAP, y la seguridad de los mensajes entre clientes y servicios es esencial para la protección de los datos. WCF proporciona una plataforma versátil e interoperable para el intercambio seguro de mensajes basada en la infraestructura de seguridad existente, y en las normas de seguridad reconocidas para los mensajes SOAP. (17)

WCF utiliza conceptos que resultan familiares si se han compilado aplicaciones seguras y distribuidas con las tecnologías existentes, por ejemplo, HTTPS, seguridad integrada de Windows, o nombres de usuario y contraseñas para autenticar a los usuarios. WCF no solo se integra en infraestructuras de seguridad existentes, sino que también extiende la seguridad distribuida más allá de los dominios solo de Windows mediante mensajes SOAP seguros. WCF puede considerarse una implementación de los mecanismos de seguridad existentes, con la ventaja principal de utilizar SOAP como protocolo, además de los protocolos existentes. Por ejemplo, las credenciales que identifican un cliente o un servicio, como el nombre de usuario y la contraseña, o los certificados X.509, poseen perfiles SOAP interoperables basados en XML. Con estos perfiles, los mensajes se intercambian de manera segura beneficiándose de especificaciones abiertas como las firmas digitales XML y el cifrado XML. (18)



## 1.4.5. VENTAJAS DE LA SEGURIDAD DE WCF

WCF es una plataforma de programación distribuida basada en mensajes SOAP. Con WCF, puede crear aplicaciones que funcionen como servicios y como clientes del servicio, creando y procesando mensajes a partir de un número ilimitado de otros servicios y clientes. En este tipo de aplicación distribuida, los mensajes pueden fluir de nodo a nodo, a través de *firewalls*, hacia Internet, y a través de los numerosos intermediarios SOAP. Esto introduce una gran variedad de amenazas del modo de seguridad. Los ejemplos siguientes muestran algunas amenazas comunes que la seguridad WCF puede ayudar a mitigar durante el intercambio de mensajes entre las entidades: (18)

- Observación del tráfico de red para obtener información confidencial. Por ejemplo, en un escenario de banca electrónica, un cliente solicita la transferencia de fondos desde una cuenta a otra. Un usuario malintencionado intercepta el mensaje y, como tiene el número de cuenta y la contraseña, más tarde realiza una transferencia de fondos desde la cuenta expuesta al peligro.
- Entidades deshonestas que actúan como servicios sin conocimiento del cliente. En este escenario, un usuario malintencionado (el deshonesto) actúa como servicio en línea e intercepta los mensajes del cliente para obtener información confidencial. A continuación, el usuario deshonesto utiliza los datos robados para transferir fondos de la cuenta expuesta al peligro. Este ataque también se conoce como ataque de suplantación de identidad (término en inglés *phishing*).
- Alteración de mensajes para obtener un resultado diferente al previsto por el autor de la llamada. Por ejemplo, modificar el número de cuenta en el que se realiza un depósito, desvía los fondos a una cuenta deshonestas.
- Una acción por la que un molesto pirata informático reproduce el mismo pedido de compra. Por ejemplo, una librería en línea recibe centenares de pedidos y envía los libros a un cliente que no los ha pedido.
- Incapacidad de un servicio para autenticar un cliente. En este caso, el servicio no puede garantizar que el rol adecuado realizó la transacción.

Resumiendo, la seguridad de la transferencia proporciona las siguientes garantías:

- Autenticación del extremo de servicio (autor de la respuesta).

- Autenticación de la entidad de seguridad de cliente (iniciador).
- Integridad del mensaje.
- Confidencialidad del mensaje.
- Detección de la reproducción.

## 1.4.6. ÁREAS FUNCIONALES DE LA SEGURIDAD DE WCF

La seguridad WCF está dividida en tres áreas funcionales: la seguridad de la transferencia, el control de acceso y la auditoría.

**Seguridad de la transferencia:** La seguridad de la transferencia abarca tres funciones de seguridad principales: integridad, confidencialidad y autenticación. La *integridad* es la capacidad de detectar si un mensaje ha sido alterado. La *confidencialidad* es la capacidad de mantener un mensaje ilegible para cualquiera que no sea el destinatario previsto; lo que se consigue mediante criptografía. La *autenticación* es la capacidad de comprobar una identidad notificada. Juntas, estas tres funciones contribuyen a garantizar que los mensajes llegan de manera segura de un punto a otro. (17)

**Modos de seguridad de transporte y de mensaje:** Se utilizan dos mecanismos principales para implementar la seguridad de la transferencia en WCF: el modo de seguridad de transporte y el modo de seguridad de mensaje. (17)

- **El modo de seguridad de transporte:** utiliza un protocolo de nivel de transporte, como HTTPS, para lograr la seguridad de la transferencia. El modo de transporte tiene la ventaja de estar ampliamente adoptado, disponible en muchas plataformas y ser menos complejo desde el punto de vista informático. Sin embargo, tiene la desventaja de que solo protege los mensajes punto a punto.
- **El modo de seguridad de mensaje:** utiliza *WS-Security* (y otras especificaciones) para implementar la seguridad de la transferencia. Dado que la seguridad de mensaje se aplica directamente a los mensajes SOAP, y está incluida en envolturas SOAP junto con los datos de la aplicación, tiene la ventaja de ser independiente del protocolo de transporte, más extensible y de garantizar la seguridad global (frente al protocolo punto a punto). La desventaja es que es mucho

más lenta que el modo de seguridad de transporte porque tiene que tratar con la naturaleza XML de los mensajes SOAP.

Un tercer modo de seguridad utiliza los dos modos anteriores y ofrece las ventajas de ambos. Este modo se denomina **TransportWithMessageCredential**. En este modo, se utiliza la seguridad de mensaje para autenticar al cliente, y la seguridad de transporte para la autenticar el servidor y proporcionar confidencialidad e integridad al mensaje. Gracias a esto, el modo de seguridad **TransportWithMessageCredential** es casi tan rápido como el modo de seguridad de transporte y proporciona la misma extensibilidad de autenticación del cliente que el modo de seguridad de mensaje. No obstante, a diferencia del modo de seguridad de mensaje, no proporciona seguridad completa de extremo a otro.

**Control de acceso:** El control de acceso también se conoce como autorización. La autorización permite a los distintos usuarios disponer de privilegios diferentes para consultar los datos. Por ejemplo, dado que los archivos de recursos humanos de una compañía contienen datos confidenciales de los empleados, solo los administradores pueden consultar estos datos. Lo que es más, los administradores solo pueden consultar datos para sus informes directos. En este caso, el control de acceso está basado tanto en la función ("administrador"), como en la identidad específica del administrador (para evitar que un administrador examine los registros de los empleados de otro administrador).

En WCF, las características de control de acceso se proporcionan mediante la integración con el *Principal Permission Attribute Common Language Runtime* (CLR), y a través de un conjunto de API conocido como el modelo de identidad.

**Auditoría:** La auditoría consiste en registrar los eventos de seguridad en el registro de eventos de Windows. Pueden registrarse eventos relacionados con la seguridad, como autenticaciones con errores (o correctas). (17)

## 1.5. ARQUITECTURAS DE LOS SERVICIOS

Existen disímiles estilos arquitectónicos que posibilitan estructurar y organizar un sistema de *software*, cada uno de ellos tiene sus particularidades, a continuación se describen el modelo de Arquitectura por Capas y el Modelo – Vista – Controlador, por ser los que más se adaptan a la solución que se propone.

## 1.5.1. ARQUITECTURA POR CAPAS

Garlan y Shaw definen el estilo en capas como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. (19)

El estilo de arquitectura basado en capas se identifica por las siguientes características:

- Describe la descomposición de servicios de forma que la mayoría de la interacción ocurre solamente entre capas vecinas.
- Las capas de una aplicación pueden residir en la misma máquina física (misma capa) o puede estar distribuido sobre diferentes computadores (n-capas).
- Los componentes de cada capa se comunican con otros componentes en otras capas a través de interfaces muy bien definidas.
- Este modelo ha sido descrito como una “pirámide invertida de re-uso” donde cada capa agrega responsabilidad y abstracción a la capa directamente sobre ella. (20)

### Principios fundamentales

Los principios comunes que se aplican cuando se diseña para usar este estilo de arquitectura incluyen:

- **Abstracción:** La arquitectura basada en capas abstrae la vista del modelo como un todo y a su vez provee suficientes detalles para entender las relaciones entre capas.
- **Encapsulamiento:** El diseño no hace exaltaciones acerca de tipos de datos, métodos, propiedades o implementación.
- **Funcionalidad claramente definida:** El diseño claramente define la separación entre la funcionalidad de cada capa. Capas superiores como la capa de Presentación envía comandos a las capas inferiores como la capa de Negocios y la capa de Datos y los datos fluyen hacia y desde las capas en cualquier sentido.
- **Alta cohesión:** Cada capa contiene funcionalidades directamente relacionadas con la tarea de dicha capa.

- **Reutilizable:** Las capas inferiores no tienen ninguna dependencia con las capas superiores, permitiéndoles ser reutilizables en otros escenarios.
- **Desacople:** La comunicación entre las capas está basada en la abstracción lo que provee un desacople entre las capas. (21)

## Beneficios

Los principales beneficios del estilo de arquitectura basado en capas son:

- **Abstracción:** Las capas permiten cambios que se realicen en un nivel abstracto. Pueden incrementar o disminuir el nivel de abstracción usado en cada capa de la “pila” jerárquica.
- **Aislamiento:** El estilo de arquitectura de capas permite aislar los cambios en tecnologías a ciertas capas para reducir el impacto en el sistema total.
- **Rendimiento:** Distribuir las capas entre múltiples sistemas (físicos) puede incrementar la escalabilidad, la tolerancia a fallos y el rendimiento.
- **Mejoras en pruebas:** La capacidad de realizar pruebas se beneficia de tener interfaces bien definidas para cada capa así como la habilidad para cambiar a diferentes implementaciones de las interfaces de cada capa. (21)

### 1.5.2. PATRÓN DE ARQUITECTURA MODELO-VISTA-CONTROLADOR

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el acceso a datos y la lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. (22) (23)

## Beneficios

Los principales beneficios del patrón de arquitectura Modelo – Vista – Controlador son:

- Menor acoplamiento.
- Desacopla las vistas de los modelos.
- Desacopla los modelos de la forma en que se muestran e ingresan los datos.

- Mayor cohesión:
  - Cada elemento del patrón está altamente especializado en su tarea (la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su objetivo de negocio).
- Las vistas proveen mayor flexibilidad y agilidad:
  - Se puede crear múltiples vistas de un modelo.
  - Las vistas pueden anidarse.
  - Se puede cambiar el modo en que una vista responde al usuario sin cambiar su representación visual.
  - Se puede sincronizar las vistas.
  - Las vistas pueden concentrarse en diferentes aspectos del modelo.
- Mayor facilidad para el desarrollo de clientes ricos en múltiples dispositivos y canales:
  - Una vista para cada dispositivo que puede variar según sus capacidades.
  - Una vista para la Web y otra para aplicaciones de escritorio.
- Más claridad de diseño.
- Facilita el mantenimiento.
- Mayor escalabilidad. (24)

## 1.6. CAPA DE ACCESO A DATOS

### 1.6.1. TIERDEVELOPER 6.1

Es una herramienta de mapeo O/R (*Object-to-Relational*) muy completa. Genera código para .NET, el patrón de persistencia que utiliza maneja las tablas que mapea de la BD (base de datos) como objetos. Tiene la desventaja de poseer licencia comercial y aunque genera código obteniendo datos desde varios gestores de BD, sólo lo hace en dos lenguajes de programación: C#.NET y VB.NET. Esta versión genera el acceso a datos para cuatro fuentes de datos: *Oracle*, *SQL<sup>9</sup> (Structured Query Language) Server*, *IBM DB2* (DB2 era el antiguo nombre del DBMS (*Database Management System*)), muchos consideran que fue

---

<sup>9</sup> En español Lenguaje de Consultas Estructurado es un lenguaje declarativo de acceso a bases de datos.

uno de los primeros productos en usar SQL) y *Microsoft Access*. Es limitada, porque no es aplicable para algunos gestores de BD y lenguajes libres. (25)

## 1.6.2. NHIBERNATE 3.0

Este software es considerado un ORM<sup>10</sup> (*Object Relational Mapping*). Es una herramienta libre muy poderosa. Fue la conversión de *Hibernate* de lenguaje Java a C# posibilitando la integración de *Hibernate* con la plataforma .NET. El código generado en *NHibernate* también es válido en Mono<sup>11</sup>. Posee una serie de características claves que se presentan a continuación: (25)

- Soporta lenguajes de programación orientada a objeto, herencia, polimorfismo, composición y las colecciones de los *frameworks de .NET*, incluyendo las colecciones genéricas.
- Los API de *NHibernate* usan los idiomas y conversiones de .NET.
- Posee una rica variedad de mapeos para colecciones de objetos dependientes.
- No hay generación de código adicional o pasos extras de procesamiento en el proceso de compilación.
- Especifica el SQL exacto que se debe usar para persistirlos objetos.
- Es un software libre, está certificado bajo el LGPL (*Lesser GNU Public License*).
- Con *.NET framework 3.5*, permite usar expresiones lambda<sup>12</sup> y LINQ<sup>13</sup> (*Language Integrated Query*). Esto ha llevado a una explosión de nuevas ventajas de configurar y de consultar datos.

## 1.6.3. ADO.NET ENTITY FRAMEWORK

ADO.NET *Entity Framework* permite 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 con acceso a datos. Las aplicaciones de *Entity Framework* ofrecen las siguientes ventajas:

<sup>10</sup> En español Mapeo Relacional-Objetos se utilizan para mapear las entidades de la base de datos a objetos.

<sup>11</sup> Proyecto de código abierto actualmente impulsado por Novell para crear un grupo de herramientas libres, basadas en GNU/Linux y compatibles con .NET.

<sup>12</sup> Una expresión lambda es una función o una subrutina sin nombre que se puede utilizar donde haya un delegado válido.

<sup>13</sup> LINQ: Lenguaje Integrado de Consulta es un conjunto de características que agregan capacidades de consulta a los lenguajes C# y Visual Basic.

- 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 LINQ proporciona validación de la sintaxis en el momento de la compilación para consultas en un modelo conceptual. (26)

## 1.7. AMBIENTE DE DESARROLLO

En la Informática, obtener un software de calidad implica la utilización de metodologías, tecnologías y herramientas que permitan uniformar la filosofía de trabajo. Para dar solución al problema planteado, se utiliza el ambiente de desarrollo definido por el proyecto Identificación, Inmigración y Extranjería de la República de Cuba de la Universidad de las Ciencias Informáticas.

### 1.7.1. METODOLOGÍA DE DESARROLLO DE SOFTWARE

#### **MSF (*Microsoft Solution Framework*) for CMMI (*Capability Maturity Model Integration*)**

MSF for CMMI es una metodología para el desarrollo de *software* para la planificación, desarrollo y gestión de proyectos tecnológicos. Está centrado en el modelo de procesos y de equipo dejando los demás aspectos en segundo plano. Se compone de varios modelos que se encargan de cada una de las fases del desarrollo de un proyecto: modelo de arquitectura del proyecto, de equipo, de procesos, de gestión de riesgo, de diseño de procesos y de aplicación. Esta metodología no es rígida ya que sabe que no existe una sola estructura que se pueda acoplar a todos los tipos de proyectos.

Es una metodología integrada, ya que combina muchos elementos y características y además, es una metodología productiva. Se enfoca más en las habilidades y cualidades de las personas que en la eficacia



de los modelos de procesos. MSF for CMMI está basado en las mejores prácticas del mundo real, basado en las experiencias de *Microsoft Process Improvement*. MSF for CMMI contiene un meta-modelo que estaba basado en la filosofía de que un proceso esta creado para ayudar y no para estorbar, este modelo es compartido para procesos ágiles y maduros. Consta de cinco fases: Previsión, Planificación, Desarrollo, Estabilización e Implementación y es continuamente refinado por clientes, consultores, grupos de desarrollo de *Microsoft*. (27)

## 1.7.2. HERRAMIENTAS DE MODELADO

### ***Altova UModel 2009 Service Pack 1 Enterprise Architects***

*Altova UML* es como un puente entre las más simples herramientas de modelado que no necesitan mayor comprensión para estructurar diagramas y las más complejas herramientas de apoyo que poseen hasta el último matiz de especificaciones. Permite construir nueve de los más comunes y útiles tipos de diagramas UML.

Los diagramas son organizados en una jerarquía de paquetes y se pueden manejar de dos formas, ya sea en la vista de modelo (que muestra la jerarquía entera), o en la vista de diagrama (donde son agrupados por tipo de diagrama). También se pueden compartir proyectos entre uno o más proyectos de UModel.

Se puede utilizar para generar código en *Java* o *C#*; o bien, importar clases de *Java* o *C#* para construir los diagramas UML correspondientes. La generación de código está basada en plantillas, que son totalmente modificables y así permite un alto nivel de control sobre el código generado. (28)

## 1.7.3. SISTEMA GESTOR DE BASE DATOS

### **Oracle 11g**

Es un sistema de gestión de base datos relacional desarrollado por *Oracle Corporation*. Se considera uno de los sistemas de bases de datos más completos, destacándose por su:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Soporte multiplataforma.

La base de datos *Oracle* en *Windows* ha evolucionado desde un nivel básico de integración del sistema operativo hasta utilizar servicios más avanzados en la plataforma *Windows*. (29)

Brinda una plataforma segura y escalable para el acceso confiable y rápido a todo tipo de información utilizando las interfaces estándar del sector. Permite una sólida administración del contenido para tipos avanzados de datos, esta versión incluye nuevas mejoras y características como son:

- La recopilación automática para lenguaje PL/SQL<sup>14</sup> (*Procedural Language/Structured Query Language*) y *Java* en la base de datos.
- *Triggers* más rápidos, con inclusión de invocaciones más eficientes de *triggers* por fila.
- Operaciones SQL más rápidas y fáciles.
- Replicación *Oracle Data Guard*<sup>15</sup> y *Oracle Streams*<sup>16</sup> más rápida.
- Conexiones directas, más rápidas y confiables, a los dispositivos de almacenamiento del sistema de archivo de red.
- Actualizaciones más rápidas.
- Recuperación más rápida de archivos de gran tamaño. (30)

## Lenguaje de programación PL/SQL

Este lenguaje es un gestor que va incorporado en *Oracle* para solventar las grandes limitaciones que existían con los editores SQL, en cuanto al control de las secuencias de ejecución de instrucciones, el uso de variables, la gestión de posibles errores.

Este lenguaje está basado en ADA<sup>17</sup>, por lo que incluye todas las características de los lenguajes de tercera generación. Esto permite manejar las variables, tener una estructura modular (procedimientos y funciones) y controlar las excepciones. Además incorpora un completo soporte para la programación orientada a objetos.

---

<sup>14</sup> Lenguaje de programación incrustado en Oracle.

<sup>15</sup> Oracle Data Guard es la funcionalidad de la base de datos Oracle que brinda la mayor y más efectiva disponibilidad, protección y recuperación ante desastres de los datos

<sup>16</sup> Oracle Streams propaga y administra datos, transacciones y eventos en una fuente de datos ya sea dentro de una base de datos, o de una base de datos a otra.

<sup>17</sup> Ada es un lenguaje de programación orientado a objetos, fuertemente tipado y concurrente.

Los programas creados con PL/SQL pueden ser almacenados en base de datos como cualquier otro objeto, quedando disponibles para los usuarios. El uso del lenguaje PL/SQL es imprescindible para construir disparadores de bases de datos.

PL/SQL está incluido en el servidor y en algunas herramientas de cliente. Soporta todos los comandos de consulta y manipulación de datos, aportando al lenguaje SQL las estructuras de control y otros elementos propios de los lenguajes de programación de tercera generación. (31)

## 1.7.4. ENTORNO DE DESARROLLO INTEGRADO

### **Microsoft Visual Studio 2010**

*Visual Studio* es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. *Visual Basic*, *Visual C#* y *Visual C++* utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Además, dichos lenguajes utilizan las funciones de *.NET framework*, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de Aplicaciones Web ASP y Servicios Web XML. Además, *Visual Studio* está optimizado para el diseño, el desarrollo y la implementación en equipo de soluciones empresariales. (32)

*Microsoft Visual Studio 2010* proporciona un entorno integrado de herramientas e infraestructura de servidor que simplifica todo el proceso de desarrollo de aplicaciones. Ofrece resultados empresariales usando procesos productivos, predecibles y personalizables, y aumenta la transparencia y el seguimiento durante el ciclo de vida con análisis detallados. Aumenta la productividad del equipo usando características de colaboración avanzadas y herramientas de pruebas y depuración integradas para detectar y corregir errores de un modo rápido y sencillo, lo que permite crear soluciones de alta calidad y reducir el coste del desarrollo de soluciones.

En la presente versión de *Visual Studio* se incorporan diferentes tecnologías a las aplicaciones que pueden ser creadas con este. Algunas de estas tecnologías es la compatibilidad con aplicaciones desarrolladas en las versiones Compact, 3.0, 3.5 y 4.0 del framework *.NET*, *Windows Presentation Foundation (WPF)*, *Windows Communication Foundation (WCF)*, *Windows Workflow Foundation*, *Silverlight*, *Windows Forms*, *ASP.NET*, *Extensible Application Markup Language (XAML)* y *Language Integrated Query (LINQ)*.

Incluye además compatibilidad con diferentes aplicaciones y lenguajes entre ellos *Visual Basic*, *Visual C#*, *Visual C++*, *Visual F#*, *JScript*, aplicaciones para *Web*, *Office*, *Windows*, *SharePoint* y *Smart Device*. (27)

## 1.7.5. CONTROL DE VERSIONES

### ***Microsoft Visual Studio 2010 Team Explorer***

Todos los miembros del equipo trabajan con proyectos de equipo utilizando *Team Explorer* en el entorno de desarrollo integrado (IDE) de *Visual Studio*, este se conecta al servidor *Team Foundation* y muestra los proyectos de equipo existentes en el servidor. A través de *Team Explorer*, los miembros del equipo pueden buscar y actualizar elementos de trabajo, ver informes, administrar documentos y trabajar con generaciones del producto. (27)

## 1.7.6. FRAMEWORK

### ***Microsoft .NET Framework 4.0.***

La plataforma .NET de *Microsoft* es un componente de *software* que puede ser añadido al Sistema Operativo *Windows*. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma. Esta solución es el producto principal en la oferta de *Microsoft*, y pretende ser utilizada por la mayoría de las aplicaciones creadas para la plataforma *Windows*.

El *framework* .NET se incluye en *Windows Server 2008*, *Windows Vista* y *Windows 7*. De igual manera, la versión actual de dicho componente puede ser instalada en *Windows XP*, y en la familia de Sistemas Operativos *Windows Server 2003*. Una versión "reducida" de *.NET framework* está disponible para la plataforma *Windows Mobile*, incluyendo teléfonos inteligentes. (27)

Este *framework* contiene muchas características antiguas y nuevas que se acoplan a las de las versiones anteriores *.NET framework 2.0*, *3.0* y *3.5*, por ejemplo, los conjuntos de características de *Windows Workflow Foundation* (WWF), *Windows Communication Foundation* (WCF) y *Windows Presentation Foundation* (WPF).

Es neutral en cuanto al lenguaje de programación y funciona en base a librerías. Se puede programar utilizando *Visual Basic.NET*, *C++.NET*, *C#* y *J#*. Esta neutralidad en el lenguaje de programación es

posible gracias a la arquitectura del *.NET framework*. Es un componente integral de *Windows* que admite la creación y la ejecución de la siguiente generación de aplicaciones y Servicios Web XML. (27)

El *framework .NET* proporciona el soporte necesario para los Servicios Web, de manera que los desarrolladores puedan codificar, descubrir, depurar, instalar y consumir Servicios Web utilizando cualquiera de los más de 20 lenguajes de programación soportados por este entorno. (16)

## **Oracle Data Provider para .NET. 10.2.0.2**

El proveedor de datos de *Oracle* es un conjunto de librerías que ofrecen un mejor acceso a la información de la base de datos de *Oracle* desde un entorno *.NET*. Que permite a los desarrolladores aprovechar las funcionalidades avanzadas de la base de datos de *Oracle*, las optimizaciones de desempeño y las características avanzadas de seguridad. Ofrece a los programadores un mejor ambiente, flexibilidad, y alternativas para sus aplicaciones *.NET*. Con él, los desarrolladores pueden utilizar *.NET*, pero sin sacrificar las poderosas capacidades de administración de datos que *Oracle* brinda. (33)

## **Lenguaje de programación C# 4.0**

Es un lenguaje de propósito general diseñado por *Microsoft* para su plataforma *.NET*. Aunque es posible escribir código para la plataforma *.NET* en muchos otros lenguajes, es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando *C#* es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que carece de elementos heredados innecesariamente en *.NET*. Por esta razón, se suele decir que es el lenguaje nativo de *.NET*.

La sintaxis y estructura es muy parecida a la de *C++* o *Java*, puesto que la intención de *Microsoft* es facilitar la migración de códigos escritos en estos lenguajes a *C#* y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son comparables con los de *Visual Basic*.

En resumen, *C#* es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como *Visual Basic*, *Java* o *C++* y las combina en uno solo. El hecho de ser relativamente reciente no implica que sea inmaduro, pues *Microsoft* ha escrito la mayor parte de su biblioteca de clases base usándolo, por lo que su compilador es el más depurado y optimizado de los incluidos en el *.NET framework SDK*. (27)

## **Extensible Markup Language (XML)**

El Lenguaje de Marcado Extensible no es más que un conjunto de reglas para definir etiquetas semánticas que organizan un documento en diferentes partes. Es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados. Se pueden definir algunas características:

- Es una arquitectura abierta y extensible. No se necesitan versiones para que puedan funcionar en futuros navegadores. Los identificadores pueden crearse de manera simple y ser adaptados en el acto en internet/intranet por medio de un validador de documentos.
- Mayor consistencia, homogeneidad y amplitud de los identificadores descriptivos del documento con XML.
- Integración de los datos de las fuentes más dispares. Se podrá hacer el intercambio de documentos entre las aplicaciones tanto en la propia PC como en una red local o extensa.
- Datos compuestos de múltiples aplicaciones. La extensibilidad y flexibilidad de este lenguaje permite agrupar una gran variedad de aplicaciones, desde páginas web hasta bases de datos.
- Gestión y manipulación de los datos desde el propio cliente web. (34)

## 1.8. CONCLUSIONES

Después del estudio realizado de los principios arquitectónicos de la Arquitectura Orientada a Servicios, de las principales arquitecturas para implementar servicios, de los ORM más utilizados en la actualidad se concluye:

- La mejor opción para el intercambio de información en la arquitectura SOA del Sistema de Identificación, Inmigración y Extranjería de Cuba es la implementación de una Arquitectura Orientada a Servicios utilizando la tecnología WCF de .NET ya que se acopla al ambiente de desarrollo del proyecto y a las necesidades del sistema.
- La implementación de los servicios se debe realizar siguiendo una arquitectura por capas bien estructuradas teniendo en la capa de Acceso a Datos, la utilización de *Entity Framework* de .NET como ORM.
- La solución propuesta se ajusta con la metodología, lenguajes y herramientas seleccionadas para el desarrollo del proyecto, las cuales se encuentran entre las más utilizadas y eficaces de la actualidad.

## Capítulo 2 PROPUESTA DE SOLUCIÓN

### 2.1. INTRODUCCIÓN

En el presente capítulo se presenta la solución que se propone para dar respuesta al problema planteado. Se describe el modelo de dominio realizado en el desarrollo de la presente investigación, cada uno de los servicios que se desarrollan y se exponen los requerimientos funcionales y no funcionales para los servicios comunes. Se toma como ejemplo el servicio *EncoderService*, el cual es el responsable de gestionar todos los nomencladores en la solución.

### 2.2. PRESENTACIÓN DE LA SOLUCIÓN

La solución es un paquete de servicios comunes basados en una Arquitectura Orientada a Servicios, para el desarrollo del Sistema de Identificación, Inmigración y Extranjería de Cuba, que utilizando la tecnología *Windows Communication Foundation (WCF)* de .NET permite la orquestación de procesos de negocio con las funcionalidades que se exponen. Su principal objetivo es proporcionar Servicios Web que permitan gestionar las necesidades del sistema a nivel nacional y abstraer al sistema de las capas de Negocio y Acceso a Datos. Los servicios siguen una arquitectura por capas bien estructuradas, teniendo en la capa de Acceso a Datos la utilización de *Entity Framework* de .NET como ORM. Además, agrupa un conjunto de servicios que le dan mayor dinamismo al desarrollo de sistemas centrado en la orquestación de procesos de negocio, específicamente para un ambiente web.

Con el paquete de servicios se abstraen las capas superiores del sistema (Presentación y Procesos) de las capas de Negocio y Acceso a Datos mediante la exposición y consumo de los Servicios Web. Los servicios están estructurados en capas las cuales tienen responsabilidades bien definidas dentro de la arquitectura de los servicios.

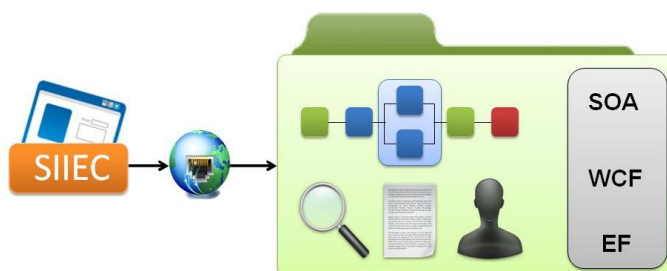


Ilustración 3 Abstracción del sistema.

## 2.3. MODELO DE DOMINIO

Después de haber hecho el análisis correspondiente no se pudieron identificar procesos de negocios bien claros, sólo elementos conceptuales; se arribó a la conclusión de que el paquete de servicios no necesita de un modelado completo del negocio, por lo que se propone un modelo de dominio. Este se ejecuta para obtener una vista general de las entidades o conceptos que intervienen en el negocio.

Con la realización del modelo de dominio se pretende ayudar a comprender los conceptos con que interaccionan los subsistemas del SIIEC y los que deberá utilizar la solución para intercambiar los datos entre las distintas capas de la arquitectura. Para su elaboración fue necesario el estudio de las funcionalidades comunes del sistema, con tales fines se realizaron las siguientes actividades:

- Identificar clases conceptuales y las relaciones que se establecen entre ellas.
- Realizar el diagrama de clases.

A continuación se describen todos los conceptos y sus relaciones presentes en el modelo de dominio:

- **SUIN:** Sistema Único de Identificación Nacional, subsistema que se utiliza en la las Oficinas de Carné de Identidad y Registro de población, donde se realizan los procesos concernientes a la identificación de personas nacionales.
- **SIEC:** Sistema de Inmigración, Extranjería y Ciudadanía, subsistema que se utiliza en la Dirección de Inmigración y Extranjería con el objetivo de realizar el control de las normativas legales relacionadas con la inmigración, extranjería y ciudadanía en Cuba.
- **SIIEC:** Sistema de Identificación, Inmigración y Extranjería de Cuba que cuenta con los subsistemas SUIN y SIEC.
- **ScorePlus:** Sistema externo con el cual interactúan el Sistema de Identificación, Inmigración y Extranjería para obtener datos operativos de las personas.
- **Trámites:** Conjunto de trámites que se realizan en las oficinas de la DIE y la DIR tales como solicitud de carné de identidad, solicitud de pasaporte, entre otros.
- **Personas:** Conjunto de personas registradas en el Sistema de Identificación, Inmigración y Extranjería que a su vez interactúan con este solicitando trámites.



- **Documentos:** Documentos utilizados para guardar datos de los trámites y otros datos de los sistemas.
- **Procesos:** Los procesos están asociados a los tramites que se realizan formando parte de estos.
- **Direcciones:** Datos de las direcciones asociados a las personas registradas en el sistema.
- **Registros\_Operativos:** Registros asociados a las personas que contienen hechos delictivos y otros datos operativos.

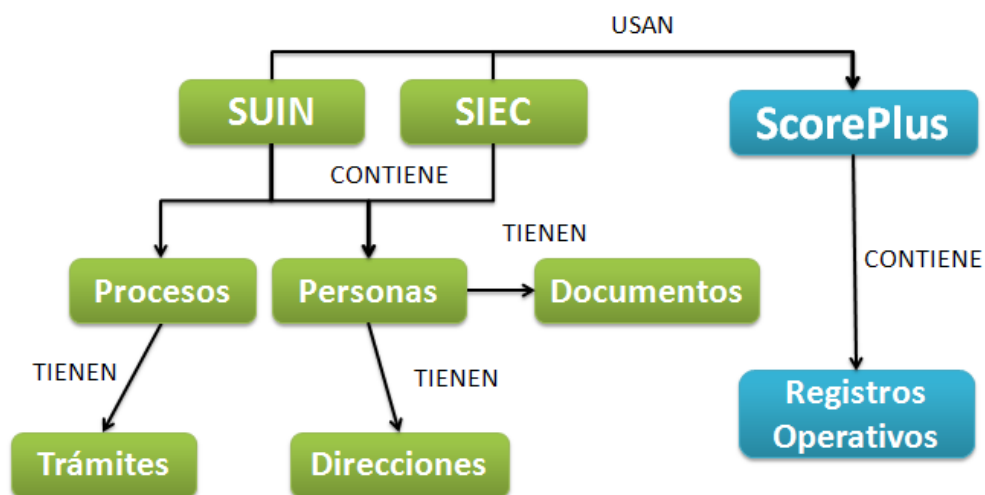


Ilustración 4 Modelo de dominio.

## 2.4. DESCRIPCIÓN DEL MODELO DE DOMINIO

El Sistema de Identificación, Inmigración y Extranjería de Cuba formado por los subsistemas: SUIN y SIEC; que pertenecen a la Dirección de Identificación y Registro y la Dirección de Inmigración y Extranjería respectivamente. Ambos subsistemas principales están compuestos por otros subsistemas, los cuales están asociados a un área específica de la organización. Estos subsistemas están divididos por módulos, lo que permite separar y agrupar las responsabilidades dependiendo de su función dentro del subsistema. Los módulos a su vez están compuestos por funcionalidades específicas o comunes, que son una o varias operaciones necesarias para su desempeño. Algunas de las funcionalidades comunes de los módulos de los subsistemas SUIN y SIEC son:

- Gestionar persona: permite gestionar los datos de una persona.

- Gestionar dirección: posibilita manejar y almacenar los datos asociados a las direcciones de las personas.
- Obtener nomenclador: interactúa con los nomencladores almacenados.
- Gestionar procesos: es la funcionalidad necesaria para manejar y almacenar los datos asociados a los procesos de las personas.
- La búsqueda de registros operativos: se emplea para comprobar si la persona tiene algún impedimento para realizar un proceso.
- Obtener expediente de la persona que tiene la responsabilidad de obtener de cada persona sus datos identificativos, biométricos y los procesos realizados por esta.

## 2.5. REQUERIMIENTOS DE LOS SERVICIOS

El propósito fundamental del flujo de trabajo de los requisitos es guiar de manera correcta el desarrollo del sistema. Esto se consigue mediante una descripción de los requisitos del sistema (es decir, las condiciones o capacidades que el sistema debe cumplir) suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente y los desarrolladores sobre qué debe y qué no debe hacer el sistema (35).

### 2.5.1. DESCRIPCIÓN DE LOS ROLES

Rol	Objetivo
Desarrollador	Utiliza las funcionalidades del paquete de servicios sin necesidad de implementarlo él, puede habilitar o deshabilitar las características que sean necesarias para el desarrollo.
Sistema	Realiza las acciones que sean necesarias de acuerdo con las configuraciones realizadas por el desarrollador. Garantiza el correcto funcionamiento de todas las funcionalidades habilitadas por el desarrollador para su solución.

Tabla 1 Descripción de los roles

### 2.5.2. DEFINICIÓN DE REQUISITOS FUNCIONALES

A continuación se muestran los requisitos funcionales del servicio de nomencladores. Para profundizar en el análisis de los restantes requisitos funcionales del paquete ver **Anexo I**.

## RF 1 Permitir la gestión de los nomencladores.

RF1.1 Mostrar los nomencladores existentes dado una clasificación de nomenclador.

RF1.2 Contar los nomencladores existentes dado una clasificación de nomenclador.

RF1.3 Mostrar los nomencladores existentes dado una clasificación de nomenclador, un número de inicio y una cantidad de total.

RF1.4 Mostrar los datos de un nomenclador según parámetros relacionales.

RF1.5 Contar los datos de un nomenclador que coinciden con los parámetros de la búsqueda realizada.

RF1.6 Mostrar los datos de un nomenclador según parámetros y un número de inicio y una cantidad de total.

### 2.5.3. ESPECIFICACIÓN DE LOS REQUISITOS FUNCIONALES

En este epígrafe se muestra un fragmento de la especificación de requisitos del servicio de nomencladores (*EncoderService*), el resto de la especificación de requerimientos se puede encontrar en el **Anexo II**.

RF1.1. Mostrar los nomencladores existentes dado una clasificación de nomenclador.		
<b>Propósito</b>	Mostrar los nomencladores existentes dado una clasificación de nomenclador.	
<b>Roles</b>	Sistema SIIEC.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NOMENCLADORES	NAREASMININT NATRIBUTODOCUMENTO NATRIBUTOOBJETOCNI NCATEGORIAEXTRANJERO NCAUSACANCELARDENEGAR NCAUSACNI NCDR NCIRCUNSCRIPCION NCODIGOOACIPAI

		NCOLORCABELLO NCOLOROJO NCOLORPIEL NCONDICIONMIGRATORIA NCONSEJOPOPULAR NCONTINENTE NDEPARTAMENTO NENTIDADTRAMITADORA NESTADOACTIVACION NESTADOCIRCULACION NESTADOFINALPROCESO NEXPRESIONROSTRO NEXTENSIONIMAGEN NFORMAENTRADAPAISPER NFORMALLEGADACNI NFORMASSALIDAPAISPER NGENERO NGRUPOLUGAR NINSTITUCION NLUGARGENERAL NLUGARTRASLADO NMUNICIPIO NNIVELCULTURAL NNOMBREANNO NORGANO NPAIS NPODERPOPULAR NPROVINCIA NPUNTOCARACTERISTICO NREGIONGEOGRAFICA
--	--	--

		NREGISTROCIVIL NREGLALUGAR NREQUISITOTIPOTRAMITE NREQUISITOTRAMITE NSECCION NSITMIGENPOLITICA NSITMIGFUERAPOLITICA NTIPOAPLICACION NTIPOCIRCULACION NTIPODATO NTIPODECISIONCNI NTIPODIRECCION NTIPODOCCNI NTIPODOCIDENTIFICACION NTIPODOCUMENTO NTIPOIMAGEN NTIPOIMAGENFACIAL NTIPOINCAPACIDAD NTIPOINCIDENCIA NTIPOINMUEBLE NTIPOPOLUGAR NTIPOPOMEDIDACNI NTIPOPONEDA NTIPOOBJETOCNI NTIPOOFICINA NTIPOPERSMISOVIAJE NTIPOPERSONA NTIPOPDERPOPULAR NTIPOPROCESO NTIPOPROCESOBLOQUEO
--	--	---

	NTIPOPROCESOTIPOTRAMITE NTIPOPUNTOSIMPORTANCIA NTIPOSERVICIOAO NTIPOSOLICITUDCIRC NTIPOPOTRAMITE NTIPOVINCULOOFAMILIAR NTIPOVISITACNI NVALIDACIONIMAGEN NVIASOLICITUDNEGREGRESO NZONA
<b>Descripción</b>	RF1.1.1. Obtener el tipo de nomenclador. RF1.1.2. Buscar los nomencladores del tipo obtenido. RF1.1.3. Mostrar los posibles nomencladores del tipo de nomenclador especificado. RF1.1.4. Mostrar detalles de cada nomenclador devuelto.
<b>Validaciones</b>	1. Validar que el tipo de nomenclador a buscar sea correcto.
<b>Postcondiciones</b>	Se devuelven los nomencladores encontrados según el tipo introducido.
<b>Prototipo</b>	No procede.

**Tabla 2 Especificación del Servicio de Nomencladores.**

## 2.5.4. ESPECIFICACIÓN DE REQUISITOS NO FUNCIONALES

### Software

- RnF 1. Utilizar Sistema Operativo: *Windows Server 2008 R2*.
- RnF 2. Utilizar *.NET framework v3.5* o superior.
- RnF 3. Utilizar *Internet Information Server 7.0*.

### Diseño e implementación

- RnF 4. Utilizar las tecnologías que brinda el *framework .NET* sobre la programación orientada a objetos.
- RnF 5. El sistema debe implementarse usando el lenguaje *C#*.

- RnF 6. El sistema gestor de bases de datos, será Oracle 11g.
- RnF 7. El sistema debe desarrollarse usando el IDE *Visual Studio TeamSystem* 2010.
- RnF 8. Se utilizará *el Microsoft Visual Studio Team Foundation Server* 2010 como control de código fuente.
- RnF 9. El paquete de servicios debe codificarse siguiendo los estilos de código definidos por el proyecto.

## Seguridad

- RnF 10. Garantizar un fuerte tratamiento de excepciones.

## Usabilidad

- RnF 11. El paquete de servicios poseerá estructura y diseño homogéneos, así como los términos utilizados se establecerán acorde al negocio en todos los servicios mejorando el entendimiento de los usuarios.

## Fiabilidad

- RnF 12. Los servicios deben estar disponibles las 24 horas durante los 7 días de la semana.
- RnF 13. No se realizarán mantenimientos preventivos en horario laboral, deberán ejecutarse en un horario estipulado o los fines de semana, para no afectar la disponibilidad de los servicios.
- RnF 14. Las fallas del *software* se dividirán en dos categorías:
- **Simple:** la solución y la actualización se realizarán en línea en un período inferior a cuatro horas.
  - **Complejas:** la solución y actualización se realizarán en un tiempo que se definirá posterior a una evaluación detallada.
- RnF 15. El sistema llevará un sistema de tracking<sup>18</sup> de errores.
- RnF 16. Se garantizará la consistencia de los datos, se realizarán comprobaciones y validaciones automáticas en todos los casos posibles.

## Eficiencia

---

<sup>18</sup> Seguimiento de incidencias.

RnF 17. Los servicios deben ser capaces de soportar una cantidad escalable de peticiones y dar respuestas efectivas y rápidas.

## Soporte

- RnF 18. Deben poseer documentos técnicos.
- RnF 19. Deben poseer el Manual de Usuario.
- RnF 20. Deben poseer el Manual de Administración y Configuración.

## Requisitos de Licencia

RnF 21. Para el desarrollo del sistema se necesitan un conjunto de aplicaciones, plataformas, Sistemas Operativos, gestores de bases de datos, herramientas, que son sistemas propietarios y necesitan de licencias para su buen desempeño y soporte, las cuales son:

- *Visual Studio Team System 2010.*
- *PL/SQL Developer.*
- *Oracle Database Enterprise Edition.*
- *Windows Server 2008 R2.*
- *Embarcadero ER Studio 8.0.*

## 2.6. DESCRIPCIÓN DE LOS SERVICIOS PROPUESTOS

Después de realizado el estudio de modelo de dominio y la especificación y descripción de los requisitos, se pudieron identificar a partir de las necesidades de las funcionalidades comunes un conjunto de servicios que dan solución al problema planteado, estos servicios se describen a continuación:

- **Administración de Personas (*PersonManagerService*):** Es el servicio más importante del paquete pues se encarga de gestionar los datos de las personas, ya sean nacionales, extranjeras o no registradas. Permite insertar y actualizar una persona con todos los datos o los básicos, además de modificar el estado fallecido de la persona o la dirección permanente de la misma.
- **Petición de Procesos (*ProcessRequestService*):** Servicio que permite iniciar uno de los procesos del Sistema de Identificación, Inmigración y Extranjería de Cuba. El flujo del servicio consiste en comprobar si se puede realizar el proceso solicitado, de ser posible se adiciona el



proceso y se crea una nueva instancia del *workflow* asociado a este, en caso que no se pueda iniciar se devuelven los procesos que impiden que se inicialice el proceso solicitado.

- **Búsqueda de Direcciones (*SearchAddressService*):** Servicio que brinda funcionalidades capaces de recopilar datos de direcciones asociadas a una persona. Este servicio después de obtener los datos específicos que conforman una dirección, genera una tira de direcciones. Permite filtrar por los siguientes parámetros de búsqueda: número de identificación de la persona y la tira<sup>19</sup> de dirección.
- **Búsqueda de Personas (*SearchPersonService*):** Servicio fundamental en el Sistema de Identificación, Inmigración y Extranjería de Cuba ya que es utilizado por varios módulos. Es capaz de buscar las personas nacionales, extranjeras y no registradas dado el número de identidad, la dirección y/o datos comunes en las personas.
- **Registros Operativos (*OperativeRecordService*):** Servicio que interactúa con un servicio externo llamado *ScorePlus* el cual brinda una interfaz que devuelve los resultados de la consulta realizada a los registros de: Circulados, Prisiones, Policía y Antecedentes Penales para una persona, devolviendo un listado de registros operativos asociado a la persona. Estas funcionalidades pueden invocarse pasándole el número de identidad de una persona o realizando una búsqueda fonética por sus datos identificativos tales como: primer nombre, segundo nombre y apellidos.
- **Servicio de Nomencladores (*EncoderService*):** Servicio encargado de gestionar los nomencladores del Sistema de Identificación, Inmigración y Extranjería de Cuba y a su vez listar los nomencladores existentes dado un tipo de nomenclador específico. Este también permite obtener la cantidad de datos de un nomenclador y especificar un rango para los resultados.

## 2.7. CONCLUSIONES

- Se realizó el modelo de dominio que permitió la comprensión de los conceptos con que interaccionan los servicios y los que utiliza la solución para intercambiar los datos entre las distintas capas de la arquitectura.

---

<sup>19</sup> Conjunto de datos que conforman una dirección.

- A partir del modelo de dominio del sistema se determinaron los requisitos funcionales con los que debe cumplir el paquete de servicios. También se mostraron los requisitos no funcionales que son las características que debe tener el sistema.

## Capítulo 3 ANÁLISIS Y DISEÑO DEL SISTEMA

### 3.1. INTRODUCCIÓN

En este capítulo se realiza una modelación del conjunto de servicios comunes. Se analizan sus características a través de casos de usos críticos o arquitectónicamente significativos. Además se especifica el comportamiento de la solución en cada uno de los servicios y componentes que se proponen. Se realiza el diseño de la base de datos, teniendo en cuenta el diagrama de clases persistentes y el modelo de datos.

### 3.2. ARQUITECTURA DE LA SOLUCIÓN

El paquete de servicios comunes (*CommonCore*) utilizando una arquitectura SOA, está diseñado siguiendo un patrón de capas bien definidas y diseñadas para reducir al máximo el acoplamiento y aumentar la reutilización entre cada una de las capas. El *CommonCore* se encuentra ubicado en la capa de Negocio de la arquitectura general de la solución y tiene parte en la capa de Acceso a Datos interactuando con los modelos y entidades generados por el *Entity Framework*. Los servicios han sido diseñados teniendo en cuenta las necesidades comunes de los trámites y procesos desarrollados por cada uno de los subsistemas del SIIEC y la visión de la organización del uso de SOA. Las capas de cada uno de los servicios que quedan definidas con la arquitectura del sistema como se muestra en la **Ilustración 5**.

- **Presentación:** La capa superior del *CommonCore* contiene la presentación de los servicios *Windows Communication Foundation* (WCF) los cuales interactúan con los servicios base que son los que invocan a las capas inferiores. En esta capa se incluyen las entidades del negocio de los servicios, se define la interfaz de los conectores, lo que permite que esta capa interactúe con más facilidad con dichos componentes y obliga al modelo a implementar las responsabilidades definidas en esta capa.
- **Controlador:** En el controlador se reciben las peticiones de la capa superior y se implementa la lógica de negocio del servicio realizando las peticiones al conector de la capa de Acceso a Datos y manejando los datos devueltos por este, necesarios para el negocio del servicio. Esta capa es la intermediaria de la arquitectura de los servicios.

- **Conector:** Los conectores permiten establecer una conexión entre las entidades del negocio y la base de datos. Interactúan con el modelo y las entidades generadas por el *Entity Framework*. En el conector se realizan las operaciones básicas del acceso a datos utilizando LINQ.
- **Modelo:** Conjunto de funcionalidades y operaciones genéricas que facilitan el trabajo de acceso a datos con las entidades mapeadas, además se definen los tipos de datos de cada una de ellas.
- **Entidades:** Conjunto de clases mapeadas por el *Entity Framework* que representan cada una de las entidades existentes en la base de datos.

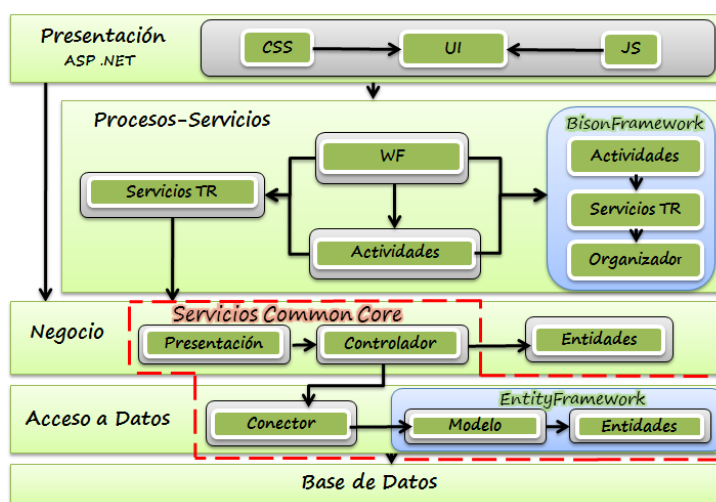


Ilustración 5 Arquitectura de los servicios propuesta.

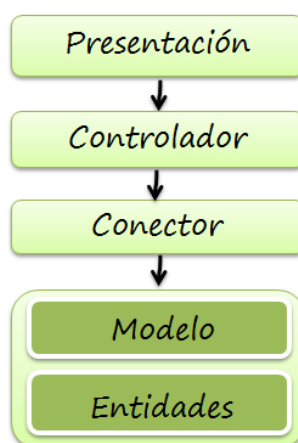
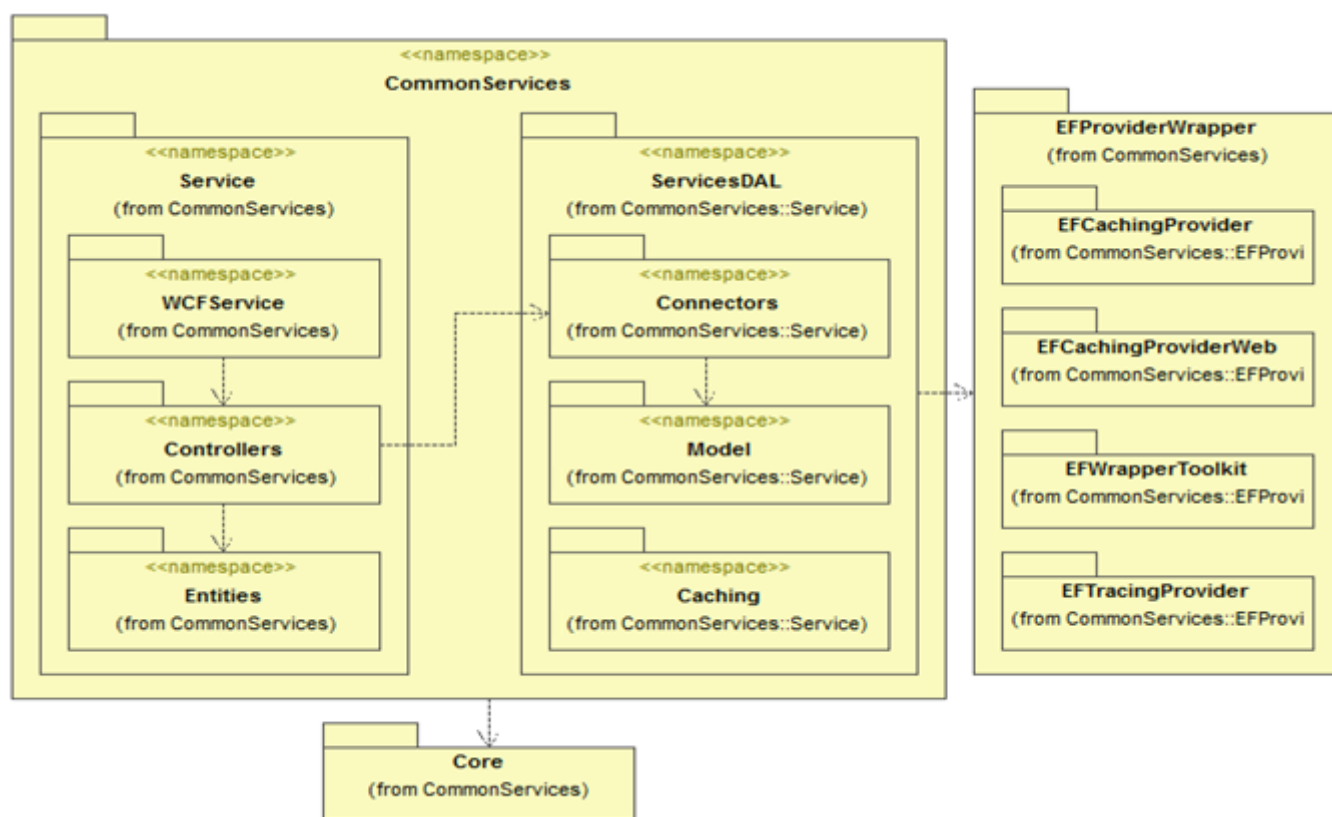


Ilustración 6 Estructura de los servicios.

En la **Ilustración 6** se muestran las capas concretadas para la arquitectura de cada uno de los servicios y su interacción con el modelo y las entidades generados por el *Entity Framework*.

### 3.3. ESTRUCTURA

La solución se ha desarrollado con una estructura que encapsula en paquetes los servicios y las entidades pertenecientes a estos, con el objetivo de formar estructuras concretas, funcionales y compactas para cada uno de los servicios y darle solución a los requerimientos particulares de estos. En la **Ilustración 7** se puede observar la estructura de la solución.



**Ilustración 7 Diagrama de Paquetes para los servicios.**

A continuación se describe el diagrama de paquetes que responde a una estructura estándar para todos los servicios del *CommonServices*. La solución está estructurada en diferentes paquetes que agrupan funcionalidades, comportamientos y relaciones con otros paquetes acorde a la función que realizan dentro de la solución. El paquete *CommonServices* es el encargado de agrupar todos los servicios que se

incluyen en la solución, cada servicio está compuesto por los paquetes “*Service*” y “*ServiceDAL*”; en el diagrama se asocia esta estructura al servicio con los paquetes “*Service*” y “*ServiceDAL*” respectivamente.

El *Service* contiene los elementos pertenecientes a la capa de Presentación y la capa de Controladores. En este paquete se implementan los controladores que atenderán las funcionalidades del servicio, contiene las entidades de negocio y aquellas que por concepto de composición se encuentran estrictamente relacionadas con el servicio y se definen las interfaces que tienen que ser implementadas por los conectores que vinculan el servicio con la fuente de datos correspondientes. La comunicación entre el *Service* y el *ServicesDAL* se manifiesta mediante la utilización por parte del controlador de las funcionalidades brindadas por el conector.

El *ServicesDAL* encapsula los conectores que son los responsables del acceso a datos, junto al modelo y las entidades mapeadas por el *Entity Framework*. También contiene el paquete de *Caching* que interactúa con el paquete *EFProviderWrapper* el cual brinda funcionalidades de trazas, seguimientos y cache para los servicios. El paquete *Caching* implementa y configura el *EFCachingProvider* para almacenar de forma temporal los datos que son consultados con más frecuencia de la base de datos disminuyendo el tiempo de respuesta de la solución.

El paquete *Core* incluye las funcionalidades necesarias para el registro de trazas y el manejo de las instancias, este paquete se relaciona con el *CommonServices* el cual utiliza las funcionalidades que este proporciona.

## 3.4. PATRONES UTILIZADOS

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos, basadas en la experiencia y que se ha demostrado que funcionan (35).

Existen problemas durante el desarrollo de software que se repiten o que son análogos, que responden a un cierto patrón. Por lo que es deseable tener una colección de dichos patrones con las soluciones más óptimas para cada caso a modo de buenas prácticas.

Durante el desarrollo de los servicios comunes se utilizaron los siguientes patrones de diseño:

- **Encapsulación:** propone esconder algunos componentes, permitiendo sólo accesos estilizados al objeto. Se hace uso de este patrón en casi todas las clases que componen al sistema permitiendo

que estas solo posean como elementos públicos aquellos que son exclusivamente necesarios. Un ejemplo de esto lo tenemos en el siguiente fragmento de código.

```
public bool ChangeTrackingEnabled
{
    get { return _changeTrackingEnabled;}
    set { _changeTrackingEnabled = value;}
}
```

- **Subclase:** propone heredar miembros por defecto de una superclase, seleccionando la implementación correcta a través de resoluciones sobre qué implementación debe ser ejecutada. Se puede encontrar este patrón con más fuerza en las entidades de negocio que por su conceptualización, las funciones y la información que almacenan pueden estar diferenciadas en cierta medida. En el siguiente fragmento de código tenemos un ejemplo de este patrón.

```
public partial class NTIPOPROCESO: Encoder, IObjectWithChangeTracker, INotifyPropertyChanged
{
    ...
}
```

- **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.

```
[Serializable]
public class NotNullException : Exception
{
    public override string Message
    {
        get { return "Ha introducido un valor nulo el cual no lo puede ser";}
    }
}
```

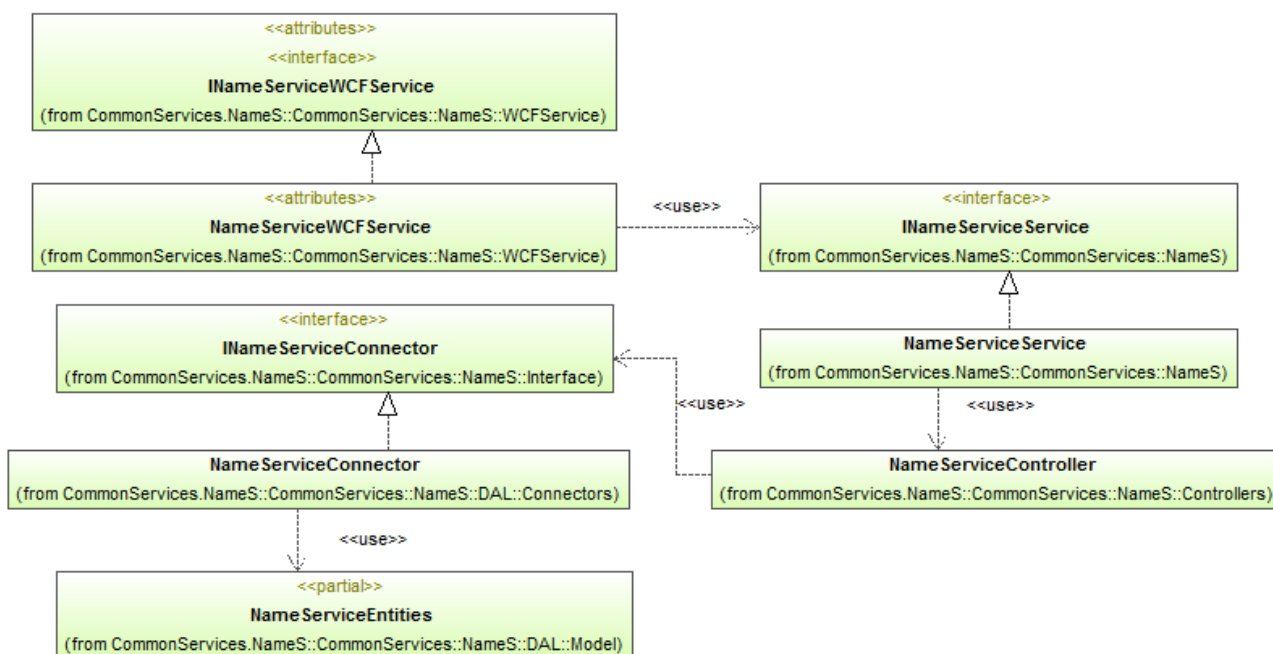
- **Fábrica:** provee de una interfaz para crear familias de objetos relacionados o dependientes sin especificar los tipos concretos de clases. Su uso se encuentra centrado a la creación de los conectores correspondientes al acceso a datos que se esté utilizando, así como en la obtención de los servicios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

```

public int UpdateEncoder<T>(T encoder) where T: class
{
    try
    {
        GetObjectSet<T>().Attach(encoder);
        return ObjectContext.SaveChanges();
    }
    catch (Exception ex)
    {
        Manager.LoggerInstance.Write(ex.ToString());
        throw;
    }
}
    
```

## 3.5. DIAGRAMAS DE LAS CLASES

En la **Ilustración 8** se observa el diagrama de clases genérico para cada servicio, donde están representadas las clases que intervienen en el proceso del servicio. Están representadas todas las clases de cada una de las capas en que están estructurados arquitectónicamente los servicios.



**Ilustración 8 Diagrama de Clases.**



## 3.5.1. DESCRIPCIÓN DE LAS CLASES ENTIDADES

- **NameServiceWCFService:** Clase que implementa de la interfaz *INameServiceWCFService* para realizar la función de exponer el servicio con WCF. Esta consume las funcionalidades del servicio de la clase principal *NameServiceService* brindándole transparencia a la presentación del servicio.
- **NameServiceService:** Clase que implementa de la interfaz *INameServiceService* para funcionar como centro del servicio ya que expone las funcionalidades del servicio para ser consumidas por *NameServiceWCFService*. *NameServiceService* usa y consume la clase *NameServiceController* que es la encargada de manipular los datos en la capa Controlador.
- **NameServiceController:** Se encuentra en la capa Controlador siendo la encargada de implementar la lógica de negocio del servicio o en caso de ser un servicio de acceso a datos realiza función de mediador entre las capas Acceso a Datos y Presentación. Implementa de la interfaz *INameServiceController*.
- **NameServiceConnector:** Implementa de *INameServiceConnector* y es el encargado de interactuar con el modelo y las entidades mapeadas por el *Entity Framework*, realizando así todo el trabajo de acceso a datos del servicio. Este desarrolla las funcionalidades del servicio.
- **NameServiceEntities:** Clase que representa e inicializa al conjunto de entidades mapeadas y funcionalidades generadas para el acceso a datos.

## 3.6. MODELO DE DATOS

A continuación se presenta un fragmento del modelo de datos utilizados por los servicios comunes, el cual es una fracción del modelo de datos utilizados por el sistema. Este modelo es utilizado por el *EncoderService* tanto para gestionar información de la base de datos como para realizar operaciones según la lógica del negocio lo necesite. En la **Ilustración 9** se puede observar una imagen del modelo de datos. Para analizar más el modelo de datos ver **Anexo III**.



DESCRIPCION	<i>Varchar2(4000)</i>	No	Descripción del tipo de proceso.
FECHAREGISTRO	<i>Date</i>	No	Fecha en que se registra el tipo de proceso.
IDAREAAPLICACION	<i>NUMBER(4, 0)</i>	No	Identificador del área de la aplicación.

**Tabla 3 Descripción de la entidad NTIPOPROCESO.**

<b>Nombre de la Entidad:</b> NTIPOTRAMITE			
<b>Descripción de la Entidad:</b> Nomenclador de tipos de trámite.			
<b>Servicio:</b> <i>EncoderService</i> .			
Atributo	Tipo de Dato	Nulo	Descripción
IDTIPOTRAMITE	<i>NUMBER(4, 0)</i>	No	Identificador para el nomenclador tipo de trámite.
DESCRIPCION	<i>Varchar2(4000)</i>	No	Descripción del tipo de trámite.
FECHAREGISTRO	<i>Date</i>	No	Fecha en que se registra el tipo de proceso.

**Tabla 4 Descripción de la Entidad NTIPOTRAMITE**

Para ver una descripción más detallada de las entidades del modelo de datos para el servicio *EncoderService* ver **Anexo IV**.

### 3.8. CONCLUSIONES

- Con la definición de la arquitectura y de los patrones de diseño queda implícito las ventajas que estos proporcionan en la flexibilidad de la ejecución de la propuesta de solución, lográndose una independencia entre las capas definidas.
- Se definió la estructura concreta del paquete, así como la de cada uno de los servicios teniendo en cuenta las necesidades comunes de los trámites y procesos desarrollados por cada uno de los sistemas.
- Se realizó un estudio del diseño del modelo de datos del paquete, para lograr una buena interacción con el *Entity Framework*.

## Capítulo 4 IMPLEMENTACIÓN Y PRUEBA

### 4.1. INTRODUCCIÓN

En el presente capítulo se describe la implementación de los servicios a través del diagrama de componentes y los estándares de codificación; se describe la manera en que se realiza el tratamiento de los errores. Además se expone la distribución de los nodos necesarios para el despliegue de la aplicación. Finalmente se muestra la realización de pruebas unitarias y de carga con el objetivo de asegurar la calidad de la solución.

### 4.2. ESTÁNDARES DE CODIFICACIÓN

La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema. El mantenimiento del código es la facilidad con que el *software* puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores o mejorar el rendimiento. Aunque la legibilidad y el mantenimiento son el resultado de muchos factores, un aspecto del desarrollo de *software* en la que todos los programadores influyen especialmente es en la técnica de codificación. El mejor método para asegurar que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones del código de rutina.

Un patrón de nomenclatura coherente es uno de los elementos más importantes de la previsibilidad y el descubrimiento de una biblioteca de clases administradas. El uso generalizado y la comprensión de estas directrices de denominación deberían eliminar muchas de las preguntas más comunes.

#### Estilos para la capitalización

Se utilizaron los siguientes 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. Se pueden utilizar los identificadores de *Pascal case* en caso de tres o más caracteres. Por ejemplo: *BackColor* (36)
- **Camello:** La primera letra en el identificador se pone en minúscula y la primera letra de cada subsiguiente palabra concatenada en mayúscula. Por ejemplo: *backColor* (36)

- **Mayúscula:** Todas las letras en el identificador se capitalizan. Esta convención se utiliza sólo para los identificadores que constan de dos o menos letras. Por ejemplo: *System.IO*, *System.Web.UI* (36)
- **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 *namespaces*<sup>20</sup> que solo se diferencien con en el uso de las mayúsculas.
  - No utilizar funciones con nombres de parámetros que se diferencian solo en el uso de las mayúsculas
  - No utilizar *namespaces* con nombres de clases que se diferenciaran solo en el uso de las mayúsculas.
  - No crear clases con propiedades que se diferencian solo en el uso de las mayúsculas.
  - No crear clases con métodos que se diferencien solo en el uso de las mayúsculas. (36)

Para nombrar las clases no se recomienda usar los nombres de los *namespaces* y otras clases comúnmente usadas, por ejemplo *System*, *Collections* o *Forms*.

- **Evitando confusión de nombre y tipo:** Distintos lenguajes de programación usan diversos términos para declarar los principales tipos de identificadores. Los diseñadores de librerías de clases deben definir una terminología de lenguaje específica. Fueron utilizados nombres que describen a sus identificadores en vez de nombres que describen el tipo de identificador.

Con el objetivo de incrementar la legibilidad del código también se emplearon comentarios en todas las declaraciones de clases y funciones más complejas. También se organizó el código de forma estructurada, en bloques de código, para una mejor lectura del mismo. (4)

### 4.3. TRATAMIENTO DE ERRORES

En los servicios comunes para lograr una respuesta rápida ante cualquier error se realizó un estudio de errores, para lanzar excepciones en un lenguaje de fácil comprensión para los usuarios. Se generó un paquete de excepciones, para informarle al desarrollador los diversos errores que cometa. En cada uno de

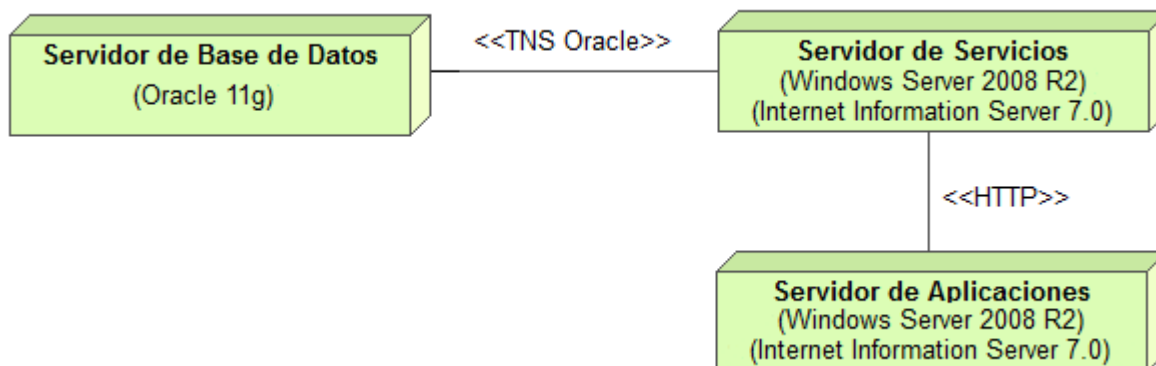
---

<sup>20</sup> Define el nombre de un espacio de trabajo que puede ser un componente o proyecto.

los servicios se evitan, minimizan y tratan los posibles errores. Los servicios utilizan un componente para el seguimiento de errores que permite la escritura de cualquier excepción en un archivo el cual puede ser controlado para mandar alertas para corrección de los errores existentes.

## 4.4. DIAGRAMA DE DESPLIEGUE

“Un diagrama de despliegue muestra la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos”. La vista de despliegue “representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación”. Un nodo es un recurso de ejecución tal como un servidor, computadora, un dispositivo o memoria. (35)



**Ilustración 10 Diagrama de Despliegue**

En la **Ilustración 10** se presenta el Diagrama de Despliegue para la solución, donde se tiene el Servidor de Aplicaciones que contiene las aplicaciones del Sistema de Identificación de Inmigración y Extranjería de Cuba, las cuales consumen los Servicios Comunes a través de una comunicación por el protocolo de transporte HTTP con el Servidor de Servicios. Este último interactúa con el Servidor de base de datos a través del protocolo de *Oracle* “*TNS Oracle*” para realizar todas las funciones de acceso a dato.

## 4.5. IMPLEMENTACIÓN

La Implementación es el flujo de trabajo donde se desarrolla el sistema en términos de componentes, los cuales pueden ser ejecutables, ficheros de código fuente, scripts, librerías, etc. En este flujo de trabajo se desarrolla la arquitectura y el sistema como un todo, definiendo y aplicando patrones y estándares para la

codificación con el fin de alcanzar una óptima organización del código. Los diagramas de componentes y despliegue conforman el modelo de implementación; el diagrama de componentes describe la organización y dependencia de cada uno de los componentes implementados, representándose la distribución de los mismos en los distintos nodos físicos en los que funcionará la aplicación. (37)

## 4.5.1. DIAGRAMA DE COMPONENTE

Los Diagramas de Componentes se encargan de mostrar las organizaciones y dependencias lógicas que existen entre los componentes de *software*, sean estos componentes de código fuente, binarios o ejecutables. Este diagrama contiene componentes y relaciones entre ellos que a su vez son utilizados para agrupar los elementos de modelado (38). A continuación se muestra el diagrama de componentes genérico para todos los servicios.

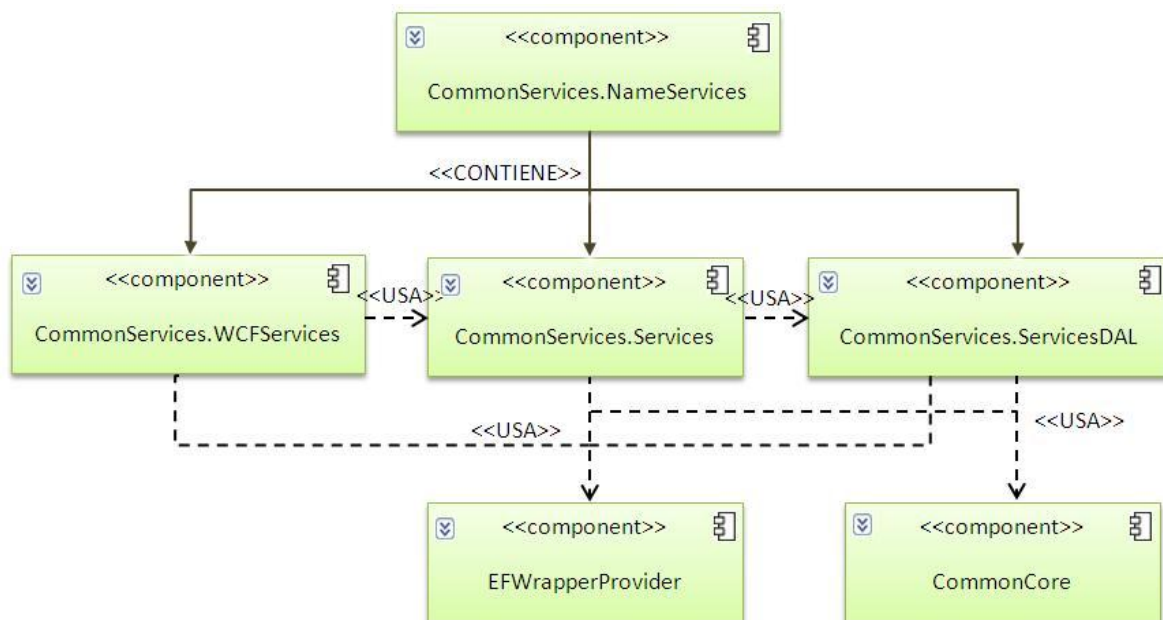


Ilustración 11 Diagrama de Componentes

El *Service* contiene tres componentes principales: *WCFService* que incluye todo lo referente a la configuración y a la exposición del servicio, *Service* que engloba toda la lógica del servicio y *ServiceDAL* que abarca el trabajo de acceso a datos. Estos dos últimos componentes usan el *Core* para el trabajo de seguimiento de error y la identificación de instancias. *Service.DAL* utiliza *EFWrapperProvider* el cual

permiten guardar en memoria la información de las peticiones anteriores para ser reutilizadas, además de almacenar las trazas de estas peticiones.

## 4.6. PRUEBAS

Durante todas las etapas del desarrollo de *software* las pruebas son fundamentales, a partir de ellas es posible controlar que los productos cumplan con los requisitos funcionales, además de verificar y revelar la calidad de un producto *software*. Las pruebas constituyen una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente.

Se realiza esta actividad con el objetivo de verificar que los requerimientos de *software* han sido satisfechos a través de pruebas de unidad, carga y aceptación.

### 4.6.1. PRUEBAS UNITARIAS

Las pruebas unitarias representan una forma de probar un fragmento de código por lo que tiene gran importancia para los desarrolladores y el equipo de pruebas. Las pruebas unitarias de *Microsoft* son una herramienta muy importante para el personal de pruebas, sobre todo para los desarrolladores. Aunque el tamaño de la "unidad" que se prueba puede variar, la herramienta para pruebas del *Visual Studio 2010* permite generar pruebas unitarias específicas para métodos y funcionalidades. (39)

#### 4.6.1.1. DISEÑO DE CASOS DE PRUEBAS

En la ingeniería del *software*, los casos de pruebas son un conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio. Con el propósito de comprobar que todos los requisitos de una aplicación son revisados, debe haber al menos un caso de prueba para cada requisito. Lo que caracteriza un diseño de caso de prueba es que hay una entrada conocida y una salida esperada, los cuales son formulados antes de que se ejecute la prueba. (40)

A continuación se muestra el diseño de caso de prueba para el *EncoderService*. Para ver los diseños de los servicios restantes ver **Anexo V**.



Escenario	Descripción	Parámetros			Respuesta del sistema	Flujo Prueba
<i>GetEncoder</i>	Devuelve los datos del nomenclador correspondiente.	<i>Tipo de Nomenclador</i>			Lista de nomencladores existentes para las áreas del MININT	
		NAREASMININT				
<i>GetEncoderByRange</i>	Devuelve los datos del nomenclador según un rango pasado por dos enteros.	<i>Tipo de Nomenclador</i>	<i>Inicio</i>	<i>Tamaño</i>	Lista de los primeros 2 nomencladores existentes para las áreas del MININT	
		NAREASMININT	0	2		
<i>GetEncoderCount</i>	Cuenta los datos del nomenclador correspondiente.	<i>Tipo de Nomenclador</i>			Cantidad de nomencladores existentes para las áreas del MININT	
		NAREASMININT				
<i>GetSimilarEncoder</i>	Busca los datos del nomenclador con datos similares al pasado.	<i>Tipo de Nomenclador</i>			Lista de nomencladores existentes con datos similares para las áreas del MININT	
		NAREASMININT (idArea=1)				
<i>GetSimilarEncoder</i>	Busca los datos del nomenclador	<i>Tipo de Nomenclador</i>			Lista de nomencladores	

Crear el objeto de entrada y compararlo con el resultado obtenido de la invocación del servicio.

	con datos similares al pasado.	NAREASMININT (idArea=20)			s vacía.
<i>GetSimilarEncoderByRange</i>	Busca los datos del nomenclador con datos similares al pasado según un rango pasado por dos enteros.	<i>Tipo de Nomenclador</i>	<i>Inicio</i>	<i>Tamaño</i>	Lista de los primeros 2 nomencladores existentes para las áreas del MININT
		NAREASMININT	0	2	
<i>GetSimilarEncoderCount</i>	Cuenta los datos del nomenclador con datos similares al pasado.	<i>Tipo de Nomenclador</i>			Cantidad de nomencladores existentes para las áreas del MININT
		NAREASMININT			

Tabla 5 Diseño de caso de prueba para el servicio *EncoderService*

## 4.6.1.2. RESULTADOS DE LAS PRUEBAS

Para el desarrollo del paquete se tuvo en cuenta 3 iteraciones. En ellas se realizaron pruebas unitarias de cada uno de los requisitos de los servicios. A continuación se muestra el código de uno de los métodos al que se le realizó la prueba unitaria para verificar su correcto funcionamiento. Se aplicaron las pruebas al servicio de nomencladores, a cada funcionalidad de este se le generan los resultados esperados y se comparan con los resultados arrojados de la invocación del servicio. Para ver los resultados de las pruebas de los demás servicios ver **Anexo VI**.

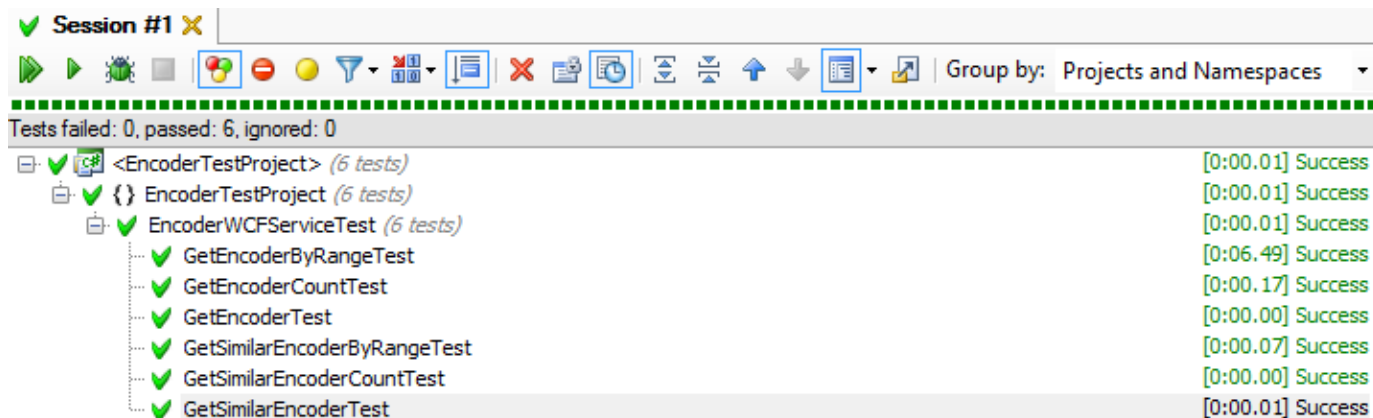
```
[TestMethod()]
public void GetEncoderTest()
{
    var target = new EncoderWCFService(); // TODO: Initialize to an appropriate value
    const EncoderType encoderType = EncoderType.NAREASMININT;

    var fecha = new DateTime(2011,5,16,8,35,0,0);

    var temp = new NAREASMININT()
        {DESCRIPCION = "Area D", IDAREASMININT = 4, FECHACIERRE = fecha, FECHAREGISTRO = fecha};
    var temp1 = new NAREASMININT()
        {
            DESCRIPCION = "Area A",
            IDAREASMININT = 1,
            FECHACIERRE = new DateTime(2011, 5, 2),
            FECHAREGISTRO = new DateTime(2011, 5, 2)
        };
    var temp2 = new NAREASMININT()
        {
            DESCRIPCION = "Area B",
            IDAREASMININT = 2,
            FECHACIERRE = new DateTime(2011, 5, 3),
            FECHAREGISTRO = new DateTime(2011, 5, 3)
        };
    var temp3 = new NAREASMININT()
        {
            DESCRIPCION = "Area C",
            IDAREASMININT = 3,
            FECHACIERRE = new DateTime(2011, 5, 4),
            FECHAREGISTRO = new DateTime(2011, 5, 4)
        };
    var final = true;
    var expected = new List<Encoder>() { temp, temp1, temp2, temp3 }; // TODO: Initialize to an appropriate value
    var actual = target.GetEncoder(encoderType);

    if (actual.Count!=expected.Count)
    {
        final = false;
    }
    for (var i = 0; i < actual.Count; i++)
    {
        if (((NAREASMININT)actual[i]).DESCRIPCION!=((NAREASMININT)expected[i]).DESCRIPCION)
        {
            final = false;
        }
        if (((NAREASMININT)actual[i]).FECHACIERRE != ((NAREASMININT)expected[i]).FECHACIERRE)
        {
            final = false;
        }
        if (((NAREASMININT)actual[i]).FECHAREGISTRO != ((NAREASMININT)expected[i]).FECHAREGISTRO)
        {
            final = false;
        }
        if (((NAREASMININT)actual[i]).IDAREASMININT != ((NAREASMININT)expected[i]).IDAREASMININT)
        {
            final = false;
        }
    }
    Assert.IsTrue(final);
}
```

Ilustración 12 Prueba unitaria del *EncoderService*.



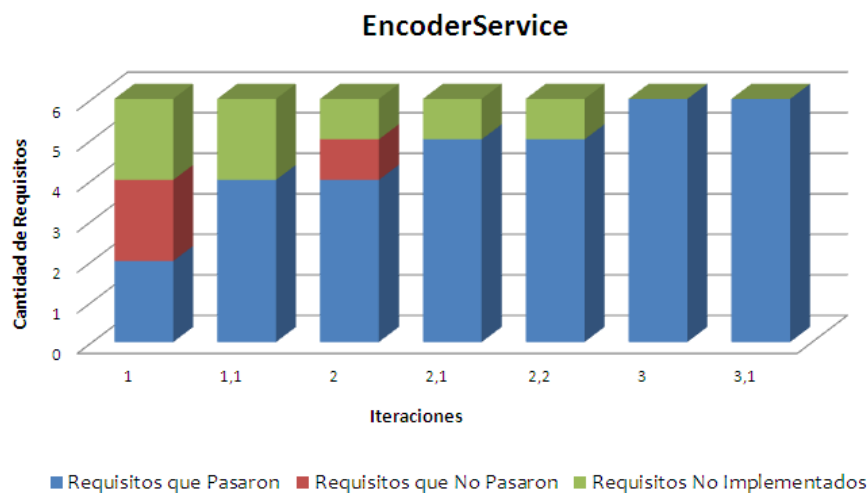
**Ilustración 13** Resultado final de las pruebas a *EncoderService*.

En la **Tabla 6** se muestran los resultados de cada una de las pruebas que se hicieron al *EncoderService*.

<i>EncoderService</i>							
Requisitos	1	1.1	2	2.1	2.2	3	3.1
Mostrar los nomencladores existentes dado una clasificación de nomenclador.	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó
Contar los nomencladores existentes dado una clasificación de nomenclador.	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó
Mostrar los nomencladores existentes dado una clasificación de nomenclador, un número de inicio y una cantidad de total.	No Pasó	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó
Mostrar los datos de un nomenclador similar según parámetros relacionales.	No Pasó	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó

Contar los datos de un nomenclador similar a los parámetros de la búsqueda realizada.	No Evaluado	No Evaluado	No Pasó	Pasó	Pasó	Pasó	Pasó
Mostrar los datos de un nomenclador similar según parámetros y un número de inicio y una cantidad de total.	No Evaluado	No Evaluado	No Evaluado	No Evaluado	No Evaluado	Pasó	Pasó

**Tabla 6 Pruebas de Requisitos por Iteración del servicio *EncoderService***



**Ilustración 14 Gráfica de las pruebas de Requisitos por Iteración del servicio *EncoderService***

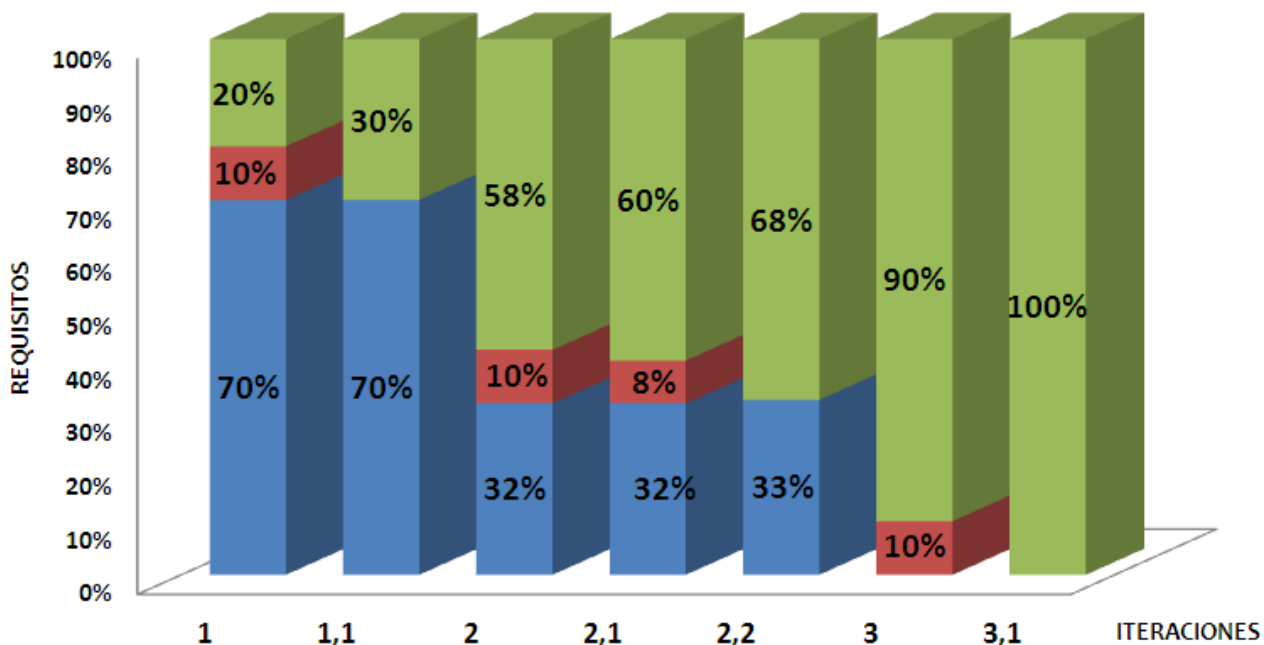
### 4.6.1.3. CONCLUSIONES PRUEBAS UNITARIAS

En las 3 iteraciones de desarrollo se realizaron un total de 7 iteraciones de pruebas en las cuales se encontraron y corrigieron errores llegando a un resultado positivo en todas las funcionalidades. El resultado de la prueba se muestra a continuación.

ITERACIONES	1	1,1	2	2,1	2,2	3	3,1
Requisitos No Implementados	28	28	13	13	13	0	0
Requisitos que No Pasaron	4	0	4	3	0	4	0

Requisitos que Pasaron	8	12	23	24	27	36	40
------------------------	---	----	----	----	----	----	----

Tabla 7 Pruebas realizadas a cada requisito funcional de los servicios por cada iteración del desarrollo



■ Requisitos No Implementados ■ Requisitos que No Pasaron ■ Requisitos que Pasaron

Ilustración 15 Gráfica de las pruebas realizadas a cada requisito funcional de los servicios por cada iteración del desarrollo

Luego de realizar un análisis estadístico de los resultados de las pruebas en cada iteración se concluye que el paquete de servicios desarrollado cumple con todas las funcionalidades y requerimientos definidos, teniendo resultados correctos y buen rendimiento.

#### 4.6.2. PRUEBAS DE CARGA AL SISTEMA

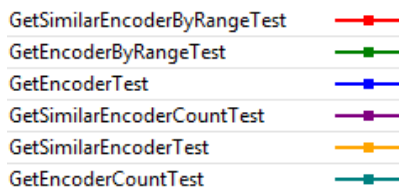
Para probar el funcionamiento del paquete de servicio se realizaron pruebas de carga, las cuales tienen como objetivo fundamental verificar el rendimiento de las funcionalidades ante diferentes situaciones en las que se puede encontrar el sistema. Se le realizaron pruebas de carga al *EncoderService* las cuales se comentan a continuación:

El servicio se publicó en dos máquinas las cuales presentan el hardware descrito en la **Tabla 8**. Las características de ambas máquinas son las que el Ministerio del Interior tiene prevista sean las que se utilicen en las máquinas para despliegue del sistema.

Hardware	Máquina 1	Máquina 2
<b>Microprocesador</b>	Intel(R) Xeon(R) CPU 3.00 GHz	Intel(R) Core(TM) 2 Duo CPU E7300 a 2.66 GHz
<b>Memoria RAM</b>	3 GB	2 GB
<b>Ancho de Banda</b>	100 Mbps	100 Mbps
<b>Disco Duro</b>	320 GB	80 GB

**Tabla 8 Descripción del hardware de las máquinas de prueba.**

Para realizar las pruebas de carga se utilizó la herramienta del *Visual Studio 2010 (Load Test)* a la cual se le definió realizar 25 hilos de peticiones constantes que son las estimadas que deben atender los servidores de aplicación de las provincias, estas peticiones fueron constantes por 10 minutos al servicio. Los resultados de las pruebas realizadas se basan en una tabla con los datos de pruebas que contienen el total de pruebas, los errores y el promedio de tiempo por pruebas. También se muestran gráficos de comportamiento del promedio de pruebas por tiempo. La leyenda de los gráficos se muestra en la **Ilustración 16**.



**Ilustración 16 Leyenda gráficos.**

**En la máquina 1:**

▼ Test Results

Name	Scenario	Total Tests	Failed Tests (% of total)	Avg. Test Time (sec)
<a href="#">GetSimilarEncoderByRangeTest</a>	Scenario 1	11.289	0 (0)	0,044
<a href="#">GetSimilarEncoderCountTest</a>	Scenario 1	10.551	0 (0)	0,042
<a href="#">GetEncoderTest</a>	Scenario 1	11.334	1 (0,0088)	0,042
<a href="#">GetEncoderCountTest</a>	Scenario 1	11.269	0 (0)	0,035
<a href="#">GetSimilarEncoderTest</a>	Scenario 1	10.556	0 (0)	0,033
<a href="#">GetEncoderByRangeTest</a>	Scenario 1	11.139	0 (0)	0,032

Tabla 9 Resultados de la prueba de carga.

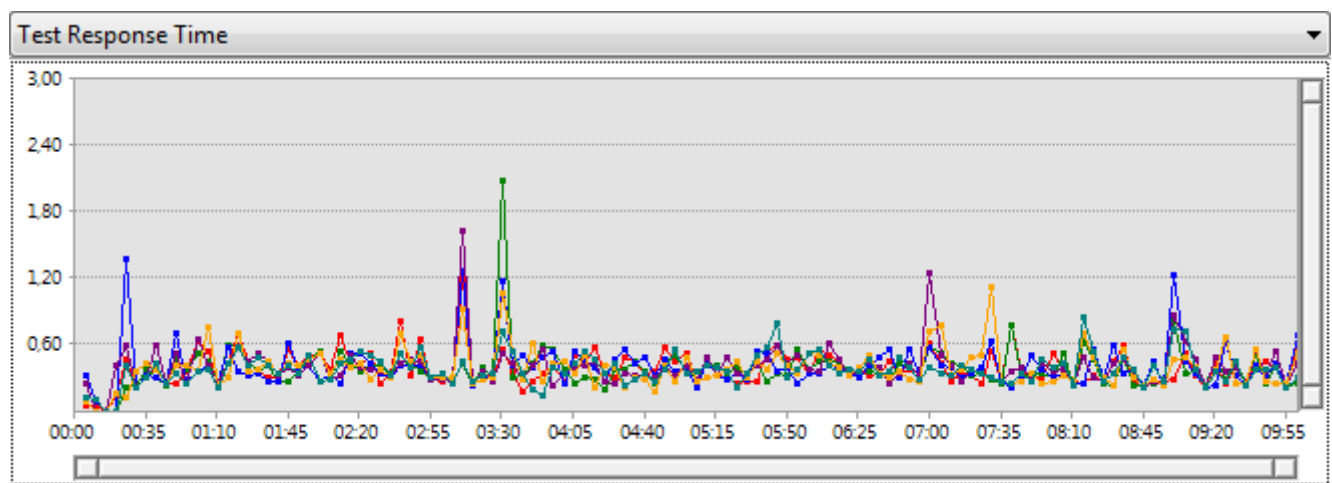


Ilustración 17 Gráfico de tiempo de respuesta de las funcionalidades respecto al tiempo de pruebas total.

En la máquina 2:

Name	Scenario	Total Tests	Failed Tests (% of total)	Avg. Test Time (sec)
<a href="#">GetSimilarEncoderByRangeTest</a>	Scenario 1	10.022	6 (0,060)	0,045
<a href="#">GetSimilarEncoderCountTest</a>	Scenario 1	10.610	5 (0,047)	0,043
<a href="#">GetEncoderTest</a>	Scenario 1	10.702	4 (0,037)	0,042
<a href="#">GetEncoderCountTest</a>	Scenario 1	9.682	4 (0,041)	0,042
<a href="#">GetSimilarEncoderTest</a>	Scenario 1	10.657	3 (0,028)	0,040
<a href="#">GetEncoderByRangeTest</a>	Scenario 1	10.615	2 (0,019)	0,037

Tabla 10 Resultados prueba de carga.



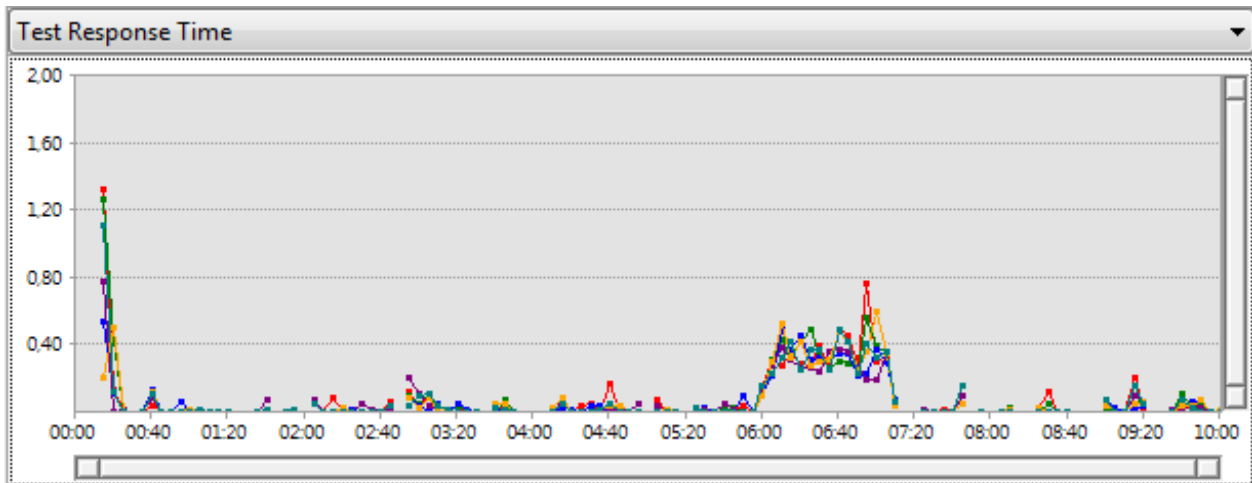


Ilustración 18 Gráfico de tiempo de respuesta de las funcionalidades respecto al tiempo de pruebas total.

### 4.6.2.1. CONCLUSIONES PRUEBAS DE CARGA AL SISTEMA

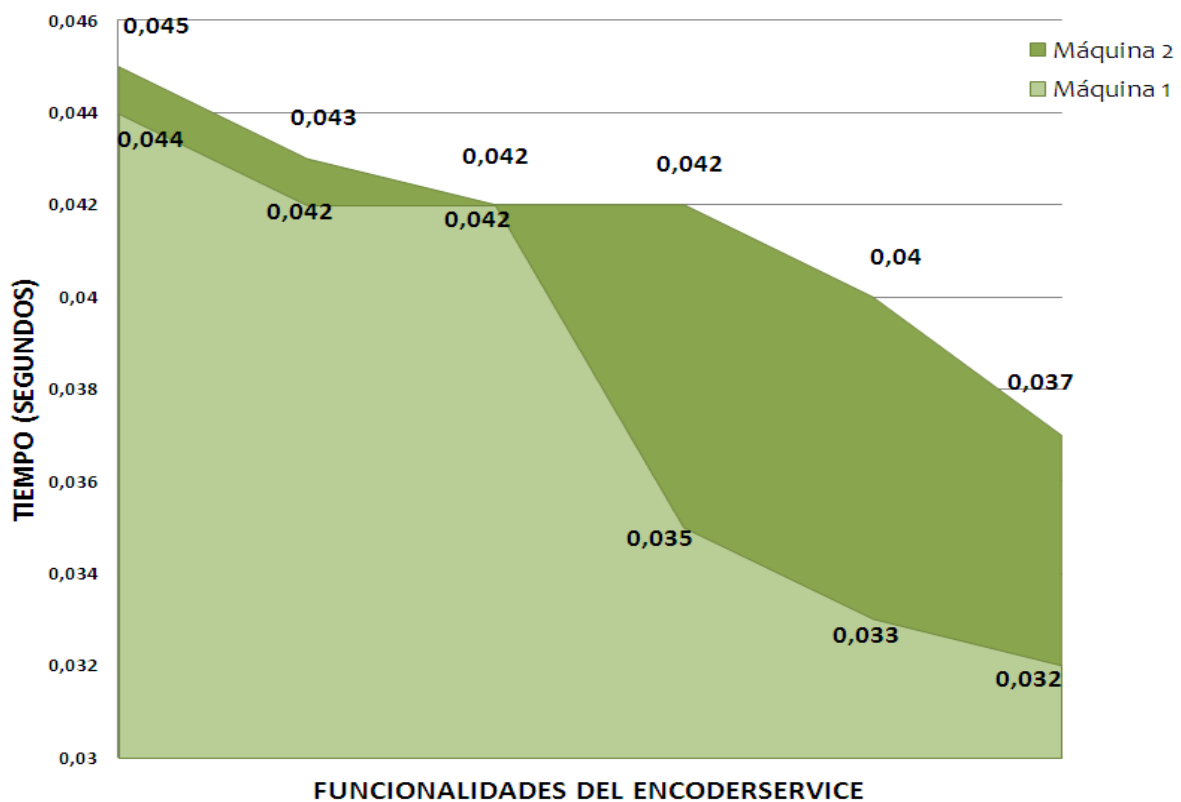


Ilustración 19 Gráfica del resultado de las Pruebas de Carga al Sistema

Las pruebas de carga realizadas demuestran que las pc 1 y 2 soportan el flujo de peticiones para los servidores provinciales ya que ambas mostraron tiempos de respuestas bajos en todas las peticiones. Se observó mejor rendimiento en la máquina 1 que es la que se propone se utilice para los servidores, pero la pc 2 puede ser una buena opción en el caso de separar el hospedaje del servidor de aplicaciones y el de servicios.

Los tiempos de respuestas obtenidos en las pruebas del sistema cumplen con las necesidades y expectativas de rendimiento del paquete de servicios.

### 4.6.3. ACEPTACIÓN DE LOS SERVICIOS

El subsistema Sistema de Personalización de Documentos de Identificación (SPDI) perteneciente al SUIN, fue liberado y sometido a pruebas y validaciones por parte del cliente y de la entidad de calidad CALISOFT<sup>21</sup>, el SPDI utiliza algunos de los principales servicios del CommonCore, como es el caso del servicio de nomencladores, buscador de personas y el servicio de petición de procesos. Al probar el SPDI fueron probados y validados los servicios antes mencionados agilizando los resultados de la solución. El Acta de aceptación de CALISOFT se encuentra en el **Anexo VII**.

### 4.7. REUTILIZACIÓN DE SERVICIOS

En función de exponer el nivel de reutilización de los servicios comunes se realizó un estudio de los módulos de los distintos subsistemas del SIIEC que utilizan cada uno de los servicios, en la tabla siguiente se muestran los resultados.

SUBSISTEMAS	SERVICIOS						
	<i>Person Manager</i>	<i>Process Request</i>	<i>Search Address</i>	<i>Encoder</i>	<i>Search Person</i>	<i>Operative Record</i>	<i>TOTAL</i>
<b>SUIN</b>							
Administración	X	X	X	X	X	X	6
Nacimiento	X	X	X	X	X	X	6
Defunción	X	X		X	X	X	5
Registro Electoral	X	X	X	X	X	X	6

<sup>21</sup> Entidad perteneciente a la UCI que brinda servicios para la realización de revisiones y pruebas de calidad a productos de software.

Irregularidades	X	X		X	X	X	5
Subsistema SPDI		X		X	X		3
<b>SIEC</b>							
Enfrentamiento	X	X		X	X	X	5
Circulado		X		X	X	X	3
Asuntos Oficiales		X		X		X	3
Proceso de Visas de Trámites por Asuntos Particulares para Extranjeros	X	X		X	X	X	5
Proceso de Pasaporte por Asuntos Particulares	X	X		X	X	X	5
<b>MEDIA</b>	73%	100%	27%	100%	91%	91%	93%

Tabla 11 Reutilización de servicios por módulos.

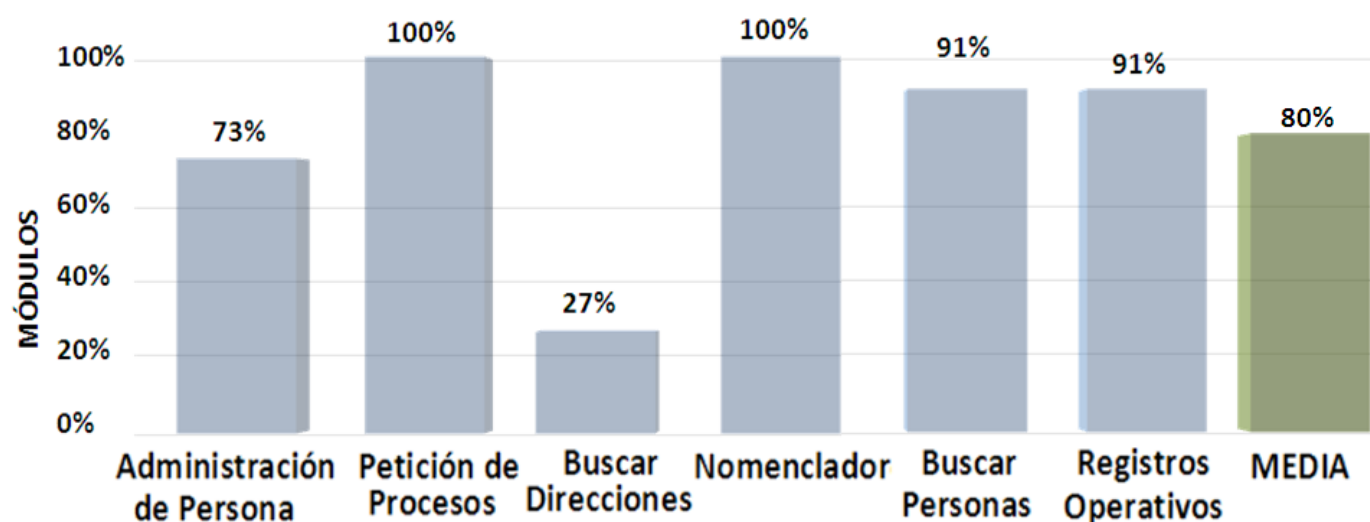


Ilustración 20 Gráfica de reutilización de servicios por módulos.

En los resultados se observa que cada servicio es utilizado como mínimo por el 27% de los módulos, dando una media de un 80% de utilización de los servicios por módulo, demostrándose así el alto nivel de reutilización del paquete de servicios dentro del SIEC cumpliéndose con los objetivos propuestos.

## 4.8. BENEFICIOS DEL PAQUETE

- Mejora en el proceso de búsqueda y actualización de la información.
- Disminución de la cantidad de personas empleadas en el desarrollo.
- Reutilización de funcionalidades comunes.
- Estandarización en la comunicación entre los subsistemas y módulos de la solución.
- Los servicios comunes posibilitan a los desarrolladores de las capas superiores abstraerse de todo el trabajo que se realiza en las capas inferiores del sistema.

## 4.9. CONCLUSIONES

- 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 validaron los requisitos no funcionales correspondientes a los servidores que publican los servicios a partir de las pruebas de carga al sistema.
- Se realizaron las pruebas al paquete de servicios a partir de los requisitos funcionales demostrándose con las pruebas de aceptación, unitarias y de sistema la calidad de la solución.
- Quedó demostrado que con el desarrollo del paquete de servicios se exponen funcionalidades comunes dentro de la Arquitectura Orientada a Servicios definida por el Sistema de Identificación, Inmigración y Extranjería de Cuba que permite su reutilización en los procesos existentes.

## CONCLUSIONES

Después del estudio realizado de los principios arquitectónicos de SOA, de las principales arquitecturas para implementar servicios, de los ORM más utilizados en la actualidad y de implementar todos los servicios teniendo en cuenta una Arquitectura Orientada a Servicios se concluye:

- La mejor opción para el intercambio de información de los procesos comunes en la arquitectura SOA del Sistema de Identificación, Inmigración y Extranjería de Cuba es la implementación de una Arquitectura Orientada a Servicios, utilizando la tecnología WCF de .NET ya que se acopla al ambiente de desarrollo del proyecto y a las necesidades del sistema.
- Se definió que la arquitectura de los servicios para el Sistema de Identificación, Inmigración y Extranjería de Cuba sea por capas bien estructuradas, teniendo en cuenta en el acceso a datos la utilización de *Entity Framework* de .NET como ORM.
- La solución propuesta se ajusta con la metodología, lenguajes y herramientas seleccionadas para el desarrollo del proyecto, las cuales se encuentran entre las más utilizadas y eficaces de la actualidad.
- Se realizó el modelo de dominio que permitió la comprensión de los conceptos con que interaccionan los servicios y los que utiliza la solución para intercambiar los datos entre las distintas capas de la arquitectura.
- A partir del modelo de dominio del sistema se determinaron los requisitos funcionales con los que debe cumplir el paquete de servicios.
- Se validaron los requisitos no funcionales correspondientes a los servidores que publican los servicios a partir de las pruebas de carga al sistema.
- Se realizaron las pruebas al paquete de servicios a partir de los requisitos funcionales demostrándose con las pruebas de aceptación, unitarias y de sistema la calidad de la solución.
- Quedó demostrado que con el desarrollo del paquete de servicios se exponen funcionalidades comunes dentro de la Arquitectura Orientada a Servicios definida por el Sistema de Identificación, Inmigración y Extranjería de Cuba que permite su reutilización en los procesos existentes.

## RECOMENDACIONES

Se recomienda:

- Realizar un estudio profundo de la seguridad en WCF para definir e implementar el esquema de seguridad para la comunicación de los servicios que cumpla con los requerimientos del Ministerio del Interior.
- Exponer servicios afines en la red externa del Ministerio del Interior en aras de propiciar el gobierno en línea de la República de Cuba.
- La generalización de los servicios para otros sistemas existentes en el Ministerio del Interior que permita garantizar la identidad de los ciudadanos cubanos.
- El estudio, una vez desplegado el sistema, del comportamiento del servidor y la posibilidad de balancear la carga del mismo con otras implementaciones de despliegue como *clúster* o granja de servidores, etc.

## BIBLIOGRAFÍA

1. **L., Ing. Maria Eugenia Arevalo.** Maria Eugenia Arevalo Blog. [En línea] 7 de enero de 2010. <http://arevalomaria.wordpress.com/2010/01/07/ejemplo-de-documento-vision-y-alcance/>.
2. **Proyecto Identidad Cuba. UCI.** *Proyecto Técnico. DIE.* [Documento] 2009.
3. —. *Proyecto Técnico. DIR.* [Documento] 2009.
4. **Blanco, Reynier y Salazar, Humberto.** *Documento de Arquitectura.* Ciudad de la Habana : s.n., 2010.
5. **Proyecto Identidad Cuba. UCI.** *Arquitectura de Software.* [Documento] 2010.
6. **Líderes del grupo de Tecnología SOA de Accenture.** *Arquitectura Orientada a Servicios.* s.l. : Un estudio publicado por el Centro de Alto Rendimiento de Accenture (CAR), 2008.
7. **Martín, Yanet Espinol.** SabeLoTodo.com. [En línea] 2007. [Citado el: 30 de 11 de 2010.] <http://www.sabetodo.com/contenidos/EkEuAuVEyISfYPHaVc.php>.
8. **Autores, Grupo de.** [En línea] 26 de Mayo de 2009. [Citado el: 7 de 12 de 2010.] [http://ingenieria.ucaldas.edu.co/auditoria/index.php/SOA\\_%28service\\_oriented\\_architectures%29](http://ingenieria.ucaldas.edu.co/auditoria/index.php/SOA_%28service_oriented_architectures%29).
9. **Barco, Antonio.** *Arquitectura Orientada a Servicios (SOA)* . [En línea] 9 de Junio de 2006. <http://arquitecturaorientadaaservicios.blogspot.com/>.
10. *La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real.* **Microsoft Corporation.** 2006.
11. **Miranda, Emmerson.** *Desarrollo Java.* [En línea] 27 de Julio de 2009. [Citado el: 4 de enero de 2011.] <http://www.emmersonmiranda.net/2009/07/que-es-un-esb.html>.
12. **Campodónico, Ing. Diego.** *Slide.* [En línea] 2008. <http://www.slideshare.net/rdiegoc/introduccion-a-soa>.
13. **Gartner, Inc.** [En línea] 18 de Abril de 2011. <http://procesamientoeventos.blogspot.com/2011/04/definicion-de-bam.html>.
14. **Microsoft.** *Características de WCF.* [En línea] 2011. <http://msdn.microsoft.com/es-es/library/ms731082.aspx>.

15. —. MSDN. *Introducing Windows Communication Foundation in .NET Framework 4*. [En línea] 2010. <http://msdn.microsoft.com/en-us/library/ee958158.aspx>.
16. **WCF, MSDN**. MSDN Library. [En línea] <http://msdn.microsoft.com/es-es/library/ms731082.aspx>.
17. **Microsoft**. Información general sobre seguridad. *MSDN*. [En línea] 2011. <http://msdn.microsoft.com/es-es/library/ms735093.aspx>.
18. —. MSDN. [En línea] 2011. [Citado el: 12 de marzo de 2011.] <http://msdn.microsoft.com/es-es/library/ms735093.aspx>.
19. **Garlan, David y Shawn, Mary**. *An Introduction to Software Architecture*.
20. **Pimentel, Luis Alberto y Rivero, Iósev Pérez**. *ArBaWeb: ARQUITECTURA BASE SOBRE LA WEB*. s.l. : UCI, junio del 2007.
21. **Microsoft**. *Microsoft Application Architecture Guide 2nd Edition*. 2009.
22. **Rojas, José Fabricio**. Desarrollo Alter.Netivo. *Arquitectura MVC*. [En línea] 22 de abril de 2009. [Citado el: 12 de marzo de 2011.] [HTTP://DEVSOFTX.WORDPRESS.COM/2009/04/22/QUE-ES-ASPNET-MVC/](http://DEVSOFTX.WORDPRESS.COM/2009/04/22/QUE-ES-ASPNET-MVC/).
23. **Creative Commons**. LibrosWeb. *La Arquitectura MVC*. [En línea] [http://www.librosweb.es/jobeeet/capitulo4/la\\_arquitectura\\_mvc.html](http://www.librosweb.es/jobeeet/capitulo4/la_arquitectura_mvc.html).
24. *Utilización del patrón Modelo - Vista - Controlador (MVC) en el diseño de software educativo*. **Febe, Ángel Ricardo**. Ciudad de la Habana : s.n., 2006.
25. **Cabrera., Irving Deither Cao**. *XCAD, Generador de Capas de Acceso a Datos*. Habana. Cuba : s.n., 2008.
26. **Microsoft**. MSDN Online. *ADO.NET Entity Framework*. [En línea] <http://msdn.microsoft.com/es-es/library/bb399572.aspx>.
27. —. MSDN Library . *MSDN Library*. [En línea] <http://www.msdn.microsoft.com/en-us..>
28. **Altova Coporation**. Altova. [En línea] <http://www.altova.com/list/xmlschema-dev/201001>.
29. **Oracle Corporation**. Oracle. [En línea] 2010. [Citado el: 12 de enero de 2011.] <http://www.oracle.com/technology/products/database/oracle11g>.



30. **Bob Bryla, Kevin Loney.** *Oracle Database 11g DBA Handbook (Osborne ORACLE Press Series).*
31. **Oracle Corporation.** PL / SQL. [En línea] 2010. [Citado el: 11 de enero de 2011.] [http://www.oracle.com/technology/tech/pl\\_sql](http://www.oracle.com/technology/tech/pl_sql).
32. **Microsoft.** MSDN. *Visual Studio 2010.* [En línea] 2011. [Citado el: 2011 de enero de 2011.] <http://msdn.microsoft.com/es-es/library/6x6bk1f4.aspx>.
33. **Oracle Corporation.** Oracle Data Provider. [En línea] 2010. [Citado el: 12 de marzo de 2011.] <http://www.oracle.com/technology/obe/hol08/dotnet>.
34. **C., Benjamín González.** XML: el lenguaje de los Servicios Web. *DesarrolloWeb.* [En línea] <http://www.desarrolloweb.com/articulos/1574.php>.
35. **Jacobson, Ivar.** *El proceso unificado de desarrollo de Software.* s.l. : Addison-Wesley Iberoamericana España, S.A., 2000.
36. **Proyecto Identidad Cuba.** *Estándares de codificación.* Ciudad de la Habana : s.n., 2009.
37. **Turossi, AUS Gustavo.** Scribd. [En línea] 2008. <http://es.scribd.com/doc/51305810/50/Implementacion>.
38. **Microsoft.** MSDN. [En línea] 2010. [Citado el: 8 de 4 de 2011.] <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.
39. —. MSDN. [En línea] 2011. [Citado el: 12 de marzo de 2011.] [http://msdn.microsoft.com/es-es/library/ms182215\(2v=vs.180\).aspx](http://msdn.microsoft.com/es-es/library/ms182215(2v=vs.180).aspx).
40. —. MSDN. [En línea] [Citado el: 5 de enero de 2011.] [http://msdn.microsoft.com/es-es/library/aa546827\(BTS.10\).aspx](http://msdn.microsoft.com/es-es/library/aa546827(BTS.10).aspx).
41. **CALISOFT.** *Acta de liberación del subsistema SPDI.* Cuba : s.n., 2011.
42. **Reinier Sotes, Sandy Moore.** *Propuesta de integración para los servicios web del Sistema de Emisión de Documentos de Identificación del Centro de Identificación y Seguridad Digital.* Ciudad Habana : Uci, Mayo 2009.
43. **M.Crispino.** Petroglifo. [En línea] 14 de septiembre de 2001. [Citado el: 5 de enero de 2011.] <http://www.concepto.com.uy/petrocsharp/hwnver03.aspx?1,N,0,0,101>. 10.

44. **W3 Schools.** w3schools.com. [En línea] 2010. <http://www.w3schools.com/soap/default.asp>.
45. **Microsoft.** MSDN. [En línea] 2011. [Citado el: 12 de marzo de 2011.] [http://msdn.microsoft.com/es-es/library/ms182215\(2=vs180\).aspx](http://msdn.microsoft.com/es-es/library/ms182215(2=vs180).aspx).

## GLOSARIO DE TÉRMINOS

**CALISOFT:** Entidad perteneciente a la UCI que brinda servicios para la realización de revisiones y pruebas de calidad a productos de software.

**DIE:** Dirección Inmigración y Extranjería

**DIR:** Dirección de Identificación y Registros.

**MININT:** Ministerio del Interior de la República de Cuba.

**SIEC:** Sistema de Inmigración, Extranjería y Ciudadanía.

**SIIEC:** Sistema de Identificación, Inmigración, Extranjería de Cuba.

**SUIN:** Sistema Único de Identificación Nacional.

**SPDI:** Sistema de Personalización de Documentos de Identidad.

**Workflow:** Automatización de los procesos de negocio, durante el cual “documentos”, “información” y “tareas” son pasados de un participante a otro, acorde a un conjunto de reglas. (5)

**WWF:** *Windows Workflow Foundation* (Tecnología para el desarrollo de sistemas basados en *workflow* sobre .NET).

**.NET:** La plataforma .NET de *Microsoft* es un componente de *software* que puede ser añadido al Sistema Operativo *Windows*. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma. Esta solución es el producto principal en la oferta de *Microsoft*, y pretende ser utilizada por la mayoría de las aplicaciones creadas para la plataforma *Windows*. (2)

## ANEXOS

### Anexo I. CATÁLOGO DE REQUISITOS FUNCIONALES

#### RF 1 Permitir la obtención de los nomencladores.

RF1.1 Mostrar los nomencladores existentes dado una clasificación de nomenclador.

RF1.2 Contar los nomencladores existentes dado una clasificación de nomenclador.

RF1.3 Mostrar los nomencladores existentes dado una clasificación de nomenclador, un número de inicio y una cantidad de total.

RF1.4 Mostrar los datos de un nomenclador similar según parámetros relacionales.

RF1.5 Contar los datos de un nomenclador similar a los parámetros de la búsqueda realizada.

RF1.6 Mostrar los datos de un nomenclador similar según parámetros y un número de inicio y una cantidad de total.

#### RF 2 Permitir la gestión de las personas.

RF2.1 Insertar persona nacional.

RF2.2 Insertar básico persona nacional.

RF2.3 Insertar persona extranjera.

RF2.4 Insertar básico persona extranjera.

RF2.5 Insertar persona no registrada.

RF2.6 Insertar básico persona no registrada.

RF2.7 Modificar persona nacional.

RF2.8 Modificar persona extranjera.

RF2.9 Modificar persona no registrada.

RF2.10 Modificar estado fallecido de la persona.

RF2.11 Modificar dirección permanente de la persona.

## **RF 3 Búsqueda de una persona en los registros operativos.**

RF3.1 Búsqueda fonética de una persona en los registros operativos según datos biográficos.

RF3.2 Buscar una persona en los registros operativos dado su número de identidad.

RF3.3 Buscar una persona en los registros operativos dado un tipo de información.

## **RF 4 Servicio dedicado a la búsqueda de personas.**

RF4.1 Buscar listado de personas dada una persona semejante.

RF4.2 Buscar una persona según el id de la persona.

RF4.3 Buscar una persona según el número de identidad de la persona.

RF4.4 Buscar personas nacionales similares a una persona dada.

RF4.5 Buscar personas que tengan una dirección dada.

RF4.6 Buscar persona nacional dado el identificador de persona.

RF4.7 Buscar persona nacional dado el número de identidad.

RF4.8 Buscar personas extranjeras, dada una persona similar.

RF4.9 Buscar persona extranjera dado un id de persona.

RF4.10 Buscar persona extranjera dados el documento de identidad y una persona semejante.

RF4.11 Buscar personas no registradas dada una persona similar.

RF4.12 Buscar persona no registrada dado el identificador de una persona.

RF4.13 Buscar persona no registrada dado el número de identidad.]

RF4.14 Obtener foto de una persona dado el identificador de la foto.

## **RF 5 Pedido de procesos para una persona dada.**

RF5.1 Comprobar si se puede realizar un proceso a una persona dada.

RF5.2 Adicionar un proceso dado a una persona.

## **RF 6 Búsqueda de direcciones.**

RF6.1 Buscar listado de direcciones dado el identificador de una persona.

RF6.2 Buscar listado de direcciones dado el ci de una persona.

RF6.3 Buscar listado de direcciones dada una tira de dirección.

RF6.4 Buscar listado de direcciones por parámetros.

## Anexo II. DESCRIPCIÓN DE REQUISITOS FUNCIONALES

### RF1. Permitir la gestión de los nomencladores.

RF1.1. Mostrar los nomencladores existentes dado una clasificación de nomenclador.

<b>Propósito</b>	Mostrar los nomencladores existentes en una lista dado una clasificación de nomenclador.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NOMENCLADORES	NAREASMININT NATRIBUTODOCUMENTO NATRIBUTOOBJETOCNI NCATEGORIAEXTRANJERO NCAUSACANCELARDENEGAR NCAUSACNI NCDR NCIRCUNSCRIPCION NCODIGOOACIPAI NCOLORCABELLO NCOLOROJO NCOLORPIEL NCONDICIONMIGRATORIA NCONSEJOPOPULAR

		<p>NCONTINENTE NDEPARTAMENTO NENTIDADTRAMITADORA NESTADOACTIVACION NESTADOCIRCULACION NESTADOFINALPROCESO NEXPRESIONROSTRO NEXTENSIONIMAGEN NFORMAENTRADAPAISPER NFORMALLEGADACNI NFORMASSALIDAPAISPER NGENERO NGRUPOLUGAR NINSTITUCION NLUGARGENERAL NLUGARTRASLADO NMUNICIPIO NNIVELCULTURAL NNOMBREANNO NORGANO NPAIS NPODERPOPULAR NPROVINCIA NPUNTOCARACTERISTICO NREGIONGEOGRAFICA NREGISTROCIVIL NREGLALUGAR NREQUISITOTIPOTRAMITE NREQUISITOTRAMITE NSECCION</p>
--	--	--

		NSITMIGENPOLITICA NSITMIGFUERAPOLITICA NTIPOAPLICACION NTIPOCIRCULACION NTIPODATO NTIPODECISIONCNI NTIPODIRECCION NTIPODOCCNI NTIPODOCIDENTIFICACION NTIPODOCUMENTO NTIPOIMAGEN NTIPOIMAGENFACIAL NTIPOINCAPACIDAD NTIPOINCIDENCIA NTIPOINMUEBLE NTIPOPOLUGAR NTIPOPOMEDIDACNI NTIPOPOMONEDA NTIPOOBJETOCNI NTIPOOFICINA NTIPOPERSMISOVIAJE NTIPOPERSONA NTIPOPDERPOPULAR NTIPOPROCESO NTIPOPROCESOBLOQUEO NTIPOPROCESOTIPOTRAMITE NTIPOPUNTOSIMPORTANCIA NTIPOSERVICIOAO NTIPOSOLICITUDCIRC NTIPOPOTRAMITE
--	--	--



		NTIPOVINCULOFAMILIAR NTIPOVISITACNI NVALIDACIONIMAGEN NVIASOLICITUDNEGREGRESO NZONA
<b>Descripción</b>	RF1.1.1. Mostrar listado de los posibles nomencladores dado el tipo de nomenclador. RF1.1.2. Mostrar detalles de cada nomenclador devuelto.	
<b>Validaciones</b>	1. Validar que el tipo de nomenclador a buscar sea correcto.	
<b>Postcondiciones</b>	1. Se devolvió el listado de nomencladores según la clasificación entrada.	
<b>Prototipo</b>	No procede.	

**Tabla 12 Mostrar los nomencladores existentes dado una clasificación de nomenclador.**

RF1.2. Contar los nomencladores existentes dado una clasificación de nomenclador.

<b>Propósito</b>	Contar los nomencladores existentes en una lista dado una clasificación de nomenclador.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NOMENCLADORES	NAREASMININT NATRIBUTODOCUMENTO NATRIBUTOOBJETOCNI NCATEGORIAEXTRANJERO NCAUSACANCELARDENEGAR NCAUSACNI NCDR NCIRCUNSCRIPCION NCODIGOOACIPAI NCOLORCABELLO

		NCOLOROJO NCOLORPIEL NCONDICIONMIGRATORIA NCONSEJOPOPULAR NCONTINENTE NDEPARTAMENTO NENTIDADTRAMITADORA NESTADOACTIVACION NESTADOCIRCULACION NESTADOFINALPROCESO NEXPRESIONROSTRO NEXTENSIONIMAGEN NFORMAENTRADAPAISPER NFORMALLEGADACNI NFORMASSALIDAPAISPER NGENERO NGRUPOLUGAR NINSTITUCION NLUGARGENERAL NLUGARTRASLADO NMUNICIPIO NNIVELCULTURAL NNOMBREANNO NORGANO NPAIS NPODERPOPULAR NPROVINCIA NPUNTOCARACTERISTICO NREGIONGEOGRAFICA NREGISTROCIVIL
--	--	---

		NREGLALUGAR NREQUISITOTIPOTRAMITE NREQUISITOTRAMITE NSECCION NSITMIGENPOLITICA NSITMIGFUERAPOLITICA NTIPOAPLICACION NTIPOCIRCULACION NTIPODATO NTIPODECISIONCNI NTIPODIRECCION NTIPODOCCNI NTIPODOCIDENTIFICACION NTIPODOCUMENTO NTIPOIMAGEN NTIPOIMAGENFACIAL NTIPOINCAPACIDAD NTIPOINCIDENCIA NTIPOINMUEBLE NTIPOPOLUGAR NTIPOMEDIDACNI NTIPOMONEDA NTIPOOBJETOCNI NTIPOOFICINA NTIPOPERSMISOVIAJE NTIPOPERSONA NTIPOPODERPOPULAR NTIPOPORCESO NTIPOPORCESOBLOQUEO NTIPOPORCESOTIPOTRAMITE
--	--	---

		NTIPOPUNTOSIMPORTANCIA NTIPOSERVICIOAO NTIPOSOLICITUDCIRC NTIPOPOTRAMITE NTIPOVINCULOOFAMILIAR NTIPOVISITACNI NVALIDACIONIMAGEN NVIASOLICITUDNEGREGRESO NZONA
<b>Descripción</b>	RF1.2.1. Contar listado de los posibles nomencladores dado el tipo de nomenclador.	
<b>Validaciones</b>	1. Validar que el tipo de nomenclador a buscar sea correcto.	
<b>Postcondiciones</b>	1. Se devolvió la cantidad de nomencladores según la clasificación entrada.	
<b>Prototipo</b>	No procede.	

**Tabla 13 Contar los nomencladores existentes dado una clasificación de nomenclador.**

RF1.3. Mostrar los nomencladores existentes dado una clasificación de nomenclador, un número de inicio y una cantidad de total.

<b>Propósito</b>	Mostrar los nomencladores existentes dado una clasificación de nomenclador, un número de inicio y una cantidad de total.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NOMENCLADORES	NAREASMININT NATRIBUTODOCUMENTO NATRIBUTOOBJETOCNI NCATEGORIAEXTRANJERO NCAUSACANCELARDENEGAR NCAUSACNI

		<p>NCDR NCIRCUNSCRIPCION NCODIGOOACIPAI NCOLORCABELLO NCOLOROJO NCOLORPIEL NCONDICIONMIGRATORIA NCONSEJOPOPULAR NCONTINENTE NDEPARTAMENTO NENTIDADTRAMITADORA NESTADOACTIVACION NESTADOCIRCULACION NESTADOFINALPROCESO NEXPRESIONROSTRO NEXTENSIONIMAGEN NFORMAENTRADAPAISPER NFORMALLEGADACNI NFORMASSALIDAPAISPER NGENERO NGRUPOLUGAR NINSTITUCION NLUGARGENERAL NLUGARTRASLADO NMUNICIPIO NNIVELCULTURAL NNOMBREANNO NORGANO NPAIS NPODERPOPULAR</p>
--	--	---

		<p>NPROVINCIA NPUNTOCARACTERISTICO NREGIONGEOGRAFICA NREGISTROCIVIL NREGLALUGAR NREQUISITOTIPOTRAMITE NREQUISITOTRAMITE NSECCION NSITMIGENPOLITICA NSITMIGFUERAPOLITICA NTIPOAPLICACION NTIPOCIRCULACION NTIPODATO NTIPODECISIONCNI NTIPODIRECCION NTIPODOCCNI NTIPODOCIDENTIFICACION NTIPODOCUMENTO NTIPOIMAGEN NTIPOIMAGENFACIAL NTIPOINCAPACIDAD NTIPOINCIDENCIA NTIPOINMUEBLE NTIPOPOLUGAR NTIPOPOMEDIDACNI NTIPOPOMONEDA NTIPOOBJETOCNI NTIPOOFICINA NTIPOPERSMISOVIAJE NTIPOPERSONA</p>
--	--	---

		NTIPOPODERPOPULAR NTIPOPROCESO NTIPOPROCESOBLOQUEO NTIPOPROCESOTIPOTRAMITE NTIPOPUNTOSIMPORTANCIA NTIPOSERVICIOAO NTIPOSOLICITUDCIRC NTIPOPOTRAMITE NTIPOVINCULOFAMILIAR NTIPOVISITACNI NVALIDACIONIMAGEN NVIASOLICITUDNEGREGRESO NZONA
<b>Descripción</b>	RF1.3.1. Mostrar listado de los posibles nomencladores. RF1.3.1.1. Mostrar detalles de cada nomenclador devuelto.	
<b>Validaciones</b>	1. Validar que el tipo de nomenclador a buscar sea correcto.	
<b>Postcondiciones</b>	1. Se devolvió el listado de nomencladores según la clasificación, el número de inicio y cantidad total.	
<b>Prototipo</b>	No procede.	

**Tabla 14 Mostrar los nomencladores existentes dado una clasificación de nomenclador, un número de inicio y una cantidad de total.**

RF1.4. Mostrar los datos de un nomenclador similar según parámetros relacionales.

<b>Propósito</b>	Mostrar los datos de un nomenclador similar según parámetros.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NOMENCLADORES	NAREASMININT NATRIBUTODOCUMENTO

		NATRIBUTOOBJETOCNI NCATEGORIAEXTRANJERO NCAUSACANCELARDENEGAR NCAUSACNI NCDR NCIRCUNSCRIPCION NCODIGOOACIPAI NCOLORCABELLO NCOLOROJO NCOLORPIEL NCONDICIONMIGRATORIA NCONSEJOPOPULAR NCONTINENTE NDEPARTAMENTO NENTIDADTRAMITADORA NESTADOACTIVACION NESTADOCIRCULACION NESTADOFINALPROCESO NEXPRESIONROSTRO NEXTENSIONIMAGEN NFORMAENTRADAPAISPER NFORMALLEGADACNI NFORMASSALIDAPAISPER NGENERO NGRUPOLUGAR NINSTITUCION NLUGARGENERAL NLUGARTRASLADO NMUNICIPIO NNIVELCULTURAL
--	--	---



		NNOMBREANNO NORGANO NPAIS NPODERPOPULAR NPROVINCIA NPUNTOCARACTERISTICO NREGIONGEOGRAFICA NREGISTROCIVIL NREGLALUGAR NREQUISITOTIPOTRAMITE NREQUISITOTRAMITE NSECCION NSITMIGENPOLITICA NSITMIGFUERAPOLITICA NTIPOAPLICACION NTIPOCIRCULACION NTIPODATO NTIPODECISIONCNI NTIPODIRECCION NTIPODOCCNI NTIPODOCIDENTIFICACION NTIPODOCUMENTO NTIPOIMAGEN NTIPOIMAGENFACIAL NTIPOINCAPACIDAD NTIPOINCIDENCIA NTIPOINMUEBLE NTIPOLUGAR NTIPOMEDIDACNI NTIPOMONEDA
--	--	---

		NTIPOOBJETOCNI NTIPOOFICINA NTIPOPERMISOVIAJE NTIPOPERSONA NTIPOPODERPOPULAR NTIPOPROCESO NTIPOPROCESOBLOQUEO NTIPOPROCESOTIPOTRAMITE NTIPOPUNTOSIMPORTANCIA NTIPOSERVICIOAO NTIPOSOLICITUDCIRC NTIPOPOTRAMITE NTIPOVINCULOOFAMILIAR NTIPOVISITACNI NVALIDACIONIMAGEN NVIASOLICITUDNEGREGRESO NZONA
<b>Descripción</b>	RF1.4.1. Buscar los datos de un nomenclador según parámetros.	
<b>Validaciones</b>	1. Validar que el tipo de nomenclador a buscar sea correcto.	
<b>Postcondiciones</b>	1. Se visualizó los nomencladores similares a los parámetros.	
<b>Prototipo</b>	No procede.	

**Tabla 15 Mostrar los datos de un nomenclador similar según parámetros relacionales.**

RF1.5. Contar los datos de un nomenclador similar a los parámetros de la búsqueda realizada.

<b>Propósito</b>	Contar los datos de un nomenclador similar a los parámetros de la búsqueda realizada.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>

	<p>NOMENCLADORES</p>	<p>NAREASMININT  NATRIBUTODOCUMENTO  NATRIBUTOOBJETOCNI  NCATEGORIAEXTRANJERO  NCAUSACANCELARDENEGAR  NCAUSACNI  NCDR  NCIRCUNSCRIPCION  NCODIGOOACIPAI  NCOLORCABELLO  NCOLOROJO  NCOLORPIEL  NCONDICIONMIGRATORIA  NCONSEJOPOPULAR  NCONTINENTE  NDEPARTAMENTO  NENTIDADTRAMITADORA  NESTADOACTIVACION  NESTADOCIRCULACION  NESTADOFINALPROCESO  NEXPRESIONROSTRO  NEXTENSIONIMAGEN  NFORMAENTRADAPAISPER  NFORMALLEGADACNI  NFORMASSALIDAPAISPER  NGENERO  NGRUPOLUGAR  NINSTITUCION  NLUGARGENERAL  NLUGARTRASLADO</p>
--	----------------------	--

		NMUNICIPIO NNIVELCULTURAL NNOMBREANNO NORGANO NPAIS NPODERPOPULAR NPROVINCIA NPUNTOCARACTERISTICO NREGIONGEOGRAFICA NREGISTROCIVIL NREGLALUGAR NREQUISITOTIPOTRAMITE NREQUISITOTRAMITE NSECCION NSITMIGENPOLITICA NSITMIGFUERAPOLITICA NTIPOAPLICACION NTIPOCIRCULACION NTIPODATO NTIPODECISIONCNI NTIPODIRECCION NTIPODOCCNI NTIPODOCIDENTIFICACION NTIPODOCUMENTO NTIPOIMAGEN NTIPOIMAGENFACIAL NTIPOINCAPACIDAD NTIPOINCIDENCIA NTIPOINMUEBLE NTIPOPOLUGAR
--	--	--

		NTIPIOMEDIDACNI NTIPIOMONEDA NTIPIOOBJETOCNI NTIPIPOOFICINA NTIPIPERMISOVIAJE NTIPIPERSONA NTIPIPODERPOPULAR NTIPIPROCESO NTIPIPROCESOBLOQUEO NTIPIPROCESOTIPOTRAMITE NTIPIPOPUNTOSIMPORTANCIA NTIPIOSERVICIOAO NTIPIOSOLICITUDCIRC NTIPIOTRAMITE NTIPIOVINCULOFAMILIAR NTIPIOVISITACNI NVALIDACIONIMAGEN NVIASOLICITUDNEGREGRESO NZONA
<b>Descripción</b>	RF1.5.1. Obtener la cantidad de datos de un nomenclador que coinciden con los parámetros de la búsqueda realizada.	
<b>Validaciones</b>	1. Validar que el tipo de nomenclador a buscar sea correcto.	
<b>Postcondiciones</b>	1. Se obtuvo el total de nomencladores que coinciden con los parámetros entrados.	
<b>Prototipo</b>	No procede.	

**Tabla 16 Contar los datos de un nomenclador similar a los parámetros de la búsqueda realizada.**

RF1.6. Mostrar los datos de un nomencladorsimilar según parámetros, un número de inicio y una cantidad de total.

<b>Propósito</b>	Obtener los datos de un nomenclador similar según parámetros, un número de inicio y una cantidad de total.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NOMENCLADORES	NAREASMININT NATRIBUTODOCUMENTO NATRIBUTOOBJETOCNI NCATEGORIAEXTRANJERO NCAUSACANCELARDENEGAR NCAUSACNI NCDR NCIRCUNSCRIPCION NCODIGOOACIPAI NCOLORCABELLO NCOLOROJO NCOLORPIEL NCONDICIONMIGRATORIA NCONSEJOPOPULAR NCONTINENTE NDEPARTAMENTO NENTIDADTRAMITADORA NESTADOACTIVACION NESTADOCIRCULACION NESTADOFINALPROCESO NEXPRESIONROSTRO NEXTENSIONIMAGEN NFORMAENTRADAPAISPER NFORMALLEGADACNI NFORMASSALIDAPAISPER

		NGENERO NGRUPOLUGAR NINSTITUCION NLUGARGENERAL NLUGARTRASLADO NMUNICIPIO NNIVELCULTURAL NNOMBREANNO NORGANO NPAIS NPODERPOPULAR NPROVINCIA NPUNTOCARACTERISTICO NREGIONGEOGRAFICA NREGISTROCIVIL NREGLALUGAR NREQUISITOTIPOTRAMITE NREQUISITOTRAMITE NSECCION NSITMIGENPOLITICA NSITMIGFUERAPOLITICA NTIPOAPLICACION NTIPOCIRCULACION NTIPODATO NTIPODECISIONCNI NTIPODIRECCION NTIPODOCNI NTIPODOCIDENTIFICACION NTIPODOCUMENTO NTIPOIMAGEN
--	--	---

		NTIPOIMAGENFACIAL NTIPOINCAPACIDAD NTIPOINCIDENCIA NTIPOINMUEBLE NTIPOPOLUGAR NTIPOPOMEDIDACNI NTIPOPOMONEDA NTIPOOBJETOCNI NTIPOOFICINA NTIPOPERSMISOVIAJE NTIPOPERSONA NTIPOPODERPOPULAR NTIPOPROCESO NTIPOPROCESOBLOQUEO NTIPOPROCESOTIPOTRAMITE NTIPOPUNTOSIMPORTANCIA NTIPOSERVICIOAO NTIPOSOLICITUDCIRC NTIPOPOTRAMITE NTIPOVINCULOFAMILIAR NTIPOVISITACNI NVALIDACIONIMAGEN NVIASOLICITUDNEGREGRESO NZONA
<b>Descripción</b>	RF1.6.1. Obtener los datos de un nomenclador similar según parámetros y un número de inicio y una cantidad de total.	
<b>Validaciones</b>	1. Validar que el tipo de nomenclador a buscar sea correcto.	
<b>Postcondiciones</b>	1. Se visualizó los datos del nomenclador similar según los parámetros, el número de inicio y la cantidad total introducida.	
<b>Prototipo</b>	No procede.	



**Tabla 17 Obtener los datos de un nomenclador similar según parámetros, un número de inicio y una cantidad de total.**

## RF2. Permitir la gestión de las personas.

RF2.1. Insertar persona nacional.

<b>Propósito</b>	Insertar persona nacional dado un conjunto de parámetros.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NACIONAL	IDPERSONA NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO FECHANACIMIENTO SEXO ANNOREGISTRO FALLECIDO IDTIPOPERSONA TOMO FOLIO IDMUNICIPIONACIMIENTO IDPROVINCIANACIMIENTO IDREGISTROCIVIL NOMBREMADRE NOMBREPADRE PESO TALLA

		DONANTE IDCOLORPIEL IDCOLOROJOS IDCOLORCABELLO IDNIVELCULTURAL CORREO SENNASPARTICULARES TELEFONOPARTICULAR TELEFONOLABORAL TELEFONOCELULAR OFICIO PROFESION CENTROLABORAL
<b>Descripción</b>	RF2.1.1. Insertar persona nacional.	
<b>Validaciones</b>	1. Validar que los datos introducidos sean válidos.	
<b>Postcondiciones</b>	1. Se insertó la persona nacional.	
<b>Prototipo</b>	No procede.	

**Tabla 18 Insertar Persona Nacional.**

RF2.2. Insertar básico persona nacional.

<b>Propósito</b>	Insertar básico persona nacional dado un conjunto de parámetros.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NACIONAL	IDPERSONA NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO

		SEGUNDOAPELLIDO FECHANACIMIENTO SEXO ANNOREGISTRO FALLECIDO IDTIPOPERSONA TOMO FOLIO IDMUNICIPIONACIMIENTO IDPROVINCIANACIMIENTO IDREGISTROCIVIL
<b>Descripción</b>	RF2.2.1. Insertar básico persona nacional.	
<b>Validaciones</b>	1. Validar que los datos introducidos sean válidos.	
<b>Postcondiciones</b>	1. Se insertó la persona nacional básica.	
<b>Prototipo</b>	No procede.	

**Tabla 19 Insertar Básico Persona Nacional.**

RF2.3. Insertar persona extranjera.

<b>Propósito</b>	Insertar persona extranjera dado un conjunto de parámetros.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	EXTRANJERO	IDPERSONA LUGARNACIMIENTO PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO IDENTPRIMARIO

		<p>IDENTSECUNDARIO  FECHANACIMIENTO  SEXO  FALLECIDO  IDTIPOPERSONA  IDCATEGORIAEXTRANJERO  CODIGOOCRBNACIONALIDAD  NOMBREMADRE  NOMBREPADRE  PESO  TALLA  DONANTE  IDCOLORPIEL  IDCOLOROJOS  IDCOLORCABELLO  IDNIVELCULTURAL  CORREO  SENNASPARTICULARES  TELEFONOPARTICULAR  TELEFONOLABORAL  TELEFONOCELULAR  OFICIO  PROFESION  CENTROLABORAL</p>
<b>Descripción</b>	RF2.3.1. Insertar persona extranjera.	
<b>Validaciones</b>	1. Validar que los datos introducidos sean válidos.	
<b>Postcondiciones</b>	1. Se insertó la persona extranjera.	
<b>Prototipo</b>	No procede.	

**Tabla 20 Insertar Persona Extranjera.**

RF2.4. Insertar básico persona extranjera.

<b>Propósito</b>	Insertar persona extranjera dado un conjunto de parámetros.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	EXTRANJERO	IDPERSONA LUGARNACIMIENTO PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO IDENTPRIMARIO IDENTSECUNDARIO FECHANACIMIENTO SEXO FALLECIDO IDTIPOPERSONA IDCATEGORIAEXTRANJERO CODIGOOCRBNACIONALIDAD
<b>Descripción</b>	RF2.4.1. Insertar básico persona extranjera.	
<b>Validaciones</b>	1. Validar que los datos introducidos sean válidos.	
<b>Postcondiciones</b>	1. Se insertó la persona extranjera básica.	
<b>Prototipo</b>	No procede.	

**Tabla 21 Insertar Básico Persona Extranjera.**

RF2.5. Insertar persona no registrada.

<b>Propósito</b>	Insertar persona no registrada dado un conjunto de parámetros.
<b>Roles</b>	Funcionario correspondiente.

<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NOREGISTRADO	IDPERSONA NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO IDENTPRIMARIO IDENTSECUNDARIO FECHANACIMIENTO SEXO ANNOREGISTRO LUGARNACIMIENTO NOMBREMADRE NOMBREPADRE PESO TALLA DONANTE IDCOLORPIEL IDCOLOROJOS IDCOLORCABELLO IDNIVELCULTURAL CORREO SENNASPARTICULARES TELEFONOPARTICULAR TELEFONOLABORAL TELEFONOCELULAR OFICIO PROFESION

	CENTROLABORAL
<b>Descripción</b>	RF2.5.1. Insertar persona no registrada.
<b>Validaciones</b>	1. Validar que los datos introducidos sean válidos.
<b>Postcondiciones</b>	1. Se insertó la persona no registrada.
<b>Prototipo</b>	No procede.

**Tabla 22 Insertar Persona no Registrada.**

RF2.6. Insertar básico persona no registrada.

<b>Propósito</b>	Insertar básico persona no registrada dado un conjunto de parámetros.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NOREGISTRADO	IDPERSONA NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERPELLIDO SEGUNDOPELLIDO IDENTPRIMARIO IDENTSECUNDARIO FECHANACIMIENTO SEXO ANNOREGISTRO LUGARNACIMIENTO
<b>Descripción</b>	RF2.6.1. Insertar básico persona no registrada.	
<b>Validaciones</b>	1. Validar que los datos introducidos sean válidos.	
<b>Postcondiciones</b>	1. Se insertó la persona no registrada básica.	
<b>Prototipo</b>	No procede.	

**Tabla 23 Insertar Básico Persona no Registrada.**

RF2.7. Modificar persona nacional.

<b>Propósito</b>	Modificar persona nacional dado un conjunto de parámetros.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NACIONAL	IDPERSONA NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO FECHANACIMIENTO SEXO ANNOREGISTRO FALLECIDO IDTIPOPERSONA TOMO FOLIO IDMUNICIPIONACIMIENTO IDPROVINCIANACIMIENTO IDREGISTROCIVIL NOMBREMADRE NOMBREPADRE PESO TALLA DONANTE IDCOLORPIEL



		IDCOLOROJOS IDCOLORCABELLO IDNIVELCULTURAL CORREO SENNASPARTICULARES TELEFONOPARTICULAR TELEFONOLABORAL TELEFONOCELULAR OFICIO PROFESION CENTROLABORAL
<b>Descripción</b>	RF2.7.1. Modificar persona nacional.	
<b>Validaciones</b>	1. Validar que los datos introducidos sean válidos.	
<b>Postcondiciones</b>	1. Se modificó la persona nacional.	
<b>Prototipo</b>	No procede.	

**Tabla 24 Modificar Persona Nacional.**

RF2.8. Modificar persona extranjera.

<b>Propósito</b>	Modificar persona extranjera dado un conjunto de parámetros.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	EXTRANJERO	IDPERSONA LUGARNACIMIENTO PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO IDENTPRIMARIO

		<p>IDENTSECUNDARIO  FECHANACIMIENTO  SEXO  FALLECIDO  IDTIPOPERSONA  IDCATEGORIAEXTRANJERO  CODIGOOCRBNACIONALIDAD  NOMBREMADRE  NOMBREPADRE  PESO  TALLA  DONANTE  IDCOLORPIEL  IDCOLOROJOS  IDCOLORCABELLO  IDNIVELCULTURAL  CORREO  SENNASPARTICULARES  TELEFONOPARTICULAR  TELEFONOLABORAL  TELEFONOCELULAR  OFICIO  PROFESION  CENTROLABORAL</p>
<b>Descripción</b>	RF2.8.1. Mostrar persona extranjera.	
<b>Validaciones</b>	1. Validar que los datos introducidos sean válidos.	
<b>Postcondiciones</b>	1. Se modificó la persona extranjera.	
<b>Prototipo</b>	No procede.	

**Tabla 25 Modificar Persona Extranjera.**

RF2.9. Modificar persona no registrada.

<b>Propósito</b>	Modificar persona no registrada dado un conjunto de parámetros.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NOREGISTRADO	IDPERSONA NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO IDENTPRIMARIO IDENTSECUNDARIO FECHANACIMIENTO SEXO ANNOREGISTRO LUGARNACIMIENTO NOMBREMADRE NOMBREPADRE PESO TALLA DONANTE IDCOLORPIEL IDCOLOROJOS IDCOLORCABELLO IDNIVELCULTURAL CORREO SENNASPARTICULARES TELEFONOPARTICULAR

		TELEFONOLABORAL TELEFONOCELULAR OFICIO PROFESION CENTROLABORAL
<b>Descripción</b>	RF2.9.1. Mostrar persona no registrada.	
<b>Validaciones</b>	1. Validar que los datos introducidos sean válidos.	
<b>Postcondiciones</b>	1. Se modificó la persona no registrada.	
<b>Prototipo</b>	No procede	

**Tabla 26 Modificar Persona no Registrada.**

RF2.10. Modificar estado fallecido de la persona.

<b>Propósito</b>	Modificar estado fallecido de la persona.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	PERSONA	IDPERSONA FALLECIDO IDTIPOPERSONA IDCONDICIONMIGRATORIA
<b>Descripción</b>	RF2.10.1. Modificar estado fallecido de la persona.	
<b>Validaciones</b>	1. Validar que los datos introducidos sean válidos.	
<b>Postcondiciones</b>	1. Se modificó el estado fallecido de la persona.	
<b>Prototipo</b>	No procede.	

**Tabla 27 Modificar Estado Fallecido de la Persona.**

RF2.11. Modificar dirección permanente de la persona.

<b>Propósito</b>	Modificar dirección permanente de la persona.
------------------	---

<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	PERSONADIRECCION	IDPERSONADIRECCION IDPERSONA IDDIRECCION IDTIPODIRECCION FECHAALTA FECHABAJA ACTIVO
<b>Descripción</b>	RF2.11.1. Modificar dirección permanente de la persona.	
<b>Validaciones</b>	1. Validar que los datos introducidos sean válidos.	
<b>Postcondiciones</b>	1. Se modificó la dirección permanente de la persona.	
<b>Prototipo</b>	No procede.	

**Tabla 28 Modificar Dirección Permanente de la Persona.**

### RF3. Búsqueda de una persona en los registros operativos.

RF3.1. Búsqueda fonética de una persona en los registros operativos según datos biográficos.

<b>Propósito</b>	Buscar personas en los registros operativos por datos biográficos.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
	<b>Concepto</b>	<b>Atributos</b>
	PERSONAOPERATIVO	IDPERSONA NUMEROIDENTIDAD EDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERPELLIDO

		SEGUNDOAPELLIDO FECHANACIMIENTO SEXO NOMBREMADRE NOMBREPADRE PESO TALLA DONANTE FALLECIDO CONDICIONMIGRATORIA PAIS COLORPIEL COLOROJOS COLORCABELLO DIRECCION TIPOPERSONA REGISTROCIVIL DATOSOPCIONALES DMA PERSONAIMAGEN
<b>Descripción</b>	RF3.1.1. Mostrar los campos para la búsqueda: <ul style="list-style-type: none"> <li>a) Nombre de la persona.</li> <li>b) Segundo nombre.</li> <li>c) Nombre de la familia.</li> <li>d) Apellidos.</li> <li>e) Tipos de informaciones que se necesitan de las personas encontradas.</li> </ul> RF3.1.2. Buscar persona en los registros. RF3.1.3. Mostrar listado de personas que coincidan con el criterio de búsqueda especificado.	
<b>Validaciones</b>		

<b>Postcondiciones</b>	1. Se devuelve el listado de personas que cumplan con los criterios de búsqueda especificados.
<b>Prototipo</b>	No procede.

Tabla 29 Búsqueda fonética de una persona en los registros operativos según datos biográficos.

RF3.2. Buscar una persona en los registros operativos dado su número de identidad.

<b>Propósito</b>	Buscar una persona en los registros operativos según su número de identidad.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	PERSONAOPERATIVO	IDPERSONA NUMEROIDENTIDAD EDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO FECHANACIMIENTO SEXO NOMBREMADRE NOMBREPADRE PESO TALLA DONANTE FALLECIDO CONDICIONMIGRATORIA PAIS COLORPIEL COLOROJOS

		COLORCABELLO DIRECCION TIPOPERSONA REGISTROCIVIL DATOSOPCIONALES DMA PERSONAIMAGEN
<b>Descripción</b>	RF3.2.1. Buscar personas por los campos. a) Número de identidad. b) Tipos de informaciones que se necesitan de las personas encontradas RF3.2.2. Realizar búsqueda. RF3.2.3. Mostrar listado de personas si la búsqueda es positiva.	
<b>Validaciones</b>	1. Validar que el número de identidad de la persona exista.	
<b>Postcondiciones</b>	1. Se muestra el listado de las personas que cumplen con el criterio de búsqueda.	
<b>Prototipo</b>	No procede.	

**Tabla 30 Buscar una persona en los registros operativos dado su número de identidad.**

RF3.3. Buscar una persona en los registros operativos dado un tipo de información.

<b>Propósito</b>	Buscar una persona en los registros operativos dependiendo del tipo de información que se desee obtener de ella.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	PERSONAOPERATIVO	IDPERSONA NUMEROIDENTIDAD EDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO



		<p>SEGUNDOAPELLIDO</p> <p>FECHANACIMIENTO</p> <p>SEXO</p> <p>NOMBREMADRE</p> <p>NOMBREPADRE</p> <p>PESO</p> <p>TALLA</p> <p>DONANTE</p> <p>FALLECIDO</p> <p>CONDICIONMIGRATORIA</p> <p>PAIS</p> <p>COLORPIEL</p> <p>COLOROJOS</p> <p>COLORCABELLO</p> <p>DIRECCION</p> <p>TIPOPERSONA</p> <p>REGISTROCIVIL</p> <p>DATOSOPCIONALES</p> <p>DMA</p> <p>PERSONAIMAGEN</p>
<b>Descripción</b>	<p>RF3.3.1. Buscar personas por especificando.</p> <ul style="list-style-type: none"> <li>a) Número de identidad.</li> <li>b) Tipos de informaciones que se necesitan de las personas encontradas                             <ul style="list-style-type: none"> <li>a) Circulados.</li> <li>b) Prisiones.</li> <li>c) Registro policial.</li> </ul> </li> </ul> <p>RF3.3.2. Realizar búsqueda.</p> <p>RF3.3.3. Mostrar listado de personas si la búsqueda es positiva.</p>	
<b>Validaciones</b>	<p>1. Validar que el número de identidad de la persona exista.</p>	
<b>Postcondiciones</b>	<p>1. Se muestra el listado de las personas que cumplen con el criterio de búsqueda.</p>	

<b>Prototipo</b>	No procede.
------------------	-------------

**Tabla 31 Buscar una persona en los registros operativos dado un tipo de información.**

## RF4. Servicio dedicado a la búsqueda de personas.

RF4.1. Buscar listado de personas dada una persona semejante.

<b>Propósito</b>	Mostrar listado de PersonSummary para las personas con datos similares a la persona pasada.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
	<b>Concepto</b>	<b>Atributos</b>
	PERSONARESUMEN	IDPERSONA FALLECIDO IDCONDICIONMIGRATORIA NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERPELLIDO SEGUNDOPELLIDO FECHANACIMIENTO NACIONALIDAD IDTIPOPERSONA DIRECCION
	PERSONA	IDPERSONA FALLECIDO IDTIPOPERSONA IDCONDICIONMIGRATORIA
<b>Descripción</b>	RF4.1.1. Buscar personas dado el criterio de búsqueda especificado. a) Mostrar listado de PersonSummary para las personas con datos similares a la	

	persona pasada.
<b>Validaciones</b>	1. Validar que los valores especificados sean válidos.
<b>Postcondiciones</b>	1. Se devuelve el listado de personas similares.
<b>Prototipo</b>	No procede.

**Tabla 32 Buscar listado de personas dada una persona semejante.**

RF4.2. Buscar una persona según el id de la persona.

<b>Propósito</b>	Buscar una persona según el id de la persona.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	PERSONA	IDPERSONA FALLECIDO IDTIPOPERSONA IDCONDICIONMIGRATORIA
<b>Descripción</b>	RF4.2.1. Buscar personas dado el criterio de búsqueda especificado. a) Mostrar datos de la persona encontrada.	
<b>Validaciones</b>	1. Validar que el id de la persona exista.	
<b>Postcondiciones</b>	1. Se devolvió la persona buscada.	
<b>Prototipo</b>	No procede.	

**Tabla 33 Buscar una persona según el id de la persona.**

RF4.3. Buscar una persona según el número de identidad de la persona.

<b>Propósito</b>	Buscar una persona según su número de identidad.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	PERSONA	IDPERSONA

		NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO FECHANACIMIENTO SEXO TOMO FOLIO ANNOREGISTRO IDTIPOPERSONA IDREGISTROCIVIL IDMUNICIPIONACIMIENTO IDPROVINCIANACIMIENTO FALLECIDO IDCONDICIONMIGRATORIA
<b>Descripción</b>	RF4.3.1. Buscar personas dado el número de identidad. a) Mostrar listado de personas encontradas.	
<b>Validaciones</b>	1. Validar que el número de identidad de la persona exista.	
<b>Postcondiciones</b>	1. Se muestra el listado de las personas que cumplen con el criterio de búsqueda.	
<b>Prototipo</b>	No procede.	

**Tabla 34 Buscar una persona según el número de identidad de la persona.**

RF4.4. Buscar personas nacionales similares a una persona dada.

<b>Propósito</b>	Buscar personas nacionales similares a una persona dada.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NACIONAL	IDPERSONA

		NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERPELLIDO SEGUNDOPELLIDO FECHANACIMIENTO SEXO ANNOREGISTRO FALLECIDO IDTIPOPERSONA TOMO FOLIO IDMUNICIPIONACIMIENTO IDPROVINCIANACIMIENTO IDREGISTROCIVIL NOMBREMADRE NOMBREPADRE PESO TALLA DONANTE IDCOLORPIEL IDCOLOROJOS IDCOLORCABELLO IDNIVELCULTURAL CORREO SENNASPARTICULARES TELEFONOPARTICULAR TELEFONOLABORAL TELEFONOCELULAR OFICIO
--	--	---

		PROFESION CENTROLABORAL
	PERSONARESUMEN	IDPERSONA FALLECIDO IDCONDICIONMIGRATORIA NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO FECHANACIMIENTO NACIONALIDAD IDTIPOPERSONA DIRECCION
<b>Descripción</b>	RF4.4.1. Buscar personas nacionales dado el criterio de búsqueda. a) Devolver listado de personas encontradas.	
<b>Validaciones</b>	1. Validar que la persona semejante no sea nula.	
<b>Postcondiciones</b>	1. Se muestra el listado de las personas que cumplen con el criterio de búsqueda.	
<b>Prototipo</b>	No procede.	

**Tabla 35 Buscar personas nacionales similares a una persona dada.**

RF4.5. Buscar personas nacionales que tengan una dirección dada.

<b>Propósito</b>	Buscar personas nacionales, dada una persona con datos similares y una dirección.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NACIONAL	IDPERSONA NUMEROIDENTIDAD

		PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO FECHANACIMIENTO SEXO ANNOREGISTRO FALLECIDO IDTIPOPERSONA TOMO FOLIO IDMUNICIPIONACIMIENTO IDPROVINCIANACIMIENTO IDREGISTROCIVIL NOMBREMADRE NOMBREPADRE PESO TALLA DONANTE IDCOLORPIEL IDCOLOROJOS IDCOLORCABELLO IDNIVELCULTURAL CORREO SENNASPARTICULARES TELEFONOPARTICULAR TELEFONOLABORAL TELEFONOCELULAR OFICIO PROFESION
--	--	---

		CENTROLABORAL
	PERSONARESUMEN	IDPERSONA FALLECIDO IDCONDICIONMIGRATORIA NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO FECHANACIMIENTO NACIONALIDAD IDTIPOPERSONA DIRECCION
	DIRECCIÓN	DESCRIPCIÓN
<b>Descripción</b>	RF4.5.1. Buscar personas nacionales dada la una persona semejante y una dirección. a) Devolver listado de personas encontradas.	
<b>Validaciones</b>	1. Validar que la persona semejante y la dirección no sean nulas.	
<b>Postcondiciones</b>	1. Se muestra el listado de las personas que cumplen con el criterio de búsqueda.	
<b>Prototipo</b>	No procede.	

**Tabla 36 Buscar personas nacionales que tengan una dirección dada.**

RF4.6. Buscar persona nacional dado el identificador de persona.

<b>Propósito</b>	Buscar personas nacionales dado un identificador de persona.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NACIONAL	IDPERSONA NUMEROIDENTIDAD



		PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO FECHANACIMIENTO SEXO ANNOREGISTRO FALLECIDO IDTIPOPERSONA TOMO FOLIO IDMUNICIPIONACIMIENTO IDPROVINCIANACIMIENTO IDREGISTROCIVIL NOMBREMADRE NOMBREPADRE PESO TALLA DONANTE IDCOLORPIEL IDCOLOROJOS IDCOLORCABELLO IDNIVELCULTURAL CORREO SENNASPARTICULARES TELEFONOPARTICULAR TELEFONOLABORAL TELEFONOCELULAR OFICIO PROFESION
--	--	---

	CENTROLABORAL
<b>Descripción</b>	RF4.6.1. Buscar personas nacionales dado el identificador de la persona. a) Devolver la persona encontrada.
<b>Validaciones</b>	1. Validar que el identificador de la persona exista.
<b>Postcondiciones</b>	1. Se muestra la persona que cumple con el criterio de búsqueda.
<b>Prototipo</b>	No procede.

**Tabla 37 Buscar persona nacional dado el identificador de persona.**

RF4.7. Buscar persona nacional dado el número de identidad.

<b>Propósito</b>	Buscar personas nacionales dado el número de identidad.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NACIONAL	IDPERSONA NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO FECHANACIMIENTO SEXO ANNOREGISTRO FALLECIDO IDTIPOPERSONA TOMO FOLIO IDMUNICIPIONACIMIENTO IDPROVINCIANACIMIENTO IDREGISTROCIVIL

		NOMBREMADRE NOMBREPADRE PESO TALLA DONANTE IDCOLORPIEL IDCOLOROJOS IDCOLORCABELLO IDNIVELCULTURAL CORREO SENNASPARTICULARES TELEFONOPARTICULAR TELEFONOLABORAL TELEFONOCELULAR OFICIO PROFESION CENTROLABORAL
<b>Descripción</b>	RF4.7.1. Buscar personas nacionales dado el número de identidad. a) Devolver persona encontrada.	
<b>Validaciones</b>	1. Validar que el número de identidad especificado exista.	
<b>Postcondiciones</b>	1. Se muestra la persona que cumple con el criterio de búsqueda.	
<b>Prototipo</b>	No procede.	

**Tabla 38 Buscar persona nacional dado el número de identidad.**

RF4.8. Buscar personas extranjeras, dada una persona similar.

<b>Propósito</b>	Buscar personas extranjeras, dada una persona con datos similares.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos</b>	<b>Concepto</b>	<b>Atributos</b>

<p><b>tratados</b></p>	<p>EXTRANJERO</p>	<p>IDPERSONA  LUGARNACIMIENTO  PRIMERNOMBRE  SEGUNDONOMBRE  PRIMERAPELLIDO  SEGUNDOAPELLIDO  IDENTPRIMARIO  IDENTSECUNDARIO  FECHANACIMIENTO  SEXO  FALLECIDO  IDTIPOPERSONA  IDCATEGORIAEXTRANJERO  CODIGOOCRBNACIONALIDAD  NOMBREMADRE  NOMBREPADRE  PESO  TALLA  DONANTE  IDCOLORPIEL  IDCOLOROJOS  IDCOLORCABELLO  IDNIVELCULTURAL  CORREO  SENNASPARTICULARES  TELEFONOPARTICULAR  TELEFONOLABORAL  TELEFONOCELULAR  OFICIO  PROFESION</p>
------------------------	-------------------	---

		CENTROLABORAL
	PERSONARESUMEN	IDPERSONA FALLECIDO IDCONDICIONMIGRATORIA NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO FECHANACIMIENTO NACIONALIDAD IDTIPOPERSONA DIRECCION
<b>Descripción</b>	RF4.8.1. Buscar personas extranjeras dado el criterio de búsqueda. a) Devolver listado de personas encontradas.	
<b>Validaciones</b>	1. Validar que la persona semejante no sea nula. 2. Validar que la persona sea extranjera.	
<b>Postcondiciones</b>	1. Se muestra el listado de las personas que cumplen con el criterio de búsqueda.	
<b>Prototipo</b>	No procede.	

**Tabla 39 Buscar personas extranjeras, dada una persona similar.**

RF4.9. Buscar persona extranjera dado un id de persona.

<b>Propósito</b>	Buscar personas extranjeras dado un id de persona	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	EXTRANJERO	IDPERSONA LUGARNACIMIENTO PRIMERNOMBRE

		SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO IDENTPRIMARIO IDENTSECUNDARIO FECHANACIMIENTO SEXO FALLECIDO IDTIPOPERSONA IDCATEGORIAEXTRANJERO CODIGOOCRBNACIONALIDAD NOMBREMADRE NOMBREPADRE PESO TALLA DONANTE IDCOLORPIEL IDCOLOROJOS IDCOLORCABELLO IDNIVELCULTURAL CORREO SENNASPARTICULARES TELEFONOPARTICULAR TELEFONOLABORAL TELEFONOCELULAR OFICIO PROFESION CENTROLABORAL
<b>Descripción</b>	RF4.9.1. Buscar personas extranjeras dado el id. a) Devolver la persona encontrada.	

<b>Validaciones</b>	<ol style="list-style-type: none"> <li>1. Validar que el id de la persona exista.</li> <li>2. Validar que la persona sea extranjera.</li> </ol>
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. Se muestra la persona que cumple con el criterio de búsqueda.</li> </ol>
<b>Prototipo</b>	No procede.

**Tabla 40 Buscar persona extranjera dado un id de persona.**

RF4.10. Buscar persona extranjera dados el documento de identidad y una persona semejante.

<b>Propósito</b>	Buscar personas extranjeras dados el número de identidad y una persona semejante.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El funcionario correspondiente debe estar autenticado en el sistema.</li> </ol>	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	EXTRANJERO	IDPERSONA LUGARNACIMIENTO PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO IDENTPRIMARIO IDENTSECUNDARIO FECHANACIMIENTO SEXO FALLECIDO IDTIPOPERSONA IDCATEGORIAEXTRANJERO CODIGOOOCRBNACIONALIDAD NOMBREMADRE NOMBREPADRE PESO

		<p>TALLA</p> <p>DONANTE</p> <p>IDCOLORPIEL</p> <p>IDCOLOROJOS</p> <p>IDCOLORCABELLO</p> <p>IDNIVELCULTURAL</p> <p>CORREO</p> <p>SENNASPARTICULARES</p> <p>TELEFONOPARTICULAR</p> <p>TELEFONOLABORAL</p> <p>TELEFONOCELULAR</p> <p>OFICIO</p> <p>PROFESION</p> <p>CENTROLABORAL</p>
	RESUMENPERSONA	<p>IDPERSONA</p> <p>FALLECIDO</p> <p>IDCONDICIONMIGRATORIA</p> <p>NUMEROIDENTIDAD</p> <p>PRIMERNOMBRE</p> <p>SEGUNDONOMBRE</p> <p>PRIMERAPELLIDO</p> <p>SEGUNDOAPELLIDO</p> <p>FECHANACIMIENTO</p> <p>NACIONALIDAD</p> <p>IDTIPOPERSONA</p> <p>DIRECCION</p>
	PERSONADOCIDENTIDAD	<p>IDPERSONADOCIDENTIDAD</p> <p>NUMEROIDENTIDAD</p> <p>FECHAREGISTRO</p> <p>IDTIPODOCIDENTIFICACION</p>



		IDPERSONA CODIGOOCRIDENTIDADEXP ACTIVO
<b>Descripción</b>	RF4.10.1. Buscar personas extranjeras dado el documento de identidad y una persona semejante. a) Devolver persona encontrada.	
<b>Validaciones</b>	1. Validar que la persona semejante y el documento de identidad especificado exista. 2. Validar que la persona sea extranjera.	
<b>Postcondiciones</b>	1. Se muestra el listado de personas que cumplen con el criterio de búsqueda.	
<b>Prototipo</b>	No procede.	

**Tabla 41 Buscar persona extranjera dados el documento de identidad y una persona semejante.**

RF4.11. Buscar personas no registradas dada una persona similar.

<b>Propósito</b>	Buscar personas no registradas dada una persona con datos similares.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NOREGISTRADO	IDPERSONA NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERAPELLIDO SEGUNDOAPELLIDO IDENTPRIMARIO IDENTSECUNDARIO FECHANACIMIENTO SEXO ANNOREGISTRO

		<p>LUGARNACIMIENTO</p> <p>NOMBREMADRE</p> <p>NOMBREPADRE</p> <p>PESO</p> <p>TALLA</p> <p>DONANTE</p> <p>IDCOLORPIEL</p> <p>IDCOLOROJOS</p> <p>IDCOLORCABELLO</p> <p>IDNIVELCULTURAL</p> <p>CORREO</p> <p>SENNASPARTICULARES</p> <p>TELEFONOPARTICULAR</p> <p>TELEFONOLABORAL</p> <p>TELEFONOCELULAR</p> <p>OFICIO</p> <p>PROFESION</p> <p>CENTROLABORAL</p>
	PERSONARESUMEN	<p>IDPERSONA</p> <p>FALLECIDO</p> <p>IDCONDICIONMIGRATORIA</p> <p>NUMEROIDENTIDAD</p> <p>PRIMERNOMBRE</p> <p>SEGUNDONOMBRE</p> <p>PRIMERAPELLIDO</p> <p>SEGUNDOAPELLIDO</p> <p>FECHANACIMIENTO</p> <p>NACIONALIDAD</p> <p>IDTIPOPERSONA</p> <p>DIRECCION</p>

<b>Descripción</b>	RF4.11.1. Buscar personas no registradas dado el criterio de búsqueda. a) Devolver listado de personas encontradas.
<b>Validaciones</b>	1. Validar que la persona semejante no sea nula. 2. Validar que la persona sea no registrada.
<b>Postcondiciones</b>	1. Se muestra el listado de las personas que cumplen con el criterio de búsqueda.
<b>Prototipo</b>	No procede.

**Tabla 42 Buscar personas no registradas dada una persona similar.**

RF4.12. Buscar persona no registrada dado el identificador de una persona.

<b>Propósito</b>	Buscar personas no registradas dado un identificador de persona	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NOREGISTRADO	IDPERSONA NUMEROIDENTIDAD PRIMERNOMBRE SEGUNDONOMBRE PRIMERPELLIDO SEGUNDOPELLIDO IDENTPRIMARIO IDENTSECUNDARIO FECHANACIMIENTO SEXO ANNOREGISTRO LUGARNACIMIENTO NOMBREMADRE NOMBREPADRE PESO TALLA

		DONANTE IDCOLORPIEL IDCOLOROJOS IDCOLORCABELLO IDNIVELCULTURAL CORREO SENNASPARTICULARES TELEFONOPARTICULAR TELEFONOLABORAL TELEFONOCELULAR OFICIO PROFESION CENTROLABORAL
<b>Descripción</b>	RF4.12.1. Buscar personas no registradas dado el identificador de una persona. a) Devolver la persona encontrada.	
<b>Validaciones</b>	1. Validar que el identificador de la persona exista. 2. Validar que la persona sea no registrada.	
<b>Postcondiciones</b>	1. Se muestra la persona que cumple con el criterio de búsqueda.	
<b>Prototipo</b>	No procede.	

**Tabla 43 Buscar persona no registrada dado el identificador de una persona.**

RF4.13. Buscar persona no registrada dado el número de identidad.

<b>Propósito</b>	Buscar personas no registradas dado el número de identidad.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NOREGISTRADO	IDPERSONA NUMEROIDENTIDAD PRIMERNOMBRE

		<p>SEGUNDONOMBRE                  PRIMERAPELLIDO                  SEGUNDOAPELLIDO                  IDENTPRIMARIO                  IDENTSECUNDARIO                  FECHANACIMIENTO                  SEXO                  ANNOREGISTRO                  LUGARNACIMIENTO                  NOMBREMADRE                  NOMBREPADRE                  PESO                  TALLA                  DONANTE                  IDCOLORPIEL                  IDCOLOROJOS                  IDCOLORCABELLO                  IDNIVELCULTURAL                  CORREO                  SENNASPARTICULARES                  TELEFONOPARTICULAR                  TELEFONOLABORAL                  TELEFONOCELULAR                  OFICIO                  PROFESION                  CENTROLABORAL</p>
<b>Descripción</b>	RF4.13.1. Buscar personas no registradas dado el número de identidad. a) Devolver persona encontrada.	
<b>Validaciones</b>	1. Validar que el número de identidad especificado exista. 2. Validar que la persona sea no registrada.	

<b>Postcondiciones</b>	1. Se muestra la persona que cumple con el criterio de búsqueda.
<b>Prototipo</b>	No procede.

**Tabla 44 Buscar persona no registrada dado el número de identidad.**

RF4.14. Obtener foto de una persona dado el identificador de la foto.

<b>Propósito</b>	Buscar foto de personas dado el identificador de la foto.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	PERSONAIMAGEN	IDPERSONAIMAGEN IMAGEN FECHAREGISTRO IDPERSONA IDTIPOIMAGEN IDVALIDACIONIMAGEN IDINFORMFACIAL IDINFORMIMAGEN
<b>Descripción</b>	RF4.14.1. Buscar foto de una persona por el identificador especificado. a) Devolver foto encontrada.	
<b>Validaciones</b>	1. Validar que el identificador especificado para la foto exista.	
<b>Postcondiciones</b>	1. Se devuelve la foto que cumple con el criterio de búsqueda.	
<b>Prototipo</b>	No procede.	

**Tabla 45 Obtener foto de una persona dado el identificador de la foto.**

**RF5. Pedido de procesos para una persona dada.**

RF5.1. Comprobar si se puede realizar un proceso a una persona dada.

<b>Propósito</b>	Comprobar si se puede realizar un proceso a una persona dada.
------------------	---

<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	PROCESO	IDTIPOPROCESO IDOFICINA FECHAFIN IDESTADOFINALPROCESO IDCAUSACANCELARDENEGAR INSTANCE_ID SERIAL SERIALPROCESOPADRE IDSERIALPROCESO
<b>Descripción</b>	RF5.1.1. Comprobar que se pueda realizar el proceso a la persona.	
<b>Validaciones</b>	1. Validar que el tipo de proceso a buscar sea correcto. 2. Validar que exista la persona.	
<b>Postcondiciones</b>	1. Devuelve si se puede realizar el proceso a la persona introducida.	
<b>Prototipo</b>	No procede.	

**Tabla 46 Comprobar si se puede realizar un proceso a una persona dada.**

RF5.2. Adicionar un proceso dado a una persona.

<b>Propósito</b>	Adicionar un proceso dado a una persona.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	PROCESO	IDTIPOPROCESO IDOFICINA FECHAFIN IDESTADOFINALPROCESO IDCAUSACANCELARDENEGAR

		INSTANCE_ID SERIAL SERIALPROCESOPADRE IDSERIALPROCESO
<b>Descripción</b>	RF5.2.1. Adicionar un proceso dado a una persona dado datos de esta.	
<b>Validaciones</b>	1. Validar que el tipo de proceso a buscar sea correcto. 2. Validar que exista el id de la persona especificado.	
<b>Postcondiciones</b>	1. Se devuelve el listado de los procesos, si existen, que impiden a la persona iniciar el proceso introducido.	
<b>Prototipo</b>	No procede.	

**Tabla 47 Adicionar un proceso dado a una persona.**

## RF6. Búsqueda de direcciones.

RF6.1. Buscar listado de direcciones dado el identificador de una persona.

<b>Propósito</b>	Mostrar listado de direcciones para una persona dado el identificador especificado.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	DIRECCION	IDDIRECCION TIRAANTERIOR TIRACODIFICADA MIGRACION COORDENADAS
	PERSONADIRECCION	IDPERSONADIRECCION IDPERSONA IDTIPODIRECCION IDDIRECCION FECHAALTA



		FECHABAJA ACTIVO
<b>Descripción</b>	RF6.1.1. Buscar direcciones dado el identificador de persona especificado. a) Mostrar listado de direcciones.	
<b>Validaciones</b>	1. Validar que exista el identificador de persona especificado.	
<b>Postcondiciones</b>	1. Se devuelve el listado de direcciones asociadas a la persona.	
<b>Prototipo</b>	No procede.	

**Tabla 48 Buscar listado de direcciones dado el identificador de una persona.**

RF6.2. Buscar listado de direcciones dado el ci de una persona.

<b>Propósito</b>	Mostrar listado de direcciones para una persona dado el ci especificado.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	DIRECCION	IDDIRECCION TIRAANTERIOR TIRACODIFICADA MIGRACION COORDENADAS
	PERSONADIRECCION	IDPERSONADIRECCION IDPERSONA IDTIPODIRECCION IDDIRECCION FECHAALTA FECHABAJA ACTIVO
<b>Descripción</b>	RF6.2.1. Buscar direcciones dado la tira de direcciones especificada. a) Mostrar listado de direcciones.	
<b>Validaciones</b>	1. Validar que exista el carné de identidad de la persona especificado.	

<b>Postcondiciones</b>	1. Se devuelve el listado de direcciones que cumplan con el criterio de búsqueda.
<b>Prototipo</b>	No procede.

**Tabla 49 Buscar listado de direcciones dado el ci de una persona.**

RF6.3. Buscar listado de direcciones dada una tira de dirección.

<b>Propósito</b>	Mostrar listado de direcciones para una persona dada una tira de dirección especificada.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	DIRECCION	IDDIRECCION TIRAANTERIOR TIRACODIFICADA MIGRACION COORDENADAS
<b>Descripción</b>	RF6.3.1. Buscar direcciones dado la tira especificada. a) Mostrar listado de direcciones.	
<b>Validaciones</b>	1. Validar que exista el identificador de la dirección buscada.	
<b>Postcondiciones</b>	1. Se devuelve el listado de direcciones asociadas a la tira.	
<b>Prototipo</b>	No procede.	

**Tabla 50 Buscar listado de direcciones dada una tira de dirección.**

RF6.4. Buscar listado de direcciones por parámetros.

<b>Propósito</b>	Mostrar listado de direcciones filtrando por parámetros.	
<b>Roles</b>	Funcionario correspondiente.	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	DIRECCION	IDDIRECCION

		TIRAANTERIOR TIRACODIFICADA MIGRACION COORDENADAS
<b>Descripción</b>	RF6.4.1. Buscar direcciones por los parámetros especificados. a) Mostrar listado de direcciones.	
<b>Validaciones</b>		
<b>Postcondiciones</b>	1. Se devuelve el listado de direcciones que cumplan con los parámetros especificados.	
<b>Prototipo</b>	No procede.	

**Tabla 51 Buscar listado de direcciones por parámetros.**

### Anexo III. FRAGMENTO DEL MODELO DE DATOS.

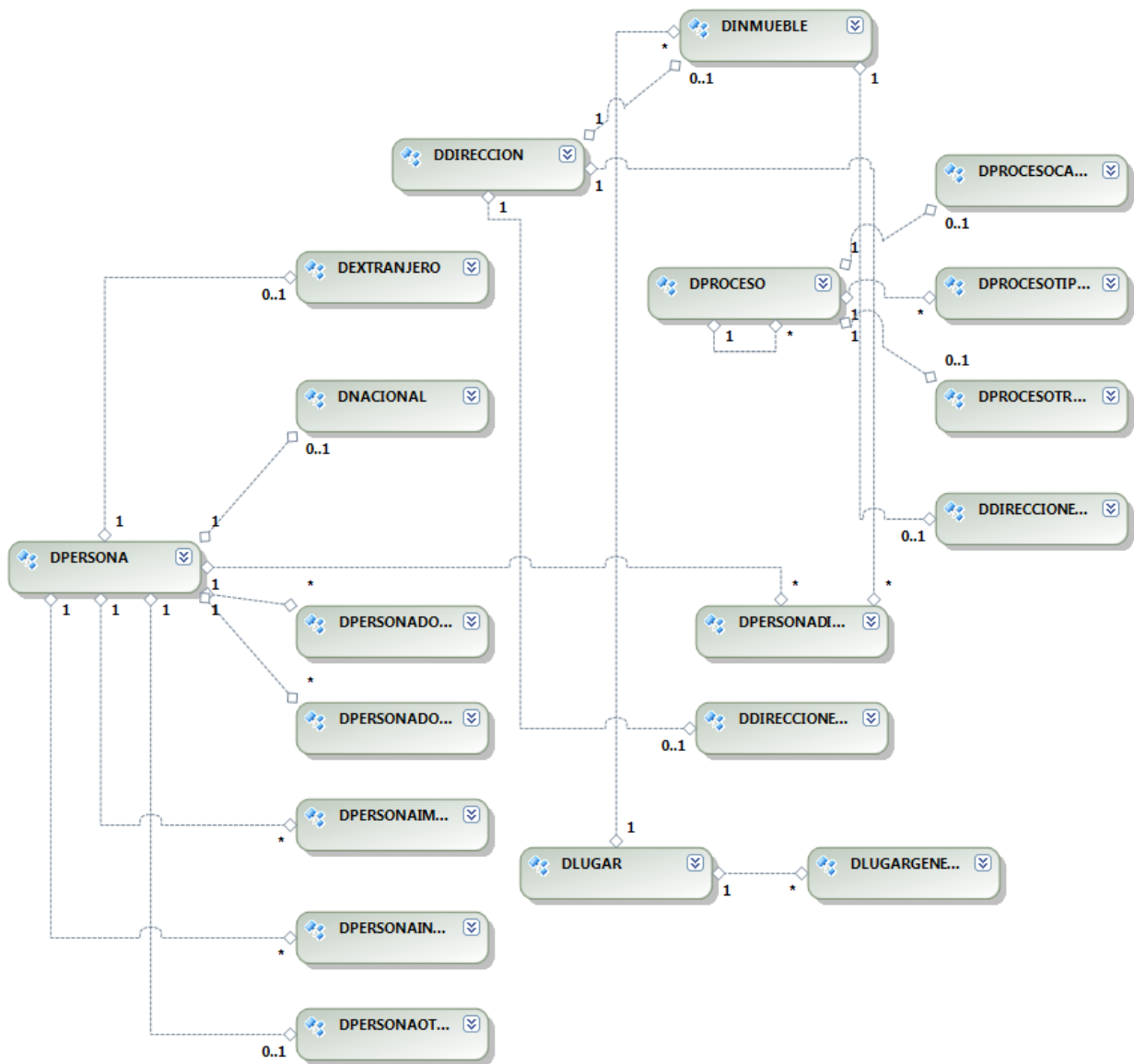


Ilustración 21 Fragmento del Modelo de Datos del Paquete de Servicios.

**Nombre de la Entidad:** DDIRECCION

**Descripción de la Entidad:** Representa las direcciones de las personas.

**Servicio:** SearchAddress.

Atributo	Tipo de Dato	Nulo	Descripción
IDDIRECCION	<i>Raw(16)</i>	No	Identificador de la dirección.
TIRAANTERIOR	<i>Varchar2(500)</i>	Si	Tira de dirección de la propiedad.
TIRACODIFICADA	<i>Varchar2(500)</i>	Si	Tira codificada de la propiedad.
MIGRACION	<i>Number(1)</i>	No	Bit que indica si la dirección (tira) fue migrada (1) o generada en la BD (0).
COORDENADAS	<i>Varchar2(500)</i>	Si	Coordenadas de la dirección.

**Tabla 52 Entidad DDIRECCION.**

**Nombre de la Entidad:** DPROCESO

**Descripción de la Entidad:** Almacena los datos correspondientes a un proceso.

**Servicio:** *ProccesRequest.*

Atributo	Tipo de Dato	Nulo	Descripción
IDSERIALPROCESO	<i>Raw(16)</i>	No	Identificador del proceso.
INSTANCE_ID	<i>Varchar2(36)</i>	No	Identificador de la instancia del <i>workflow</i> .
SERIAL	<i>Varchar2(500)</i>	No	Serial del <i>workflow</i> .
FECHAINICIO	<i>Date</i>	No	Fecha de inicio del proceso.
FECHAFIN	<i>Date</i>	No	Fecha de fin del proceso.
SERIALPROCESOPADRE	<i>Raw(16)</i>	No	Identificador del proceso padre.
IDOFICINA	<i>NUMBER(10)</i>	No	Identificador de la oficina donde se realiza el proceso.
IDTIPOPROCESO	<i>NUMBER(4)</i>	No	Identificador del tipo de proceso.
IDESTADOFINALPROCESO	<i>NUMBER(8)</i>	Si	Identificador del estado final del proceso.
IDCAUSACANCELARDENEGAR	<i>NUMBER(4)</i>	Si	Identificador de causa de cancelación.

**Tabla 53 Entidad DPROCESO.**

**Nombre de la Entidad:** DPERSONA

**Descripción de la Entidad:** Entidad que representa la generalización de la persona

**Servicio:** *SearchPerson, PersonManager*

Atributo	Tipo de Dato	Nulo	Descripción
IDPERSONA	<i>UNIQUEID</i>	No	Identificador único de la persona.
FALLECIDO	<i>BIT</i>	No	Bit que representa si una persona esta fallecida o no.
IDTIPOPERSONA	<i>NUMERIC(4, 0)</i>	No	Identificador para cada tipo de persona.
IDCONDICIONMIGRATORIA	<i>NUMERIC(4, 0)</i>	No	Identificador para la condición de residencia de la persona.

**Tabla 54 Entidad DPERSONA**

## Anexo IV. DESCRIPCIÓN DE ENTIDADES DE ENCODERSERVICES

**Nombre de la Entidad:** NAREAAPLICACION

**Descripción de la Entidad:** Nomenclador de las áreas de aplicación.

**Servicio:** *EncoderService.*

Atributo	Tipo de Dato	Nulo	Descripción
NAREAAPLICACION	<i>NUMBER(4, 0)</i>	No	Dominio del tipo de dato numérico para las secuencias que se generan como identificadores de los nomencladores.
DESCRIPCION	<i>Varchar2(4000)</i>	No	Descripción del nomenclador.
NAREAAPLICACION	<i>NUMBER(4, 0)</i>	No	Identificador único del tipo de aplicación.

**Tabla 55 Descripción de la entidad de *EncoderServices* NAREAAPLICACION**

**Nombre de la Entidad:** NATRIBUTODOCUMENTO

**Descripción de la Entidad:** Nomenclador de los distintos atributos de cada documento.

**Servicio:** *EncoderService.*

Atributo	Tipo de Dato	Nulo	Descripción
NATRIBUTODOCUMENTO	<i>NUMBER(4, 0)</i>	No	Identificador para el atributo que tiene

			cada documento.
NATRIBUTODOCUMENTO	<i>VARCHAR2(4000)</i>	No	Describe el atributo del documento.
NATRIBUTODOCUMENTO	<i>DATE</i>	No	Fecha de registro del atributo del documento.
NATRIBUTODOCUMENTO	<i>NUMBER(4, 0)</i>	No	Identificador del tipo de documento.
NATRIBUTODOCUMENTO	<i>NUMBER(4, 0)</i>	No	Identificador para el tipo de dato.

**Tabla 56 Descripción de la entidad de EncoderServices NATRIBUTODPCUMENTO**

<b>Nombre de la Entidad:</b> NCATEGORIAEXTRANJERO			
<b>Descripción de la Entidad:</b> Nomenclador para las distintas categorías de extranjeros.			
<b>Servicio:</b> <i>EncoderService</i> .			
Atributo	Tipo de Dato	Nulo	Descripción
NCATEGORIAEXTRANJERO	<i>NUMBER(4, 0)</i>	No	Identificador único para cada categoría del Extranjero.
NCATEGORIAEXTRANJERO	<i>VARCHAR2(80)</i>	No	Descripción del nomenclador.
NCATEGORIAEXTRANJERO	<i>DATE</i>	No	Fecha de registro de la categoría de extranjero.

**Tabla 57 Descripción de la entidad de EncoderServices NCATEGORIAEXTRANJERO**

<b>Nombre de la Entidad:</b> NCAUSACANCELARDENEGAR			
<b>Descripción de la Entidad:</b> Nomenclador para causas por lo que un proceso pasa a estado final cancelado o denegado.			
<b>Servicio:</b> <i>EncoderService</i> .			
Atributo	Tipo de Dato	Nulo	Descripción
IDCAUSACANCELARDENEGAR	<i>NUMBER(4, 0)</i>	No	Identificador para el nomenclador causa por lo que el proceso es cancelado o denegado.
DESCRIPCION	<i>VARCHAR2(80)</i>	No	Descripción del nomenclador.
FECHAREGISTRO	<i>DATE</i>	No	Fecha en que se registra las causas.

IDTIPOPROCESO	NUMBER(4, 0)	No	Identificador del tipo de proceso.
---------------	--------------	----	------------------------------------

Tabla 58 Descripción de la entidad de *EncoderServices* NCAUSACANCELARDENEGAR

<b>Nombre de la Entidad:</b> NREQUISITOTIPOTRAMITE			
<b>Descripción de la Entidad:</b> Nomenclador que representa los requisitos de un tipo de trámite dado.			
<b>Servicio:</b> <i>EncoderService</i> .			
Atributo	Tipo de Dato	Nulo	Descripción
IDTIPOTRAMITE	NUMBER(4, 0)	No	Identificador para el nomenclador tipo de trámite.
IDREQUISITO	RAW(16)	No	Identificador único del requisito del trámite.
FECHAREGISTRO	DATE	No	Fecha en que se registra los requisitos de un tipo de trámite asociado.

Tabla 59 Descripción de la entidad de *EncoderServices* NREQUISITOTIPOTRAMITE

<b>Nombre de la Entidad:</b> NREQUISITOTRAMITE			
<b>Descripción de la Entidad:</b> Nomenclador que registra los requisitos de cada trámite.			
<b>Servicio:</b> <i>EncoderService</i> .			
Atributo	Tipo de Dato	Nulo	Descripción
DREQUISITO	RAW(16)	No	Identificador único del requisito del trámite.
DESCRIPCION	VARCHAR2(80)	No	Descripción del nomenclador.
FECHAREGISTRO	DATE	No	Fecha en que se registra los requisitos del tipo de trámite.
IDTIPODOCUMENTO	NUMBER(4, 0)	No	Identificador del tipo de documento.
TRAMITESASOCIADOS	RAW(16)	No	Identificador único del requisito del trámite.

Tabla 60 Descripción de la entidad de *EncoderServices* NREQUISITOTRAMITE

<b>Nombre de la Entidad:</b> NTIPOAPLICACION			
<b>Descripción de la Entidad:</b> Nomenclador que representa los tipos de aplicación.			
<b>Servicio:</b> <i>EncoderService</i> .			



Atributo	Tipo de Dato	Nulo	Descripción
IDTIPOAPLICACION	NUMBER(4, 0)	No	Identificador único del tipo de aplicación.
DESCRIPCION	VARCHAR2(4000)	No	Descripción del tipo de aplicación.
FECHAREGISTRO	DATE	No	Fecha en que se registra el tipo de aplicación.

**Tabla 61 Descripción de la entidad de *EncoderServices* NTIPOAPLICACION**

**Nombre de la Entidad: NTIPODATO**

**Descripción de la Entidad:** Nomenclador que representa los tipos de datos a los que pertenece cada atributo de los documentos.

**Servicio:** *EncoderService*.

Atributo	Tipo de Dato	Nulo	Descripción
DESCRIPCION	VARCHAR2(4000)	No	Descripción del tipo de dato.
EXPRESIONREGULAR	VARCHAR2(60)	No	Expresión regular con la cual se puede representar todos los datos del atributo específico.
FECHAREGISTRO	DATE	No	Fecha de registro de un tipo de dato.

**Tabla 62 Descripción de la entidad de *EncoderServices* NTIPODATO**

**Nombre de la Entidad: NTIPODOCIDENTIFICACION**

**Descripción de la Entidad:** Nomenclador que representa los tipos de documentos de identificación

**Servicio:** *EncoderService*.

Atributo	Tipo de Dato	Nulo	Descripción
IDTIPODOCIDENTIFICACION	NUMBER(4, 0)	No	Dominio del tipo de dato numérico para las secuencias que se generan como identificadores de los nomencladores.
DESCRIPCION	VARCHAR2(80)	No	Descripción del nomenclador.
FECHAREGISTRO	DATE	No	Fecha de registro de un tipo de documento identificativo.

FECHACIERRE	DATE	No	Dominio del tipo de dato date para las fechas.
-------------	------	----	--

Tabla 63 Descripción de la entidad de *EncoderServices* NTIPODOCIDENTIFICACION

<b>Nombre de la Entidad:</b> NTIPODOCUMENTO			
<b>Descripción de la Entidad:</b> Nomenclador que representa los tipos de documentos.			
<b>Servicio:</b> <i>EncoderService</i> .			
Atributo	Tipo de Dato	Nulo	Descripción
IDTIPODOCUMENTO	NUMBER(4, 0)	No	Identificador del tipo de documento.
DESCRIPCION	VARCHAR2(4000)	No	Descripción del tipo de documento.
XMLCAPTACION	VARCHAR2(60)	No	XML de la captación del tipo de documento.
FECHAREGISTRO	DATE	No	Fecha de registro del tipo de documento.

Tabla 64 Descripción de la entidad de *EncoderServices* NTIPODOCUMENTO

<b>Nombre de la Entidad:</b> NTIPOMEDIDACNI			
<b>Descripción de la Entidad:</b> Nomenclador que representa los tipos de medidas que son aplicadas para cada tipo de proceso que se genera en el Centro Nacional de Internamiento.			
<b>Servicio:</b> <i>EncoderService</i> .			
Atributo	Tipo de Dato	Nulo	Descripción
IDTIPOMEDIDACNI	NUMBER(4, 0)	No	Identificador del tipo de medida aplicada a un tipo de proceso en el CNI.
DESCRIPCION	VARCHAR2(80)	No	Descripción del nomenclador.
FECHAREGISTRO	DATE	No	Fecha de registro del tipo de medida aplicada a un tipo de proceso en el CNI.

Tabla 65 Descripción de la entidad de *EncoderServices* NTIPOMEDIDACNI

<b>Nombre de la Entidad:</b> NTIPOPROCESOBLOQUEO			
<b>Descripción de la Entidad:</b> Nomenclador que representa los tipos de procesos que bloquean.			

**Servicio:** *EncoderService*.

Atributo	Tipo de Dato	Nulo	Descripción
IDTIPOPROCESO	<i>NUMBER(4, 0)</i>	No	Identificador del tipo de proceso.
IDPROCESOBLOQUEADO	<i>NUMBER(4, 0)</i>	No	Identificador del tipo de proceso de bloqueo.
FECHAREGISTRO	<i>DATE</i>	No	Fecha de registro del tipo de proceso de bloqueo.

**Tabla 66 Descripción de la entidad de *EncoderServices* NTIPOPROCESOBLOQUEO**

**Nombre de la Entidad:** NTIPOPROCESOTIPOTRAMITE

**Descripción de la Entidad:** Nomenclador que representa los tipos de trámites que pueden existir en un tipo de proceso.

**Servicio:** *EncoderService*.

Atributo	Tipo de Dato	Nulo	Descripción
IDTIPOPROCESO	<i>NUMBER(4, 0)</i>	No	Identificador del tipo de proceso.
IDTIPOTRAMITE	<i>NUMBER(4, 0)</i>	No	Identificador para el nomenclador tipo de trámite.
FECHAREGISTRO	<i>DATE</i>	No	Fecha en que se registra el tipo de trámite de un tipo de proceso asociado.

**Tabla 67 Descripción de la entidad de *EncoderServices* NTIPOPROCESOTIPOTRAMITE**

## Anexo V. DISEÑO DE CASOS DE PRUEBAS

Escenario	Descripción	Parámetros	Respuesta del sistema	Flujo central
<i>InsertNationalPerson</i>	<i>Inserta los datos relacionados a una persona nacional.</i>	<i>Persona Nacional</i>	<i>Inserta la persona Nacional</i>	<i>Crear el objeto de entrada y compararlo con el resultado</i>

<i>BasicInsertNationalPerson</i>	<i>Inserta los datos básicos relacionados a una persona nacional.</i>	<i>Persona Nacional</i>	<i>Inserta la persona Nacional</i>	<i>obtenido de la invocación del servicio.</i>
<i>InsertForeignPerson</i>	<i>Inserta los datos relacionados a una persona extranjera.</i>	<i>Persona Extranjera</i>	<i>Inserta la persona extranjera</i>	
<i>BasicInsertForeignPerson</i>	<i>Inserta los datos básicos relacionados a una persona extranjera.</i>	<i>Persona Extranjera</i>	<i>Inserta la persona extranjera</i>	
<i>InsertNonRegisterPerson</i>	<i>Inserta los datos relacionados a una persona extranjera.</i>	<i>Persona No Registrada</i>	<i>Inserta la persona no registrada.</i>	
<i>BasicInsertNonRegisterPerson</i>	<i>Inserta los datos básicos relacionados a una persona no registrada.</i>	<i>Persona No Registrada</i>	<i>Inserta los datos básicos relacionados a una persona no registrada.</i>	
<i>UpdateNationalPerson</i>	<i>Modifica los datos a una persona nacional.</i>	<i>Persona Nacional</i>	<i>Modifica los datos a una persona nacional.</i>	

<i>UpdateForeignPerson</i>	<i>Modifica los datos a una persona extranjera.</i>	<i>Persona Extranjera</i>	<i>Modifica los datos a una persona extranjera.</i>
<i>UpdateNonRegisterPerson</i>	<i>Modifica los datos a una persona no registrada.</i>	<i>Persona No Registrada</i>	<i>Modifica los datos a una persona no registrada.</i>
<i>RegisterNationalPerson</i>	<i>Registra una persona no registrada a nacional.</i>	<i>Persona Nacional</i>	<i>Registra una persona no registrada a nacional.</i>
<i>RegisterForeignPerson</i>	<i>Registra una persona no registrada a extranjera.</i>	<i>Persona Extranjera</i>	<i>Registra una persona no registrada a extranjera.</i>
<i>ChangeDeathStatus</i>	<i>Cambia el estado de fallecido.</i>	<i>Boolean</i>	<i>Cambia el estado de fallecido.</i>
<i>ChangeAddress</i>	<i>Cambia la dirección de una persona.</i>	<i>ID Persona, ID Dirección</i>	<i>Cambia la dirección de una persona.</i>
<i>GenerateIdentityNumber</i>	<i>Genera el número de identidad.</i>	<i>Fecha Nacimiento, Sexo</i>	<i>Genera el número de identidad.</i>

**Tabla 68** Diseño de caso de prueba para el servicio **PersonManagerService**

Escenario	Descripción	Parámetros	Respuesta del sistema	Flujo Prueba
<i>SearchPerson</i>	<i>Busca una persona.</i>	<i>Persona Similar</i>	<i>Busca y devuelve una persona por datos similares.</i>	<i>Crear el objeto de entrada y compararlo con el resultado obtenido de la invocación del servicio.</i>
<i>SearchPersonById</i>	<i>Busca una persona por el ID.</i>	<i>ID Persona</i>	<i>Busca y devuelve una persona por el id.</i>	
<i>SearchPersonByIdentityNumber</i>	<i>Busca una persona por el número de identidad.</i>	<i>Número de Identidad</i>	<i>Busca y devuelve una persona por el número de identidad.</i>	
<i>SearchNationalPerson</i>	<i>Busca una persona nacional.</i>	<i>Persona Nacional Similar</i>	<i>Busca y devuelve una persona nacional por datos similares.</i>	
<i>SearchNationalPersonByAddresses</i>	<i>Busca una persona nacional por la dirección.</i>	<i>Persona Nacional Similar, Lugar</i>	<i>Busca y devuelve una persona nacional por la dirección.</i>	

<i>SearchNationalPersonById</i>	<i>Busca una persona nacional por el ID.</i>	<i>ID Persona Nacional</i>	<i>Busca y devuelve una persona nacional por el id.</i>
<i>SearchNationalPersonByIdentity Number</i>	<i>Busca una persona nacional por el número de identidad.</i>	<i>Número de Identidad</i>	<i>Busca y devuelve una persona nacional por el número de identidad.</i>
<i>SearchForeignPerson</i>	<i>Busca una persona extranjera.</i>	<i>Persona Extranjera Similar</i>	<i>Busca y devuelve una persona extranjera.</i>
<i>SearchForeignPersonById</i>	<i>Busca una persona extranjera por el id.</i>	<i>ID Persona Extranjera</i>	<i>Busca y devuelve una persona extranjera por el id.</i>
<i>SearchForeignPersonByIdentity Number</i>	<i>Busca una persona extranjera por el número de identidad.</i>	<i>Número de Identidad</i>	<i>Busca y devuelve una persona extranjera por el número de identidad.</i>

<i>SearchForeignPersonByIdentity Document</i>	<i>Busca una persona extranjera por los documentos de identidad.</i>	<i>Documento de Identidad</i>	<i>Busca y devuelve una persona extranjera por los documentos de identidad.</i>	
<i>SearchNonRegisterPerson</i>	<i>Busca una persona no registrada.</i>	<i>Persona Nacional Similar</i>	<i>Busca y devuelve una persona no registrada similar.</i>	
<i>SearchNonRegisterPersonById(</i>	<i>Busca una persona no registrada por el ID.</i>	<i>ID Persona No Registrada</i>	<i>Busca y devuelve una persona no registrada por el ID.</i>	
<i>SearchNonRegisterPersonByIdentityNumber</i>	<i>Busca una persona no registrada por el número de identidad.</i>	<i>Número de Identidad</i>	<i>Busca y devuelve una persona no registrada por el número de identidad.</i>	

**Tabla 69** Diseño de caso de prueba para el servicio *SearchPersonService*

<b>Escenario</b>	<b>Descripción</b>	<b>Parámetros</b>	<b>Respuesta del sistema</b>	<b>Flujo Prueba</b>
------------------	--------------------	-------------------	------------------------------	---------------------



<i>SearchAddressByAddressId</i>	<i>Busca una dirección por el id de dirección.</i>	<i>ID Dirección</i>	<i>Busca una dirección por el id de dirección.</i>	<i>Crear el objeto de entrada y compararlo con el resultado obtenido de la invocación del servicio.</i>
<i>SearchAddressByIdPerson</i>	<i>Busca una dirección por el ID de una persona.</i>	<i>ID Persona</i>	<i>Busca una dirección por el ID de una persona.</i>	
<i>SearchAddressByIdPersonWithParameter</i>	<i>Busca una dirección por el ID de una persona y otros parámetros.</i>	<i>ID Persona, Activa, ID Tipo Dirección</i>	<i>Busca una dirección por el ID de una persona y otros parámetros.</i>	
<i>SearchAddressByIdentityNumber</i>	<i>Busca una dirección por el número de identidad de una persona.</i>	<i>Número de Identidad</i>	<i>Busca una dirección por el número de identidad de una persona.</i>	
<i>SearchAddressByIdentityNumberWithParameter</i>	<i>Busca una dirección por el número de identidad de una persona y otros parámetros.</i>	<i>Número de Identidad, Activa, ID Tipo Dirección</i>	<i>Busca una dirección por el número de identidad de una persona y otros parámetros.</i>	

<i>SearchAddressForStrip</i>	<i>Busca una dirección por la tira de la dirección.</i>	<i>Tira de Dirección</i>	<i>Busca una dirección por la tira de la dirección.</i>
<i>SearchAddressForGeneralPlace</i>	<i>Busca una dirección por parámetros de Lugar General.</i>	<i>ID Lugar General, Descripción Lugar General</i>	<i>Busca una dirección por parámetros de Lugar General.</i>
<i>SearchAddressForIdGeneralPlace</i>	<i>Busca una dirección por el id de Lugar General.</i>	<i>ID Lugar General</i>	<i>Busca una dirección por el id de Lugar General.</i>

**Tabla 70** Diseño de caso de prueba para el servicio *SearchAddressService*

<b>Escenario</b>	<b>Descripción</b>	<b>Parámetros</b>	<b>Respuesta del sistema</b>	<b>Flujo Prueba</b>
<i>GetPersonByPhonetic</i>	<i>Busca y devuelve una persona por los datos biométricos de esta y por tipos de información operativa.</i>	<i>Nombre, Segundo Nombre, Nombre familiares, Apellidos, Tipo de Información</i>	<i>Busca y devuelve una persona por los datos biométricos de esta y por tipos de información operativa.</i>	<i>Crear el objeto de entrada y compararlo con el resultado obtenido de la invocación del servicio.</i>
<i>GetPersonByIdentityNumber</i>	<i>Busca y devuelve una persona por el número de</i>	<i>ID Persona</i>	<i>Busca y devuelve una persona por el número de</i>	

	<i>identidad y por tipos de información operativa.</i>		<i>identidad y por tipos de información operativa.</i>	
--	--	--	--	--

**Tabla 71** Diseño de caso de prueba para el servicio *OperativeRecordService*

Escenario	Descripción	Parámetros	Respuesta del sistema	Flujo Prueba
<i>Request</i>	<i>Verifica si se le puede asignar un proceso a una persona dada y se lo adiciona.</i>	Solicitud de Proceso	<i>Verifica si se le puede asignar un proceso a una persona dada y se lo adiciona.</i>	<i>Crear el objeto de entrada y compararlo con el resultado obtenido de la invocación del servicio.</i>
<i>AssignInstanceld</i>	<i>Le asigna una instancia a un proceso.</i>	ID Proceso, Instancia	<i>Le asigna una instancia a un proceso.</i>	

**Tabla 72** Diseño de caso de prueba para el servicio *ProcessRequestService*

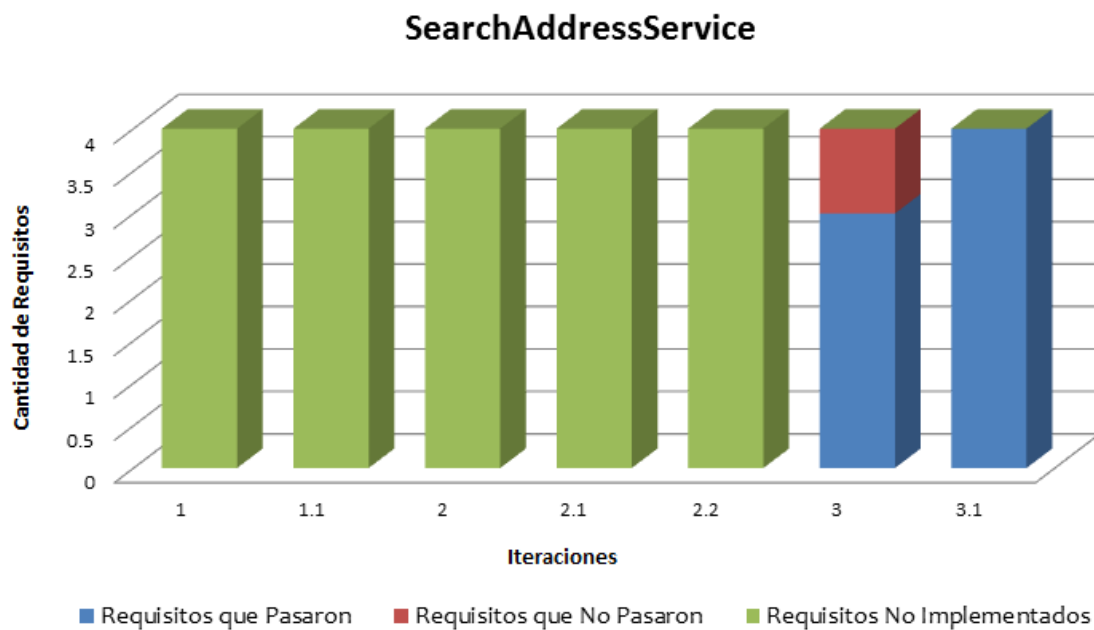
## Anexo VI. PRUEBAS UNITARIAS.

### Servicio *SearchAddressService*:

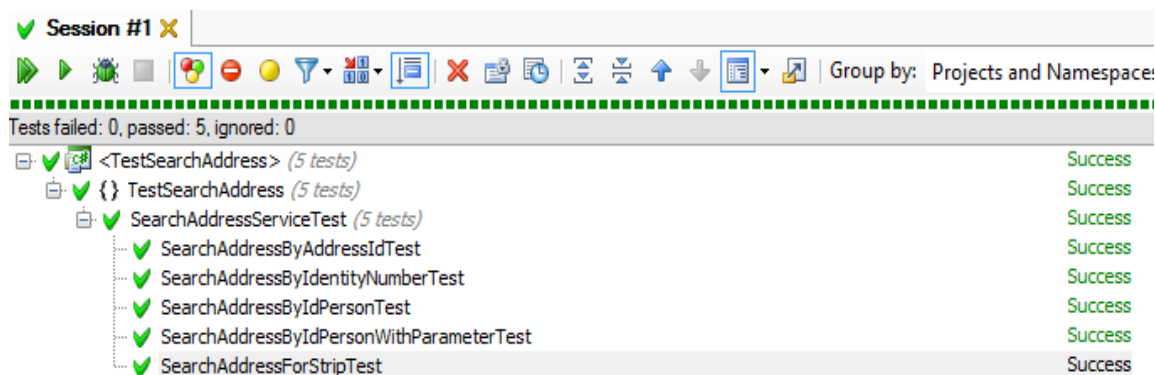
<b><i>SearchAddressService</i></b>							
Requisitos	1	1.1	2	2.1	2.2	3	3.1
Buscar listado de direcciones dado el identificador de una persona.	No evaluado	No evaluado	No evaluado	No evaluado	No evaluado	Pasó	Pasó
Buscar listado de direcciones dado el ci de una persona.	No evaluado	No evaluado	No evaluado	No evaluado	No evaluado	Pasó	Pasó
Buscar listado de	No evaluado	No evaluado	No evaluado	No evaluado	No evaluado	No Pasó	Pasó

direcciones dada una tira de dirección.							
Buscar listado de direcciones por parámetros.	No evaluado	No evaluado	No evaluado	No evaluado	No evaluado	Pasó	Pasó

**Tabla 73 Pruebas de Requisitos por Iteración del servicio SearchAddressService**



**Ilustración 22 Gráfica de las pruebas de Requisitos por Iteración del servicio SearchAddressService**



**Ilustración 23 Resultado final de las pruebas a SearchAddressService**

**Servicio SearchPersonService:**

<b>SearchPersonService</b>							
<b>Requisitos</b>	<b>1</b>	<b>1,1</b>	<b>2</b>	<b>2,1</b>	<b>2,2</b>	<b>3</b>	<b>3,1</b>
Buscar listado de personas dada una persona semejante.	No evaluado	No evaluado	Pasó	Pasó	Pasó	Pasó	Pasó
Buscar una persona según el id de la persona.	No evaluado	No evaluado	Pasó	Pasó	Pasó	Pasó	Pasó
Buscar una persona según el número de identidad de la persona.	No evaluado	No evaluado	No Pasó	No Pasó	Pasó	Pasó	Pasó
Buscar personas nacionales similares a una persona dada.	No evaluado	No evaluado	Pasó	Pasó	Pasó	Pasó	Pasó
Buscar personas	No evaluado	No evaluado	No evaluado	No evaluado	No evaluado	Pasó	Pasó

que tengan una dirección dada.							
Buscar persona nacional dado el identificador de persona.	No evaluado	No evaluado	Pasó	Pasó	Pasó	Pasó	Pasó
Buscar persona nacional dado el número de identidad.	No evaluado	No evaluado	Pasó	Pasó	Pasó	Pasó	Pasó
Buscar personas extranjeras, dada una persona similar.	No evaluado	No evaluado	Pasó	Pasó	Pasó	Pasó	Pasó
Buscar persona extranjera dado un id de persona.	No evaluado	No evaluado	Pasó	Pasó	Pasó	Pasó	Pasó
Buscar persona extranjera dados el documento de identidad y una persona semejante.	No evaluado	No evaluado	No evaluado	No evaluado	No evaluado	No Pasó	Pasó
Buscar personas no registradas dada una persona similar.	No evaluado	No evaluado	No evaluado	No evaluado	No evaluado	Pasó	Pasó
Buscar persona no registrada dado el identificador de una	No evaluado	No evaluado	No evaluado	No evaluado	No evaluado	Pasó	Pasó

persona.							
Buscar persona no registrada dado el número de identidad.	No evaluado	No evaluado	Pasó	Pasó	Pasó	Pasó	Pasó
Obtener foto de una persona dado el identificador de la foto.	No evaluado	No evaluado	No Pasó	No Pasó	Pasó	Pasó	Pasó

Tabla 74 Pruebas de Requisitos por Iteración del servicio *SearchPersonService*

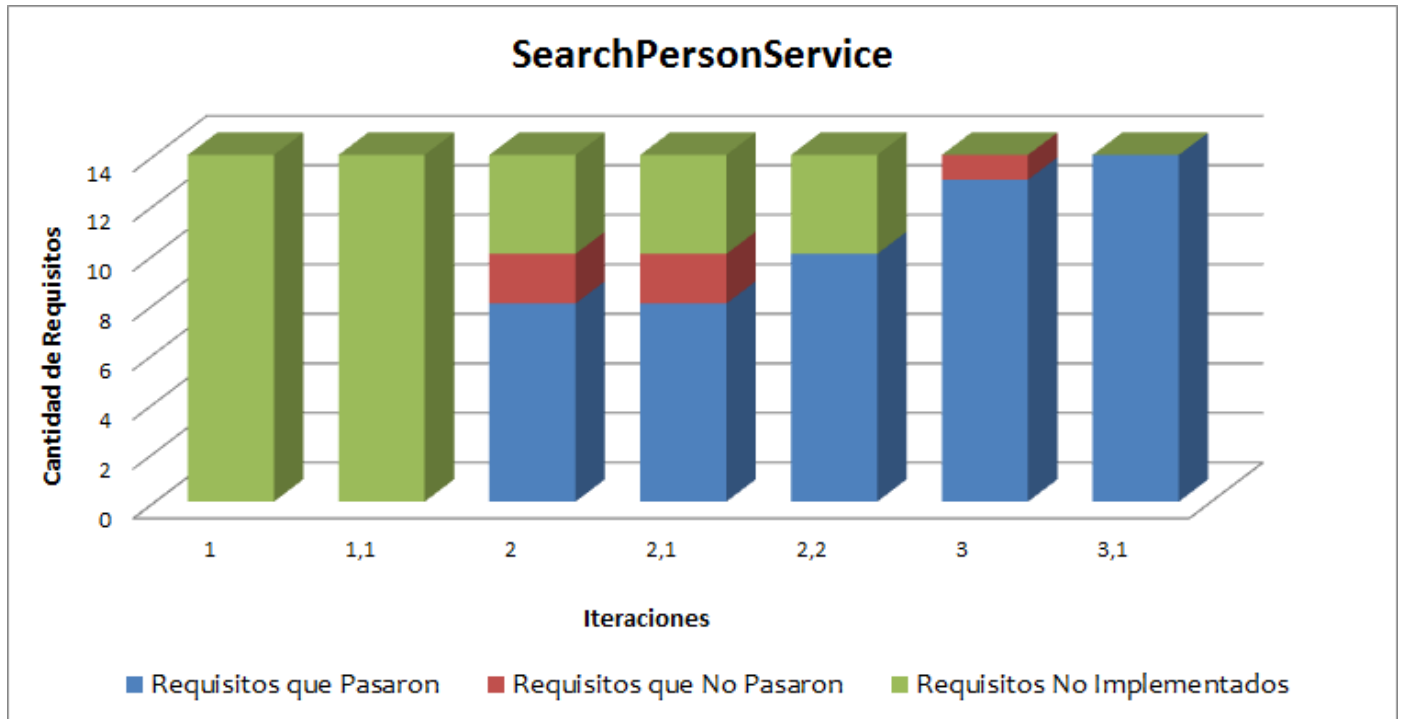


Ilustración 24 Gráfica de las pruebas de Requisitos por Iteración del servicio *SearchPersonService*

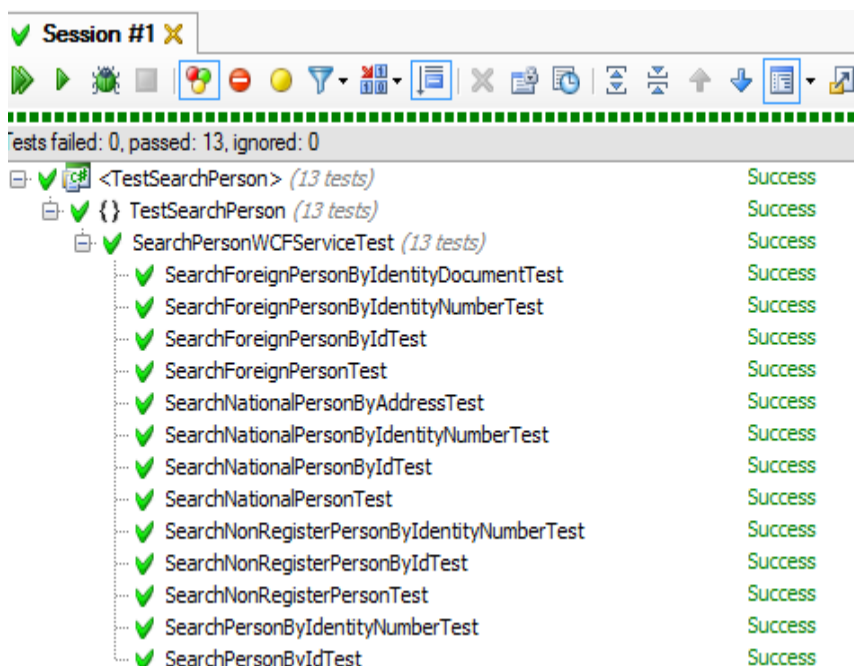


Ilustración 25 Resultado final de las pruebas a *SearchPersonService*

**Servicio *PersonManagerService*:**

<i>PersonManagerService</i>							
Requisitos	1	1,1	2	2,1	2,2	3	3,1
Insertar persona nacional.	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó
Insertar básico persona nacional.	No Pasó	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó
Insertar persona extranjera.	No Pasó	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó
Insertar básico persona extranjero.	No Evaluado	No Evaluado	No Pasó	No Pasó	Pasó	Pasó	Pasó
Insertar persona no registrada.	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó



Insertar básico persona no registrado.	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó
Modificar persona nacional.	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó
Modificar persona extranjera.	No Evaluado	No Evaluado	No Evaluado	No Evaluado	No Evaluado	Pasó	Pasó
Modificar persona no registrada.	No Evaluado	No Evaluado	No Evaluado	No Evaluado	No Evaluado	No Pasó	Pasó
Modificar estado fallecido de la persona.	No Evaluado	No Evaluado	No Evaluado	No Evaluado	No Evaluado	Pasó	Pasó
Modificar dirección permanente de la persona.	No Evaluado	No Evaluado	No Evaluado	No Evaluado	No Evaluado	No Pasó	Pasó

**Tabla 75 Pruebas de Requisitos por Iteración del servicio *PersonManagerService***

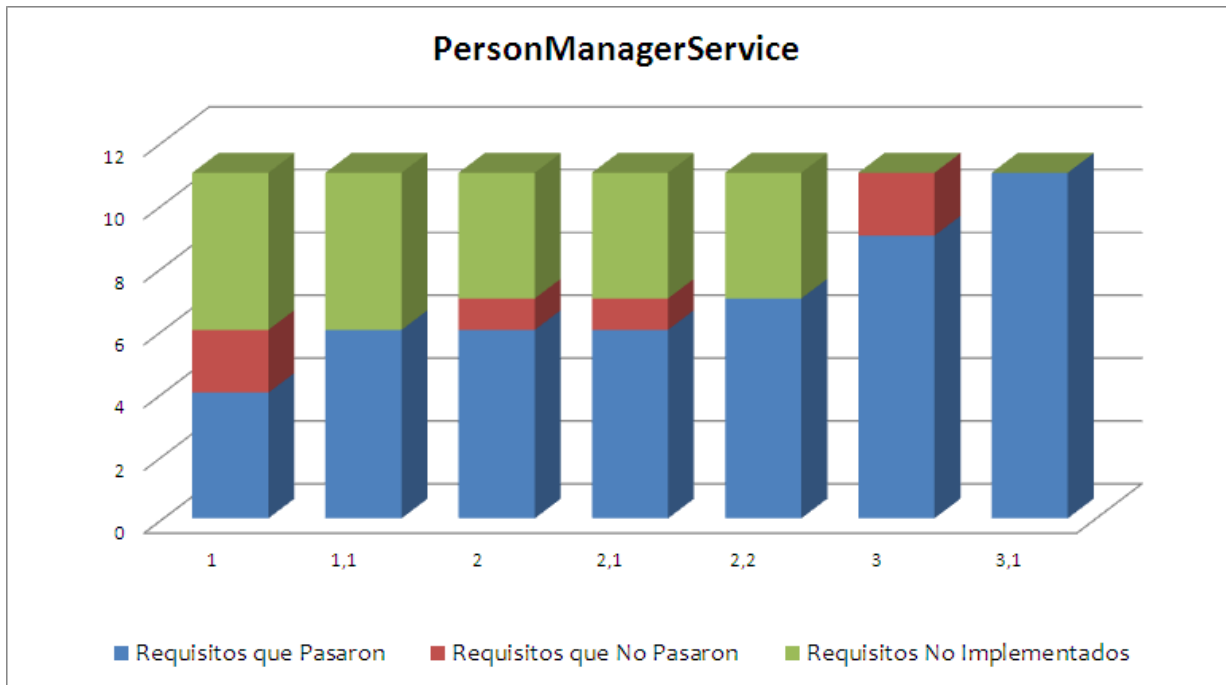


Ilustración 26 Gráfica de las pruebas de Requisitos por Iteración del servicio *PersonManagerService*

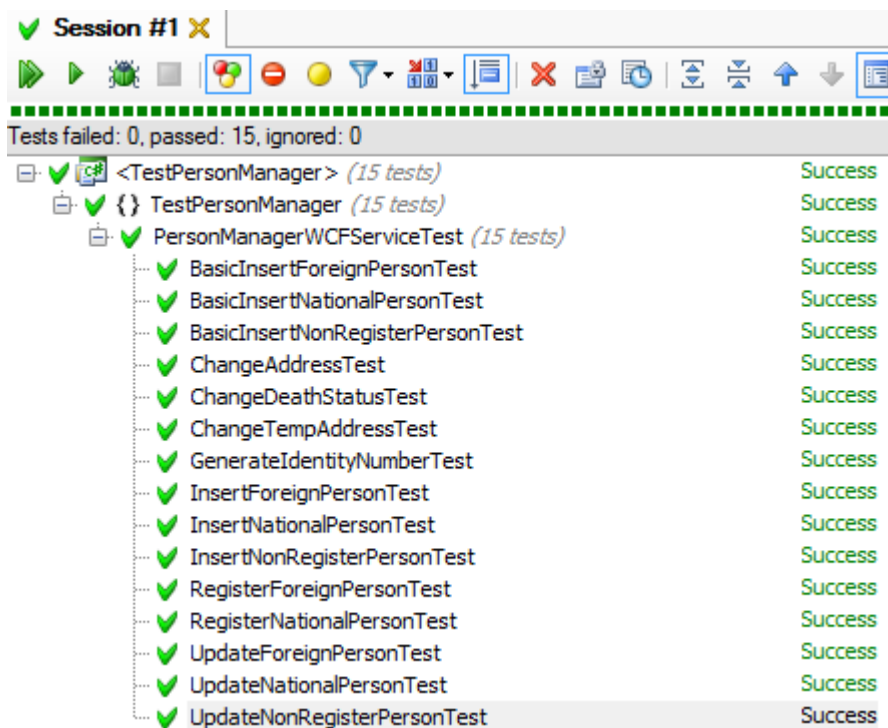


Ilustración 27 Resultado final de las pruebas a *PersonManagerService*

**Servicio *OperativeRecordService*:**

<i>OperativeRecordService</i>							
Requisitos	1	1,1	2	2,1	2,2	3	3,1
Búsqueda fonética de una persona en los registros operativos según datos biográficos.	No evaluado	No evaluado	Pasó	Pasó	Pasó	Pasó	Pasó
Buscar una persona en los registros operativos dado su número de identidad.	No evaluado	No evaluado	Pasó	Pasó	Pasó	Pasó	Pasó
Buscar una persona en los registros operativos dado un tipo de información.	No evaluado	No evaluado	Pasó	Pasó	Pasó	Pasó	Pasó

Tabla 76 Pruebas de Requisitos por Iteración del servicio *OperativeRecordService*



Ilustración 28 Gráfica de las pruebas de Requisitos por Iteración del servicio *OperativeRecordService*

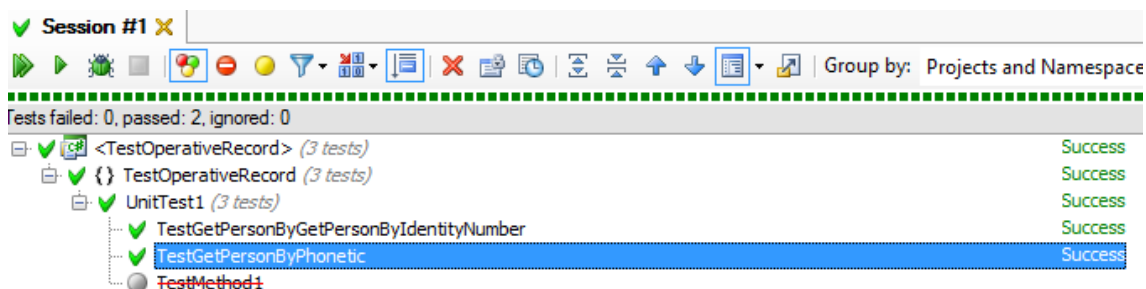


Ilustración 29 Resultado final de las pruebas a *OperativeRecordService*

### Servicio *ProcessRequestService*:

<i>ProcessRequestService</i>							
Requisitos	1	1,1	2	2,1	2,2	3	3,1
Comprobar si se puede realizar un proceso a una persona dada.	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó
Adicionar un proceso dado a una persona.	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó	Pasó

Tabla 77 Pruebas de Requisitos por Iteración del servicio *ProcessRequestService*

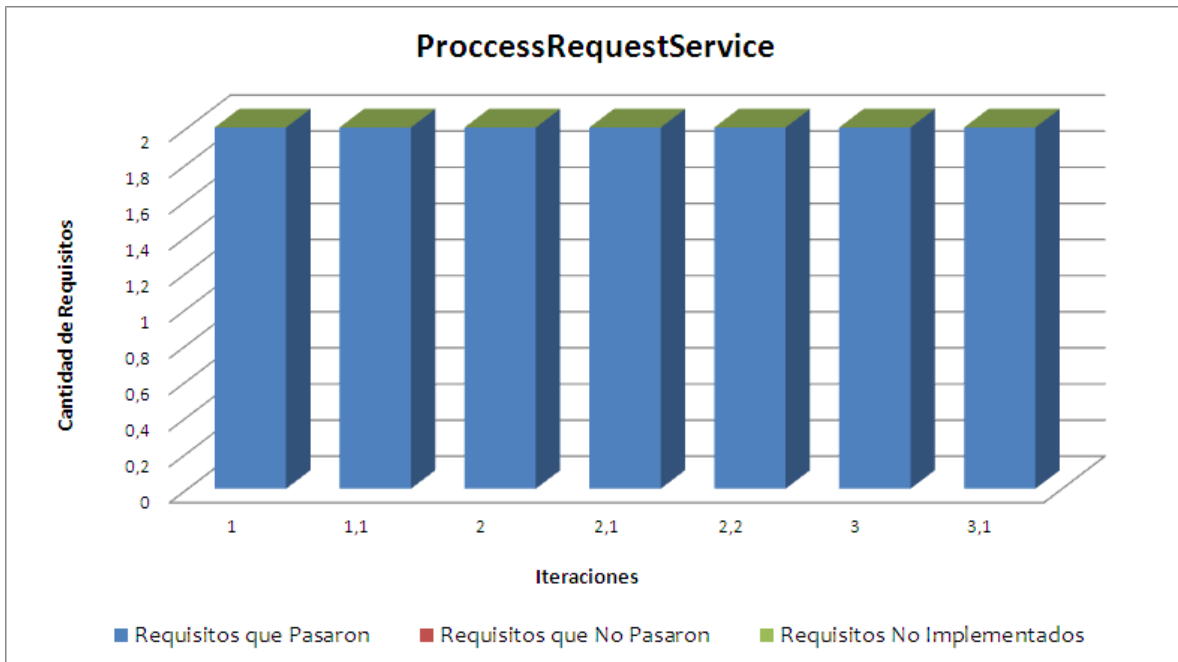


Ilustración 30 Gráfica de las pruebas de Requisitos por Iteración del servicio *ProcessRequestService*

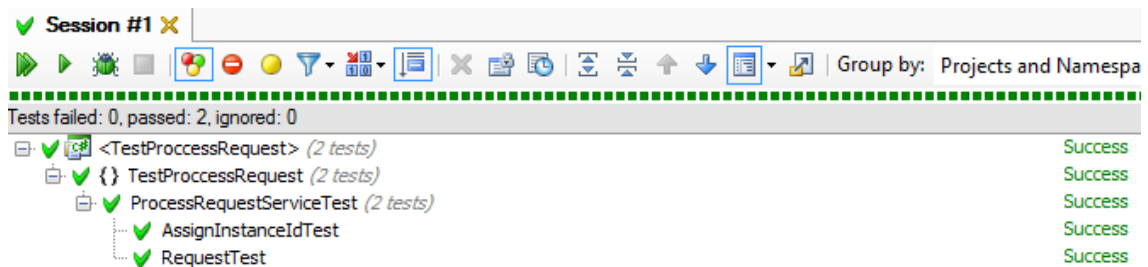


Ilustración 31 Resultado final de las pruebas a *ProcessRequestService*

## Anexo VII. PRUEBAS DE ACEPTACIÓN



## Acta de Liberación de Productos Software

**Fecha de liberación: 14 de mayo de 2011**

**Emitida a favor de: Proyecto Identidad, Inmigración y Extranjería de Cuba.**

### 1. Datos del producto

Artefacto	Versión	Estado final	Cantidad Iteraciones	Tipos de pruebas realizadas
SPDI	1.0	0	3	Pruebas Funcionales
Manual de Usuario	1.1	0	3	Evaluación Estática

Yadira Machado Peña

Nombre y Apellidos  
Responsable Calisoft



Yudenia Ramírez Mastrapa

Nombre y Apellidos  
Proyecto Identidad, Inmigración y Extranjería de Cuba.

**Ilustración 32 Acta de liberación del subsistema SPDI. (41)**