

**Universidad de las Ciencias Informáticas**



**Desarrollo del módulo Chequeo del Sistema de Control Migratorio en la  
frontera aérea de la Dirección de Inmigración y Extranjería**

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

**Autor(es):**

Yuleydis Pulido Delgado

Adriel Conde Traba

**Tutor(es):**

Ing. Katuska Cuba García

Ing. Yanisleivi Valdés Fernández

**Ciudad de la Habana.**

Junio del 2010

El único autógrafo digno de un hombre es el que deja escrito con sus obras.

José Martí

## DECLARACIÓN DE AUTORÍA

Declaramos que somos autores de este trabajo y autorizamos a la Universidad de la Ciencias Informáticas y a la Dirección de Informática y Comunicaciones del Ministerio del Interior a hacer uso del mismo en su beneficio.

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

---

Yuleydis Pulido Delgado  
Firma del Autor

---

Adriel Conde Traba  
Firma del Autor

---

Ing. Katuska Cuba García  
Firma del Tutor

---

Ing. Yanisleivi Valdés Fernández  
Firma del Tutor

## DATOS DE CONTACTO

### **Ing. Katuska Cuba García:**

- Ingeniera en Ciencias Informáticas, Universidad de las Ciencias Informática 2009.
- Vinculada a diferentes proyectos productivos en la UCI.
- Trabajadora civil de la Dirección de Informática y Comunicaciones, MININT.
- Correo electrónico: [kcuba@uci.cu](mailto:kcuba@uci.cu)

### **Ing. Yanisleivi Valdés Fernández:**

- Ingeniera en Ciencias Informáticas. Universidad de las Ciencias Informática 2008
- Profesora del Departamento de Sistemas Digitales. Facultad 1, Universidad de las Ciencias Informáticas.
- Vinculada a diferentes proyectos productivos dentro de la misma.
- Correo electrónico: [yvfernandez@uci.cu](mailto:yvfernandez@uci.cu)

## AGRADECIMIENTOS

## DEDICATORIA

...A la Revolución Cubana y a su obra creadora, a Fidel y Raúl y en especial a nuestros padres y hermanos.

## RESUMEN

La Dirección de Inmigración y Extranjería del Ministerio del Interior de la República de Cuba, cuenta con un grupo de sistemas desarrollados durante los últimos años, en su mayoría obsoletos desde el punto de vista de los avances actuales y que no satisfacen la gama de servicios que debe prestar la dirección. Un ejemplo es el sistema Centinel 2.0 que se utiliza en los aeropuertos internacionales del país.

El presente trabajo de diploma tiene como objetivo el desarrollo del módulo Chequeo que perfecciona el chequeo migratorio en la frontera aérea de la Dirección de Inmigración y Extranjería; permitiendo aumentar su seguridad, y lograr un mayor nivel de información sobre las personas que viajan por Cuba. Para la realización del módulo se llevó a cabo un estudio detallado de algunos sistemas dedicados a este proceso como son: Entrada/Salida-Tire por Circulado, Sistema de Migración y Extranjería de Venezuela y más específicamente el Centinel 2.0.

El módulo Chequeo está diseñado mediante la metodología *MSF for CMMI*, utilizando la herramienta de modelado Altova UModel 2009, el entorno de programación Visual Studio Team System 2008 y el lenguaje de programación C#. Comparte recursos con el gestor de base datos Oracle 11g y se ejecuta sobre un Servidor Windows Server 2003. Esto les permite a los usuarios, desde un sistema operativo de Windows XP o superior, el chequeo de los pasajeros mediante una búsqueda de información en la base de datos central nacional y una mayor eficiencia en el despacho migratorio.

## PALABRAS CLAVES

Centinel 2.0, frontera aérea, chequeo.

## ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....</b>	<b>6</b>
1.1. INTRODUCCIÓN .....	6
1.2. CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA .....	6
1.3. DESCRIPCIÓN DEL PROCESO .....	6
1.4. ESTADO DEL ARTE .....	7
1.5. PROPUESTA DE SOLUCIÓN .....	10
1.6. TECNOLOGÍAS A UTILIZAR .....	10
1.6.1. Metodología de Desarrollo de Software .....	11
1.6.1.1. Microsoft Solution Framework (MSF) .....	11
1.6.1.2. MSF for CMMI Process Improvement .....	12
1.6.2. Lenguaje de Modelado orientados a la representación de sistemas de negocios .....	13
1.6.2.1. UML .....	13
1.6.2.2. BPMN .....	14
1.6.3. Herramienta de Modelado .....	14
1.6.3.1. Altova-UModel 2009 .....	15
1.6.4. Aplicaciones web .....	15
1.6.5. Plataforma de Desarrollo .....	16
1.6.5.1. Microsoft .Net .....	16
1.6.6. Visual Studio 2008 (VS2008) .....	16
1.6.7. Visual Studio Team System 2008 .....	17
1.6.8. Framework .NET 3.5 .....	17
1.6.9. Lenguaje de Programación C# .....	18
1.6.10. Flujo de trabajo .....	18
1.6.10.1. Tipos de flujos de trabajo .....	19
1.6.11. Windows Workflow Foundation .....	19
1.6.12. Gestor de Base de Datos .....	20
1.6.12.1. Oracle 11g .....	20
1.6.13. ADO.NET Entity Framework .....	21
1.6.14. Embarcadero ER/Studio 8.0 .....	22
1.6.15. Bison Framework .....	22
1.6.15.1. Actividades .....	23
1.6.15.2. Services .....	23
1.6.15.3. Hosting .....	24
1.7. CONSIDERACIONES .....	24
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA .....</b>	<b>25</b>
2.1. INTRODUCCIÓN .....	25
2.2. ANÁLISIS CRÍTICO DE LA EJECUCIÓN DE LOS PROCESOS .....	25
2.3. MODELADO DE LOS PROCESOS DE NEGOCIO .....	26

2.3.1.	<i>Definición de los actores del negocio</i> .....	26
2.3.2.	<i>Definición de los trabajadores del negocio</i> .....	26
2.3.3.	<i>Descripción de los procesos</i> .....	27
2.3.3.1.	<i>Descripción del proceso Entrada/Salida de Pasajeros</i> .....	27
2.3.4.	<i>Reglas del negocio</i> .....	29
2.3.4.1.	<i>Tipos de Reglas</i> .....	29
2.3.4.2.	<i>Relación de las reglas del negocio</i> .....	30
2.4.	DESCRIPCIÓN DEL SISTEMA.....	30
2.5.	DESCRIPCIÓN DE LOS MÓDULOS DEL SISTEMA.....	31
2.6.	DESCRIPCIÓN DE LOS ROLES.....	32
2.7.	DIAGRAMAS DEL PROCESO MEJORADO CHEQUEO DE PASAJEROS Y TRIPULANTES.....	33
2.8.	ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE .....	34
2.8.1.	<i>Definición de los requisitos funcionales</i> .....	34
2.8.2.	<i>Descripción de requisitos funcionales</i> .....	34
2.9.	DEFINICIÓN DE LOS REQUISITOS NO FUNCIONALES.....	41
2.10.	CONCLUSIONES.....	44
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA .....</b>		<b>45</b>
2.11.	INTRODUCCIÓN .....	45
2.12.	ARQUITECTURA DE LA SOLUCIÓN .....	45
3.2.1.	<i>Descripción de las capas</i> .....	45
3.3.	<i>Patrones de diseño</i> .....	48
3.4.1.	<i>Ejemplos de Patrones</i> .....	48
3.4.2.	<i>Patrones de Flujo de trabajo</i> .....	49
3.4.3.	<i>Estándares de codificación y tratamiento de errores</i> .....	49
2.13.	CLASES UTILIZADAS .....	52
3.5.1.	<i>Clases Entidades</i> .....	53
3.5.2.	<i>Diagrama de clases del diseño</i> .....	55
2.14.	SERVICIOS DEL SISTEMA.....	55
3.6.1.	<i>Descripción de los Servicios</i> .....	56
2.15.	DISEÑO DE LOS FLUJOS DE TRABAJO.....	57
3.6.1.	<i>Descripción del flujo de trabajo Chequeo</i> .....	58
2.16.	MODELO DE DATOS .....	60
3.8.1.	<i>Descripción de las entidades fundamentales</i> .....	61
2.17.	CONCLUSIONES.....	62
<b>CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA .....</b>		<b>63</b>
3.1.	INTRODUCCIÓN .....	63
3.2.	DIAGRAMA DE DESPLIEGUE.....	63
3.3.	DIAGRAMAS DE COMPONENTES .....	64
3.4.	INTERFACES DEL MÓDULO CHEQUEO.....	65
3.5.	PRUEBAS.....	65
4.6.1.	<i>Diseño de casos de prueba</i> .....	65

4.6.2. Pruebas Unitarias.....	70
4.6.3. Pruebas de sistema.....	72
3.6. CONCLUSIONES.....	74
<b>CONCLUSIONES .....</b>	<b>75</b>
<b>RECOMENDACIONES.....</b>	<b>76</b>
<b>BIBLIOGRAFÍA.....</b>	<b>77</b>
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>80</b>
<b>ANEXOS.....</b>	<b>¡ERROR! MARCADOR NO DEFINIDO.</b>

## ÍNDICE DE TABLAS

TABLA 1. ACTORES DEL NEGOCIO .....	26
TABLA 2. TRABAJADORES DEL NEGOCIO .....	27
TABLA 3. PROCESO ENTRADA/SALIDA DE PASAJEROS.....	28
TABLA 4. RELACIÓN DE REGLAS DEL NEGOCIO .....	30
TABLA 5. DESCRIPCIÓN DE LOS ROLES EN EL MÓDULO CHEQUEO .....	32
TABLA 6. DESCRIPCIÓN DEL RF1 . CHEQUEAR PASAJEROS Y TRIPULANTES EN LA ENTRADA O SALIDA DEL PAÍS .....	41
TABLA 7. DESCRIPCIÓN DE LA ENTIDAD PROCESO MIGRATORIO .....	62
TABLA 8. CASO DE PRUEBA DEL REQUISITO CHEQUEAR PASAJEROS Y TRIPULANTES EN LA ENTRADA O SALIDA DEL PAÍS. 70	

## ÍNDICE DE FIGURAS

FIGURA 1. VISTA GLOBAL DEL PROCESO DE ENTRADA Y SALIDA DE PASAJEROS Y TRIPULANTES .....	25
FIGURA 3. CHEQUEO DE PASAJEROS Y TRIPULANTES. ....	33
FIGURA 4. VISTA GLOBAL DE LA ARQUITECTURA DE SOFTWARE. ....	45
FIGURA 10. ENTIDAD PROCESO MIGRATORIO .....	53
FIGURA 11. ENTIDAD EXPEDIENTE DE PASAJERO .....	54
FIGURA 12. ENTIDADES USADAS PARA ASEGURAR EL FUNCIONAMIENTO DEL MÓDULO CHEQUEO .....	54
FIGURA 13. DIAGRAMA DE CLASES DEL DISEÑO DEL MÓDULO CHEQUEO .....	55
FIGURA 14. SERVICIO INFORMATIONCAPTURERSERVICE. ....	56
FIGURA 16. DISEÑO DEL FLUJO DE TRABAJO CHEQUEO (WFCHEQUEO1).....	60
FIGURA 17. MODELO DE DATOS .....	61
FIGURA 18. DIAGRAMA DE DESPLIEGUE DEL MÓDULO CHEQUEO .....	63
FIGURA 19. DIAGRAMA COMPONENTES DEL MÓDULO CHEQUEO .....	64
FIGURA 20. RESULTADO DE LA PRUEBA UNITARIA AL MÉTODO VISAFREEAGREEMNET. ....	72

## INTRODUCCIÓN

La Dirección de Inmigración y Extranjería (DIE, por sus siglas en español) del Ministerio del Interior (MININT, por sus siglas en español) de la República de Cuba, es el órgano responsable de aplicar y controlar las normativas legales relacionadas con la migración, la extranjería y la ciudadanía en Cuba. Durante el último período del 2008 e inicios del 2009 se realizó un diagnóstico del estado actual de las distintas áreas en función de proponer la modernización de los procesos y tecnologías para facilitar la ejecución de los diversos trámites correspondientes a su objeto de trabajo, así como la informatización de diferentes áreas que soportan el correcto funcionamiento de la Dirección.

La DIE cuenta con un grupo de sistemas desarrollados durante los últimos años, en su mayoría obsoletos desde el punto de vista de los avances actuales, o que no satisfacen toda la gama de servicios que debe prestar la Dirección.

El Sistema de Control Migratorio Centinel 2.0 se implantó en una de las áreas de la Dirección, con el objetivo de integrar todo el flujo de información de los aeropuertos internacionales, en una base de datos utilizando el gestor de base de datos Oracle 9i, que recoge la información de los vuelos y sus respectivos pasajeros, las incidencias detectadas durante la entrevista del despacho migratorio, los datos de las personas o documentos circulados y la información de auditoría del sistema.

En Cuba existen 10 aeropuertos internacionales que trabajan con el sistema Centinel 2.0, ellos son:

<b>Provincias</b>	<b>Aeropuertos Internacionales.</b>
Ciudad Habana	José Martí
Matanzas	Juan Gualberto Gómez
Cienfuegos	Jaime González
Villa Clara	Abel Santamaría
Ciego de Ávila	Jardines del Rey
Camagüey	Ignacio Agramonte
Holguín	Frank País
Granma	Sierra Maestra
Santiago de Cuba	Antonio Maceo
Municipio Especial Isla de la Juventud	Vilo Acuña (Cayo Largo)

El Centinel 2.0 sustituyó al Sistema Entrada/Salida-Tire por Circulados que anteriormente funcionaba en el país y que carecía de muchas funcionalidades que necesitaba la frontera aérea en aquel tiempo. Con la puesta en funcionamiento del Centinel 2.0 se logró el uso de una interfaz gráfica que permitió que las pantallas de datos y reportes contaran con un mayor nivel informativo a diferencia de las anteriores interfaces que estaban basadas solo en texto. Proporcionó el uso de la fotografía y las facilidades que pueden brindar a través de los distintos dispositivos de captura de imagen como cámaras digitales y lectores de pasaportes. Asimismo logró que el acceso a la información se realizara a través del login, es decir cada persona que deseaba acceder a la aplicación debía tener un usuario y una contraseña, permitiendo a los usuarios acceder solo a la información que les está permitida manipular. Además, brindó la posibilidad al administrador del sistema de insertar circulaciones de documentos de viaje.

Actualmente el sistema Centinel 2.0 se mantiene en explotación en todos los puntos fronterizos aéreos del país, pero a raíz de la modernización tecnológica en la que se encuentra enfrascado el MININT, es necesario sustituirlo por un nuevo sistema que incluya las nuevas funcionalidades y permita su integración con el Sistema de Inmigración, Extranjería y Ciudadanía (Siec, por sus siglas en español), de la República de Cuba.

A pesar de la aceptación que ha tenido el sistema Centinel 2.0 y la eficiencia que ha aportado al trabajo que se realiza en los aeropuertos internacionales, actualmente no satisface todas las funcionalidades que necesita la frontera aérea para poder realizar su labor diaria y garantizar la seguridad de la misma.

Entre las funcionalidades que no posee el sistema están:

1. No posee una base de datos única que contenga toda la información de inmigración y extranjería que se recoge en todos los aeropuertos internacionales del país, imposibilitando el trabajo operativo y de enfrentamiento que necesita la frontera.

2. Falta de integración con sistemas de otras entidades como DIE, ECASA<sup>1</sup>, Tráfico aéreo y Operaciones propiciando que exista:

- Menos automatización de los procesos de chequeo migratorio.
- Cierre manual de los vuelos.
- No hay homogeneidad en la información de los vuelos.

---

<sup>1</sup> Empresa Cubana de Aeropuertos y Servicios Aeronáuticos

3. No posee todos los reportes que se requieren actualmente en la frontera aérea.
4. No permite la asociación de los pasajeros menores de edad al mayor responsable, lo que impide el control exhaustivo de los mismos.
5. A pesar de permitir la lectura del chip electrónico, no se comprueba la autenticidad del documento al no poder verificar el chip con las llaves públicas que emiten los países que lo utilizan.
6. No posee un módulo de Despacho Diferido que proporcione un análisis profundo del pasajero con problema.
7. El completamiento de la información no se realiza en tiempo real, demorándose hasta 24 horas después del paso por la frontera .
8. No está implementada la comparación de fotografías.

Teniendo en cuenta la situación antes señalada, se formula el siguiente **problema científico**: ¿Cómo perfeccionar el chequeo migratorio en la frontera aérea de la Dirección de Inmigración y Extranjería del MININT?

Hacia la solución de este problema se encamina esta tesis, al poseer como **objeto de estudio**: Los procesos de control migratorio en la frontera aérea de la Dirección de Inmigración y Extranjería del Ministerio del Interior, y como **campo de acción**: El chequeo migratorio en la frontera aérea de la Dirección de Inmigración y Extranjería del MININT.

En correspondencia con el problema planteado se elabora el siguiente **objetivo**: Desarrollar un módulo que perfeccione el chequeo migratorio en la frontera aérea de la Dirección de Inmigración y Extranjería del MININT.

**Objetivos específicos:**

- Analizar el marco teórico de la investigación.
- Modelar los procesos involucrados en el chequeo migratorio en la frontera aérea.
- Implementar el módulo Chequeo.
- Realizar pruebas unitarias y de sistema al módulo desarrollado.

Para dar solución a la problemática se plantea la siguiente **hipótesis**: El desarrollo del módulo Chequeo, permitirá perfeccionar el chequeo migratorio en la frontera aérea de la Dirección de Inmigración y Extranjería del MININT.

Dada la hipótesis planteada anteriormente se pueden definir como variables de la investigación:

- Variable independiente: Desarrollo del módulo Chequeo.
- Variable dependiente: Perfeccionar el chequeo migratorio en la frontera aérea de la Dirección de Inmigración y Extranjería del MININT.

Para poder cumplir con los objetivos se proponen las siguientes **tareas de la investigación**:

- Identificación de los procesos de negocio y los problemas que presentan.
- Estudio de sistemas similares para el control migratorio en la frontera aérea.
- Descripción de las herramientas, tecnologías y la metodología definida por el proyecto de Identificación, Inmigración y Extranjería.
- Modelación de los procesos de negocios.
- Definición de los requisitos funcionales y no funcionales.
- Confección del diseño del módulo Chequeo.
- Implementación del módulo Chequeo.
- Diseño de los casos de prueba que serán aplicados al módulo Chequeo.
- Documentación de las bases del sistema.

Los **métodos científicos** utilizados en la investigación fueron:

**Métodos teóricos:** Hipotético-Deductivo, Método de la modelación.

- Hipotético-Deductivo: A partir de la hipótesis y siguiendo la lógica de deducción tomada por el equipo de desarrollo se llega a nuevos conocimientos y predicciones, que son sometidos a verificaciones.
- Método de la modelación: Se lleva a cabo una modelación del objeto, pues la modelación de un proceso permite predecir la respuesta de dicho proceso a variaciones de algunos de sus parámetros, sin tener que ejecutar el proceso en la realidad. Para ello se decidió hacer una modelación del sistema usando la metodología *Microsoft Solution Framework*.

**Métodos empíricos:** Describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional. Estos métodos son empleados durante todo el proceso de implementación.

- Observación: Permite analizar cada fase del proceso e ir observando cada tarea que se realice y

tomar experiencia de esta para aplicarla en todas las demás.

➤ **Entrevista:** Esta investigación utiliza la estrategia descriptiva, ya que su objetivo es describir el fenómeno y reflejar lo esencial y más significativo del mismo, sin tener en cuenta las causas que lo originan, para lo que es necesario captar sus relaciones internas y regularidades.

El trabajo de diploma que se presenta está estructurado en 4 capítulos que se describen brevemente a continuación:

➤ **Capítulo 1 Fundamentación Teórica:** Este capítulo es el respaldo teórico de los temas tratados en el informe, necesarios para el entendimiento correcto de la solución que se propone. Se describen los conceptos fundamentales asociados al dominio del problema y el objeto de estudio, haciéndose un análisis de la situación actual y de distintos sistemas que realizan el mismo proceso. Además contiene la fundamentación de las tecnologías utilizadas para el diseño del sistema definidas por el proyecto de Identificación, Inmigración y Extranjería.

➤ **Capítulo 2 Características del sistema:** Este capítulo presenta una descripción general de las actividades del flujo actual de procesos mediante el modelo de negocio y un análisis crítico de estos procesos con el fin de proponer mejoras potenciales. Además, se levantan los requerimientos funcionales y no funcionales con los que debe cumplir el módulo propuesto, así como sus actores y diagramas de actividades del proceso mejorado de desarrollo con la descripción expandida de cada uno de ellos.

➤ **Capítulo 3 Análisis y diseño del sistema:** En este capítulo se muestran los diagramas de clases del análisis y del diseño mediante los flujos de trabajo, utilizando los componentes que brinda el Bison Framework<sup>2</sup>. Se describe el diseño de la base de datos, las características de las entidades y los servicios que sustentan las funcionalidades del sistema y otras restricciones del diseño.

➤ **Capítulo 4 Implementación y Prueba:** En este capítulo se generan los modelos de despliegue y de componente y se detallan las pruebas que se le realizan al módulo, especificando los casos de pruebas y las iteraciones realizadas al mismo.

---

<sup>2</sup> Marco de trabajo desarrollado por el Proyecto de Identificación, Inmigración y Extranjería de la República de Cuba orquestando los procesos de negocio con Windows Workflow Foundation.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### 1.1. Introducción

El presente capítulo brinda una visión general de los aspectos relacionados con el proceso de control migratorio en la frontera aérea, específicamente los relacionados con el chequeo migratorio de los pasajeros y tripulantes. También se describen los principales conceptos asociados al dominio del problema y la metodología, herramientas y tecnologías definidas por el proyecto para la realización del módulo Chequeo.

### 1.2. Conceptos asociados al dominio del problema

A continuación se muestran una serie de conceptos que están asociados al dominio del problema:

**Frontera:** Puntos legalmente establecidos para el arribo al Territorio Nacional de las naves aéreas y marítimas procedentes del exterior, en los que se ejecuta el control documental y físico, comúnmente llamado despacho o chequeo migratorio, que realizan los Inspectores de Inmigración y Extranjería para determinar si los pasajeros y tripulantes cumplen con las disposiciones establecidas en la Ley de Migración y su Reglamento, para entrar o salir del país.

**Control migratorio:** Es el control policíaco tanto de personas como de aeronaves en la entrada, permanencia y salida del Territorio Nacional. Comprende el despacho migratorio, el control de la aeronave y el régimen legal de permanencia.

**Control migratorio en la frontera:** Es el conjunto de medidas que se toman en el punto de frontera para garantizar el examen, comprobación y validación del movimiento de entrada y salida de naves, aeronaves, pasajeros y tripulantes.

**Control migratorio en la frontera aérea:** Es el proceso que se lleva a cabo para controlar todas las aeronaves, pasajeros y tripulantes que se encuentren entrando o saliendo de la frontera aérea cubana.

### 1.3. Descripción del proceso

El Centinel 2.0 es el sistema que actualmente se utiliza para llevar a cabo el control migratorio en la frontera aérea cubana. Brinda ciertas funcionalidades que se realizan en los aeropuertos internacionales, para llevar a cabo el despacho migratorio de todos los pasajeros y tripulantes de un avión. Lo primero que

se realiza cuando el viajero se presenta en la taquilla, es realizarle el físico<sup>3</sup>, capturar la foto mediante una cámara web y los datos del pasaporte, de forma manual o mediante la lectura mecánica del mismo. Después con estos datos se procede a la búsqueda en una base de datos con información de personas, documentos de viajes y circulaciones, con el objetivo de constatar que el pasajero no tenga algún impedimento a la hora de realizar la entrada o salida del país. Una vez capturada toda la información necesaria para el chequeo del pasajero, se pasa a comprobar o incluir incidencias detectadas durante la entrevista y se verifica la autenticidad del pasaporte. Si el chequeo del pasajero no presentó ningún problema, entonces la información capturada pasa a ser registrada en el sistema. Si por el contrario, el chequeo arroja algún impedimento, se analiza nuevamente a la persona, consultando el hecho con el Jefe de Grupo.

## 1.4. Estado del Arte

En el mundo existe un gran número de sistemas dedicados al control migratorio en las fronteras aéreas, marítimas y terrestres. La mayoría, por ser sistemas gubernamentales, no publican su información técnica pues están regidos por las leyes migratorias de los distintos países que lo han implementado y solo existe información publicitaria.

Por esta razón se analizaron solo 4 sistemas dedicados al control migratorio, que ayudaron a confeccionar la propuesta de solución y ver con claridad la necesidad de implementar un nuevo sistema que incluya las funcionalidades que necesita la frontera cubana. Estos sistemas son: Entrada/Salida-Tire por Circulado de Cuba, Sistema Centinel 2.0 de Cuba, el Sistema Integral de Operación Migratoria de México y el Sistema Administrativo de Identificación, Migración y Extranjería de la República Bolivariana de Venezuela.

### **Sistema Entrada/Salida-Tire por Circulado.**

Este sistema es obsoleto en relación con el desarrollo actual porque está basado en tecnologías de programación muy atrasadas, provocando que no se le puedan agregar nuevos servicios. Existen dos sistemas básicos: el Tire por Circulados que se encontraba instalado en los aeropuertos, soportado sobre el lenguaje de programación FoxPlus en una red Novel, y el sistema de Entrada/Salida que se ejecutaba sobre un servidor de aplicaciones Unix desarrollado con en el mismo lenguaje. El primero se empleaba

---

<sup>3</sup> Acto de comprobación de forma visual, del rostro del pasajero con la foto del pasaporte.

para buscar a los viajeros en una base de datos de circulaciones, mientras el segundo se utilizaba para procesar la información de los pasajeros y volver a buscarlos por la mencionada base de datos de circulados. Ninguno estaba soportado sobre interfaces gráficas imposibilitando así el aprovechamiento de tecnologías de avanzada. Ambos solo simulaban sistemas multiusuario, pero en realidad tenían la problemática de que varios usuarios no podían incluir o actualizar información a la vez sobre la misma base de datos siendo esta una de las causas fundamentales de disímiles dificultades.

## **Sistema Centinel 2.0**

El Centinel 2.0 es un sistema automatizado para el control migratorio de la frontera que se utiliza actualmente en todos los aeropuertos internacionales de Cuba, cuya meta es aumentar la seguridad de la frontera a la vez que se agiliza el despacho migratorio, el procesamiento de datos y el traslado de la información de todos los pasajeros que entran o salen del territorio nacional. Éste unificó en un mismo sistema las principales funciones que realizan los oficiales de Inmigración en los puntos fronterizos, y para ello en su diseño se concibieron 4 módulos programados en el lenguaje de programación Visual Basic 6.0 y orquestado por el sistema de gestor de base de datos Oracle 9i en el cual se almacena toda la información. Estos módulos son: Chequeo, Procesamiento y control, Administración y Monitoreo. Nuestra investigación se centra en el estudio de los módulos que intervienen de alguna forma en el aseguramiento del chequeo de pasajeros y tripulantes:

➤ **Módulo de Chequeo:** Permite el chequeo de los pasajeros y tripulantes en la frontera mediante la búsqueda de sus datos en una base de datos con información de personas y documentos de viajes circulados, capturar la fotografía del mismo, comprobar o incluir las incidencias detectadas durante la entrevista con el pasajero, verificar la autenticidad del pasaporte y verificar el físico de la persona con fotos tomadas con anterioridad.

➤ **Módulo de Procesamiento y control:** Es utilizado para procesar la información de todos los vuelos con sus respectivos pasajeros, el control de su calidad, hacerle un reanálisis de todos los datos identificativos con la base de datos de circulaciones y la edición de las fotografías capturadas durante el despacho migratorio.

## **Sistema Integral de Operación Migratoria de México:**

Este programa permite conectar electrónicamente los principales puertos, aeropuertos internacionales y puntos fronterizos de México para facilitar a las autoridades migratorias el registro de los ingresos y

salidas de personas.

Constituye una red de computadoras que está conectada a una base de datos central que se nutre con informaciones judiciales, policíacas, líneas navieras y aéreas de instituciones del gobierno.

Cada punto cuenta con terminales que poseen escáneres con rayos infrarrojos y ultravioletas donde se graba la fotografía, datos del pasaporte y permisos migratorios de las personas que entran o salen del país.

Con estas tecnologías se hace posible la identificación de documentos falsos y la ágil actualización y administración de los listados de personas con antecedentes criminales o terroristas, así con el uso de información inteligente se protege al país y la seguridad nacional.

El sistema reduce los tiempos de espera en garitas, aeropuertos y puestos fronterizos.

Es también un mecanismo fundamental para ordenar los flujos migratorios, reducir los tiempos de espera al ingresar al país y agilizar de esta manera los trámites de carácter migratorio. (Saldaña, 2004).

**Servicio Administrativo de Identificación, Migración y Extranjería (Saime, por sus siglas en español):**

El Saime es un sistema de migración y control de prohibiciones que fue elaborado para garantizar el control automatizado de los movimientos migratorios en la República Bolivariana de Venezuela, dando la posibilidad de chequear en cada caso la existencia de prohibiciones, así como otros múltiples problemas que puede presentar un ciudadano al entrar o salir del país. Y garantiza la gestión de las prohibiciones de entradas y salidas.

Este sistema posee el módulo de Oficina Migración, que fue el que se analizó con más detenimiento. Las funcionalidades que brinda este módulo son:

- Permite realizar el control migratorio, tanto para entradas como para salidas del país, integrando la lectura mecánica con equipamiento especializado.
- Realiza el chequeo de las prohibiciones, antecedentes u otros problemas que pueda presentar el ciudadano durante el trámite de migración.
- Almacena los datos personales de los ciudadanos, incluyendo su fotografía, con posibilidades de ajuste de imagen.
- Permite un chequeo avanzado del pasaporte, mostrando imagen ultravioleta e infrarroja.

- Posibilita realizar el completamiento de los datos pertenecientes a la tarjeta de migración.

## 1.5. Propuesta de solución

La propuesta de solución que se presenta está basada en el funcionamiento de la mayoría de los procesos del sistema Centinel 2.0, por ser el que más se ajusta a las leyes migratorias cubanas y estar probado durante varios años en nuestro país. Será implementado sobre una plataforma única lo cual hará corresponder simultáneamente todos los servicios que brindarán los subsistemas Identidad, Enfrentamiento y Circulado, del Sistema de Inmigración, Extranjería y Ciudadanía de Cuba. Además, contará con un repositorio orquestado por el Gestor de base de datos Oracle11g que proporcionará una mayor rapidez y seguridad a la hora de brindar la información que soliciten todos los sistemas.

### **Las mejoras que se propone lograr con el desarrollo del módulo Chequeo son:**

- Proporcionar un despacho migratorio más seguro, ya que el Módulo de Procesamiento y Control de Datos se va a fusionar con el chequeo actual posibilitando que se almacene en tiempo real toda la información que se recoge en los aeropuertos internacionales del país.
- Asegurar la integridad y confidencialidad de la información que se maneja al darle la mínima información posible a los inspectores de chequeo, que en esta nueva versión no tienen acceso a las circulaciones, ni problemas que puedan presentar los pasajeros y tripulantes y tampoco tiene el nivel de decisión suficiente para darle solución a estos problemas.
- Al contar con una base de datos única a nivel nacional, se tendrá acceso a la misma desde cualquier punto de control migratorio del país.
- Proveer la lectura del chip electrónico de los pasaportes, permitiendo su verificación, mediante la utilización de las llaves públicas que se emiten internacionalmente.
- Brindar la posibilidad de acceder a las réplicas de la base de datos central que estarán en cada aeropuerto internacional, permitiendo una mayor rapidez en el trabajo.
- Realizar un control más detallado con los menores, determinando quienes son los responsables del mismo mediante la asociación de éste con el mayor, posibilitando así el posterior control migratorio de estos pasajeros y de los mayores responsables.

## 1.6. Tecnologías a utilizar

A continuación se caracterizan las metodologías, tecnologías y herramientas utilizadas en el desarrollo

del módulo propuesto, las cuales fueron definidas por las políticas de desarrollo del proyecto de Identificación, Inmigración y Extranjería de la República de Cuba.

## 1.6.1. Metodología de Desarrollo de Software

Para desarrollar un software es necesario guiar el proceso a través de una metodología, la cual será la encargada de elaborar “el plano” sobre el que se apoyará el equipo de desarrollo, permitiendo construir un software de alta calidad, en el tiempo y con el costo esperado. Proporciona un sistema flexible ante los cambios que solicite el usuario, ya sea cuando el sistema esté en desarrollo o una vez terminado, contribuyendo a que no se produzcan atrasos en los proyectos por cuestiones organizativas, de entendimiento y de comunicación entre los miembros del equipo de desarrollo, eliminando así la incomodidad de los clientes y desarrolladores ante los cambios que puedan existir.

### 1.6.1.1. Microsoft Solution Framework (MSF, por sus siglas en inglés)

*Microsoft Solution Framework* o Marco de Soluciones para Microsoft constituye un marco para el desarrollo de sistemas de software basado en principios, modelos, disciplinas, conceptos, prácticas y recomendaciones propias derivadas de la experiencia de Microsoft. Se autodefine como “marco” y no como metodología porque considera que no hay una única estructura de procesos válida para todos los proyectos. (Turner, 2006)

MSF propone una secuencia generalizada de actividades para la construcción de soluciones empresariales. Este proceso es flexible y se puede adaptar al diseño y desarrollo de una amplia gama de proyectos de una empresa. Además, está basado en fases, puntos de transición y de carga de forma iterativa que se puede aplicar en el desarrollo de aplicaciones tradicionales, soluciones empresariales para comercio electrónico así como aplicaciones web distribuidas. (Keeton, 2006)

#### **Características de MSF:**

**Adaptable:** es usado en cualquier parte y no se limita a un equipo o proyecto específico.

**Escalable:** puede organizar equipos pequeños entre 3 ó 4 personas, así como proyectos que requieren 50 personas o más.

**Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.

**Tecnología Agnóstica:** porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

## 1.6.1.2. MSF for CMMI<sup>4</sup> Process Improvement

Con la aparición del producto *Microsoft Visual Studio Team System* se ha actualizado MSF a la versión 4.0, produciendo dos variantes: *MSF for Agile Software Development* para el trabajo en entornos que emplean metodologías ágiles, y *MSF for CMMI Process Improvement* para el trabajo en entornos con el modelo CMMI.

*MSF for CMMI* es un proceso ágil de desarrollo de software, el cual cumple con los requerimientos para el nivel 3 de CMMI. La principal diferencia entre *MSF Agile* y *MSF for CMMI* es que este último está orientado a proyectos donde el nivel de formalidad es mucho mayor y se exige una cultura de mejoramiento continuo de procesos. Uno de los beneficios de implementar la metodología de *MSF for CMMI* es contar con una evaluación estándar por medio de la cual se puede validar la habilidad de desarrollar software en una organización. (26)

Dentro de los aspectos más interesantes acerca de esta metodología está el hecho de que ya viene integrada a la plataforma de desarrollo de *Microsoft Visual Studio Team System*. *Visual Studio Team System* (VSTS, por sus siglas en inglés) es la solución de Microsoft para gestionar el ciclo de vida completo de soluciones para la plataforma Windows. Por medio de VSTS es posible contar con una serie de plantillas y guías adaptadas a cada metodología y orientadas a los roles definidos en cada una. Además, permite reducir la complejidad del proceso de desarrollo, mejorar la colaboración de todos los miembros del equipo, mejorar la productividad reduciendo los tiempos de desarrollo y pruebas, y gestionar los elementos de trabajo.

Esta metodología propone los primeros 3 niveles de CMMI:

### ➤ Inicial

- Estado inicial donde el desarrollo se basa en la heroicidad y responsabilidad de los individuos.
- Los procedimientos son inexistentes o localizados en áreas concretas.
- No existen plantillas definidas a nivel corporativo.

### ➤ Gestionado

- Se normalizan las buenas prácticas en el desarrollo de proyectos (en base a la experiencia y al método).

---

<sup>4</sup> Capability Maturity Model Integration o Modelo Integrado de Madurez de las Capacidades.

- En este nivel consolidado, las buenas prácticas se mantienen en los momentos de estrés.
- Están definidos los productos a realizar.
- Se definen hitos para la revisión de los productos.
- **Definido**
  - La organización entera participa en el proceso eficiente de proyecto de software.
  - Se conocen de antemano los procesos de construcción de software.
  - Existen métodos y plantillas bien definidas y documentadas.
  - Los procesos no afectan solamente a los equipos de desarrollo, sino a toda la organización relacionada.
  - Los proyectos se pueden definir cualitativamente. (Keeton, 2006)

## 1.6.2. Lenguaje de Modelado orientados a la representación de sistemas de negocios

Una vez definida la metodología de desarrollo, el equipo de trabajo debe comenzar a modelar los procesos de negocio del área donde el sistema será desplegado. Las herramientas que facilitan esta actividad visualizan, especifican, construyen y documentan el sistema informático en progreso permitiendo la captura de información de larga vida que forma parte de los artefactos generados por el proyecto en sí.

Es importante tener en cuenta el lenguaje de modelado y la notación que se utilizan para llevar a cabo esta tarea.

### 1.6.2.1. UML<sup>5</sup>

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

---

<sup>5</sup> Unified Modeling Language o Lenguaje Unificado de Modelado por su significado en español.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar.

## 1.6.2.2. BPMN<sup>6</sup>

Es una notación gráfica estandarizada que permite el modelado de procesos de negocio en un formato de flujo de trabajo. Constituye un nuevo estándar de modelado de los asuntos de negocio, donde se presentan gráficamente las diferentes etapas del proceso. La notación ha sido diseñada específicamente para coordinar la secuencia de eventos y los mensajes que fluyen entre los diferentes procesos participantes. BPMN cubre casi totalmente los patrones de flujos de trabajo con los cuales se especifican los procesos.

Los objetivos de BPMN son:

- Proporcionar a los negocios la capacidad de entender sus procedimientos internos en una notación gráfica, facilitando a las organizaciones la habilidad para comunicarlos de una manera estándar.
- Proveer una notación que sea fácilmente entendida por todos los usuarios por ejemplo los analistas del negocio, el desarrollador técnico y la propia gente del negocio.
- Crear un puente estandarizado para el vacío existente entre el diseño del proceso de negocio y su implementación.
- Asegurar que los lenguajes para la ejecución de los procesos de negocio puedan ser visualizados con una notación común.
- Comunicar una amplia variedad de información a una amplia variedad de audiencias. (Pérez, 2009)

## 1.6.3. Herramienta Modelado

Las herramientas de modelado permiten incrementar la productividad y el control de la calidad en cualquier proceso de elaboración de software, ya que transforman la actividad de desarrollar software en un proceso automatizado. A medida que los sistemas que hoy se construyen se tornan más complejos, las herramientas de modelado con UML ofrecen muchos beneficios para todos los involucrados en un proyecto.

---

<sup>6</sup> Business Process Management Notation (BPMN por, sus siglas en inglés) o Notación para el Modelado de procesos de Negocio.

### 1.6.3.1. Altova-UModel 2009

Altova UModel 2009 se considera como el punto de partida para el desarrollo de software de éxito. Diseña visualmente modelos de aplicaciones en BPMN y UML, generando código Java, C#, o Visual Basic .NET y la documentación del proyecto. Realiza ingeniería inversa de los programas existentes pasándolos a diagramas UML 2.0. Esta herramienta hace el diseño visual de software práctico para cualquier proyecto, permitiendo de manera simple y asequible dibujar en UML. Combina una rica interfaz visual con funciones de usabilidad superiores para ayudar a nivelar la curva de aprendizaje de UML. Las características de UModel 2009 para el desarrollo de software basado en las capacidades de modelado avanzado son:

- Soporte para los 14 tipos de diagramas UML
- Diagramas de proceso de negocio (BPMN)
- Sincronizado de modelo y código a través de ingeniería de ida y vuelta
- Crea diagramas de secuencia desde el código fuente de la ingeniería inversa
- Generación de documentación personalizable de proyecto
- Compartir subproyectos para colaboración o reutilización
- Capas de diagramas con visibilidad selectiva
- Hipervínculos entre diagramas, documentos, o páginas web
- Integración con sistemas de control de versiones
- Estrecha integración con Visual Studio y Eclipse (Danysoft | Haciendo visible lo invisible, 2009)

### 1.6.4. Aplicaciones web

La meta fundamental de los desarrolladores de software en la actualidad es crear aplicaciones cada vez mejores y en el menor tiempo posible, por esta razón es imprescindible entender conceptos asociados a este tema. Se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. Es decir, se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. Las aplicaciones web son populares debido a la facilidad para actualizar y mantenerlas sin tener que distribuir e instalar software a miles de usuarios. Algunos ejemplos son los correos web, weblog o tiendas en línea. Una página web puede contener elementos que permiten la comunicación activa entre el usuario

y la información, accediendo a los datos de modo interactivo, como rellenar y enviar formularios, participar en juegos, etc. (Hooping.net, 2008)

## **1.6.5. Plataforma de Desarrollo**

Toda aplicación web debe poseer una plataforma que la sustente, entendiéndose como plataforma de desarrollo al entorno de software común en el cual se desenvuelve la programación de un grupo definido de aplicaciones. Comúnmente se encuentra relacionada directamente a un sistema operativo; sin embargo, también es posible encontrarla ligada a una familia de lenguajes de programación o a una interfaz de programación de aplicaciones.

### **1.6.5.1. Microsoft.Net**

La plataforma .NET no es un solo producto sino un conjunto de productos. Desde sistemas operativos como Windows XP, servidores de aplicaciones como SQL Server 2000, productos de oficina como Office XP, herramientas de desarrollo como Visual Studio .NET hasta servicios web provistos por Microsoft como Visual Studio.NET. Tanto la invocación de los servicios como su ejecución pueden ser hechas en cualquier dispositivo y sistema operativo, y accedido desde Internet. Los sitios se comunican entre sí y acceden a servicios y contenidos sin la intervención humana. Por eso se le llama la nueva generación de Internet "Internet inteligente". (DesarrolloWeb.com , 2004)

## **1.6.6. Visual Studio 2008 (VS2008, por sus siglas en inglés)**

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++, Visual C# y Visual J# utilizan el mismo entorno de desarrollo integrado (IDE, por sus siglas en inglés), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios web XML.

Precisamente uno de los primeros puntos a destacar de Visual Studio 2008 es que el producto reúne lo que hasta ahora eran diversos complementos, como Windows Communication Foundation (WCF, por sus siglas en inglés) y Windows Workflow Foundation (WF, por sus siglas en inglés), junto con la versión más reciente de la plataforma .NET Framework 3.5 runtime.

La tecnología Silverlight, integrada en VS2008, es una plataforma de desarrollo para la creación de aplicaciones web independientes del sistema operativo.

Entre otras muchas funcionalidades se encuentra la seguridad del código generado, Visual Studio 2008 Professional incluye ahora herramientas de pruebas unitarias, que previamente sólo estaban en las versiones de grupo de trabajo, Team Suite. Esto permite que los pequeños equipos de desarrollo o incluso los creadores individuales no necesiten herramientas adicionales para llevar a cabo pruebas unitarias. (MKM Publicaciones, 2008)

### 1.6.7. Visual Studio Team System 2008

*Visual Studio Team System* facilita la comunicación y la colaboración del equipo de desarrollo, proporcionando un repositorio unificado de todos los datos del proyecto, junto con las herramientas para definir, ejecutar y automatizar los procesos deseados. En su conjunto de herramientas integradas *Team Foundation Server 2008*, permite dar soporte al código fuente y al control de versiones, así como dar seguimiento a los elementos de trabajo, construir y automatizar los controles de calidad. Los artefactos, datos de proyectos, control de código fuente y las pruebas se guardan en un almacén de datos, mientras que los paneles e informes eficaces proporcionan datos de las tendencias de historial, capacidad de seguimiento total y visibilidad en tiempo real para la calidad y el progreso del objetivo de la empresa. (DanySoft, 2008)

### 1.6.8. Framework .NET 3.5

Este marco de trabajo agrega a las mejoras de desempeño, escalabilidad y seguridad con respecto a la versión anterior, estas novedades son:

➤ Las mejoras en las capacidades de pruebas unitarias permiten ejecutarlas más rápido independientemente de si lo hacen en el entorno IDE<sup>7</sup> o desde la línea de comandos. Se incluye además, un nuevo soporte para diagnosticar y optimizar el sistema a través de las herramientas de pruebas de Visual Studio. Con ellas se podrán ejecutar perfiles durante las pruebas para que ejecuten cargas, prueben procedimientos contra un sistema y registren su comportamiento; y utilizar herramientas integradas para depurar y optimizar.

➤ .NET 3.5 incluye biblioteca ASP.NET<sup>8</sup> AJAX para desarrollar aplicaciones web más eficientes, interactivas y altamente personalizadas que funcionen para los navegadores más populares y utilicen las últimas tecnologías y herramientas web. (ASP.NET, 2010)

---

<sup>7</sup> Integrated Development Environment (IDE, por sus siglas en inglés) o Ambiente de Desarrollo Integrado.

<sup>8</sup> Parte del .NET Framework dedicada al desarrollo web.

## 1.6.9. Lenguaje de Programación C#

El lenguaje de programación C# fue creado por el danés Anders Hejlsberg que diseñó también los lenguajes Turbo Pascal y Delphi. C# (pronunciado en inglés “C Sharp” o en español “C sostenido”) es un lenguaje de programación orientado a objetos. Con este nuevo lenguaje se quiso mejorar con respecto a los dos lenguajes anteriores de los que deriva el C, y el C++.

Aunque en la plataforma .NET es prácticamente posible programar en cualquier lenguaje, C# es el lenguaje de propósito general diseñado por Microsoft para ser utilizado en ella, por lo que programar usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes.

Entre sus principales características se destacan:

- C# provee el beneficio de un ambiente elegante y unificado.
- No soporta herencia múltiple, solamente el runtime<sup>9</sup> .NET permite la herencia múltiple en la forma de interfaces, las cuales no pueden contener implementación.
- El manejo de errores está basado en excepciones.
- Soporta los conceptos como encapsulación, herencia y polimorfismo de la programación orientada a objetos.
- No existen funciones globales, variables o constantes. Todo debe ser encapsulado dentro de la clase, como un miembro de la instancia (accesible vía una instancia de clase) o un miembro estático (vía el tipo).
- Los métodos que se definen en las clases son por defecto no virtuales (no pueden ser sobrescritos al derivar clases).
- Solamente se permite una clase base, si se requiere herencia múltiple es posible implementar interfaces.
- No es posible utilizar variables no inicializadas.
- No es posible hacer el casteo de un entero a un tipo de referencia. (Mossca.rog., 2010)

## 1.6.10. Flujo de trabajo

Es uno de los mecanismos usados por los negocios para expresar los procesos como series de

---

<sup>9</sup> De la definición en inglés de tiempo real

actividades autocontenidas. Los Administradores de Procesos de Negocio (BPM, por sus siglas en inglés, Business Process Management) proveen de un entorno a los desarrolladores para crear, ejecutar y manejar flujos de trabajo. Estos flujos de trabajo son generalmente expresados usando máquinas de estado finito, UML o diagramas de flujo.

Un flujo de trabajo es un conjunto de actividades guardadas como un modelo que describe un proceso del mundo real. El trabajo pasa a través del modelo desde el principio hasta el final, y las actividades pueden ser ejecutadas por personas o por funciones del sistema. El flujo de trabajo provee una forma de describir el orden de ejecución y la dependencia de las relaciones entre las piezas de corta o larga duración. Mientras que es posible escribir un flujo de trabajo completamente en código, éste en general es mejor visto gráficamente. Una vez que el modelo es compilado, puede ser ejecutado dentro de cualquier proceso de Windows, incluyendo aplicaciones de consola, servicios Windows y web, como también páginas ASP.NET.

#### 1.6.10.1. Tipos de flujos de trabajo

**Secuenciales:** el estilo de flujo de trabajo secuencial es directo, simple y útil para operaciones repetitivas o predecibles, tales como diseñar un conjunto de actividades para ser ejecutadas en una secuencia predecible que es siempre la misma. Esta arquitectura es similar a la del diseño de una aplicación basada en procedimientos.

**Máquina de estado:** consiste en un conjunto de estados manejados por eventos. El mismo flujo de trabajo está compuesto por un conjunto de estados. Un estado se indica como estado inicial. Cada estado puede recibir un cierto juego de eventos. Sobre la activación de uno de ellos puede cambiar de un estado a otro. Este flujo de trabajo puede tener también un estado final. Cuando una transición es realizada al estado final, el flujo de trabajo es finalizado. (Jose Luis Olivares, BS-Buffera Sistemas)

#### 1.6.11. Windows Workflow Foundation

*Microsoft Windows Workflow Foundation* es un marco de trabajo extensible para desarrollar soluciones de flujos de trabajo sobre la plataforma Windows. Provee una interfaz de programación de aplicaciones y herramientas para el desarrollo y la ejecución de aplicaciones basadas en flujos de trabajo. *Windows Workflow Foundation* provee un único modelo unificado para crear soluciones de punta a punta que abarcan categorías de aplicaciones, incluyendo flujos de trabajo tanto humanos como de sistemas.

Esta tecnología de WF complementa al Framework .NET con un grupo de componentes basados en

flujos de trabajo que brindan a los desarrolladores la habilidad de definirlos, compilarlos, instanciarlos, depurarlos y rastrearlos.

El soporte para los flujos de trabajo de prolongada duración es otra gran característica de *Microsoft Windows Workflow Foundation*. Estos flujos de trabajo se basan en persistencia, transacciones, tracking y procesamiento del Host para completar sus tareas. Cuando el flujo de trabajo está inactivo, esperando información de usuarios o sistemas, es automáticamente persistido y eliminado de la memoria para reducir consumo de recursos. (Jose Luis Olivares, BS-Buffera Sistemas)

## 1.6.12. Gestor de Base de Datos

Un Sistema gestor de base de datos (SGBD, por sus siglas en español) es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Así se trata de un software de propósito general. Ejemplo de SGBD son Oracle y SQL Server de Microsoft.

Algunas de las características deseables en un SGBD son:

- Control de la redundancia: La redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.
- Restricción de los accesos no autorizados: cada usuario ha de tener unos permisos de acceso y autorización.
- Cumplimiento de las restricciones de integridad: el SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad. (garbagecollector, 2004).

### 1.6.12.1. Oracle 11g

Oracle Database 11g proporciona funcionalidades que garantizan alto rendimiento, alta escalabilidad,

fiabilidad y seguridad mediante el uso de plataformas grid<sup>10</sup>, asegurando a la vez altos niveles de calidad de servicio e incrementos de la flexibilidad de negocio, reduciendo además, los costes de explotación. Se pueden resolver las problemáticas de negocio más exigentes en todas las áreas, incluyendo aplicaciones transaccionales, de inteligencia de negocio y de gestión de contenidos.

Oracle Database11g incluye funcionalidades que permiten hacer pruebas de cambios en aplicaciones simulando las cargas reales generadas por los usuarios en los entornos de producción. Aplicación de Pruebas Reales, permite reducir de manera drástica los tiempos, riesgos y costes derivados de la implantación de cambios, asegurando que las aplicaciones se comportarán de manera adecuada y predecible tras las modificaciones. Con la Aplicación de Pruebas Reales los clientes ganan en flexibilidad puesto que pueden responder de manera más efectiva a los requerimientos cambiantes del negocio y hacer una gestión del cambio más efectiva. (Oracle Corporation, 2009)

Oracle 11g tiene como características básicas:

- El soporte brinda la disponibilidad sobre varias plataformas (Windows, Linux, Unix).
- La fiabilidad posibilita la continua disponibilidad de las aplicaciones y los datos.
- Es escalable al presentar la condición que tienen los sistemas de adaptarse al crecimiento de trabajo de manera fluida y hacerse más grande sin perder calidad en los servicios ofrecidos.
- La seguridad y protección de datos permite compartir la red de recursos de una empresa con la confianza de que la privacidad se mantiene.
- La auto-gestión de Oracle automatiza muchas de las funciones de infraestructura de modo que un solo Administrador puede administrar cientos de servidores.

### 1.6.13. ADO.NET Entity Framework

*Entity Framework* es un conjunto de tecnologías de *ADO.NET* que facilitan el desarrollo de aplicaciones de software orientadas a datos. Permite a los programadores trabajar con datos en forma de objetos y propiedades específicos del dominio, por ejemplo, con clientes y direcciones, sin tener que pensar en las tablas de las bases de datos subyacentes y en las columnas en las que se almacenan estos datos. Para ello, se eleva el nivel de abstracción en la que los programadores pueden trabajar al tratar con datos y se

---

<sup>10</sup> Infraestructura que permite la integración y el uso colectivo de ordenadores de alto rendimiento, redes y bases de datos que son propiedad y están administrados por diferentes instituciones

reduce el código requerido para crear y mantener las aplicaciones orientadas a datos. Dado que *Entity Framework* es un componente de *.NET Framework*, las aplicaciones de *Entity Framework* se pueden ejecutar en cualquier equipo en que esté instalado *.NET Framework 3.5 Service Pack 1 (SP1)*, por sus siglas en inglés). A través de la infraestructura de servicios de objeto de *Entity Framework*, ADO.NET expone una vista conceptual común de los datos, incluidos los datos relacionales, como objetos del entorno .NET. Esto hace que la capa de objetos sea un objetivo ideal para la compatibilidad con *LINQ*<sup>11</sup> (Microsoft Corporation, 2006)

### 1.6.14. Embarcadero ER/Studio 8.0

Herramienta para el modelado de datos que permite descubrir, documentar, y reutilizar los activos en datos. *ER/Studio* permite una separación entre modelos lógicos y físicos de los datos, da soporte a SGBD heterogéneos, tiene motores y navegación para formateado de diagramas y presenta una arquitectura flexible.

*ER/Studio 8.0* también incluye nuevas funcionalidades de sub-modelado en el nivel de atributos y opciones de comparación, que ahorran tiempo. Algunas funcionalidades y beneficios del *ER/Studio 8.0* son:

- Separación de modelos físicos y lógicos con integración completa, transformación automatizada, y mapeo de tipos personalizados de datos.
- Publicación de modelos e informes en una gran variedad de formatos incluyendo HTML<sup>12</sup>, RTF<sup>13</sup> y, Microsoft Office Output.
- Generación de esquemas XML desde modelos físicos y lógicos.
- Ingeniería inversa y directa.
- Integración de modelos y metadatos.
- Diseños de Calidad de Bases de Datos, entre otras. (Danysoft, 2009)

### 1.6.15. Bison Framework

---

<sup>11</sup> Language-INtegrated Query(LINQ, por sus siglas en inglés)

<sup>12</sup> Lenguaje de Marcado de Hipertexto(HTML, por sus siglas en inglés)

<sup>13</sup> Rich Text Format (formato de texto enriquecido)

Es un marco de trabajo para la orquestación de procesos de negocio con Windows Workflow Foundation. Su principal objetivo es proporcionar un componente que permita gestionar las instancias de flujo de trabajo. Además, encapsula un conjunto de actividades y servicios que le dan mayor dinamismo al desarrollo de sistemas centrado en la orquestación de procesos de negocio con flujos de trabajo y son de vital importancia para la arquitectura.

Entre las ventajas que brinda el Bison Framework se pueden mencionar:

- Proporciona una mayor aproximación a los usuarios de negocio, así como más rapidez y flexibilidad para modelar y cambiar los procesos.
- Brinda flexibilidad para poder cambiar los procesos según las necesidades.
- Aporta escalabilidad o capacidad de crecer.
- Fortifica el puente creado por el flujo de trabajo para la comunicación entre el analista y el desarrollador.
- Propone una arquitectura donde se encuentran perfectamente definidas las capas de presentación y negocio.
- Posee actividades y servicios especializados en la orquestación de pantallas, que permiten definir el flujo y la pantalla que se desea mostrar de una manera gráfica dentro del flujo de trabajo.

### 1.6.15.1. Actividades

En este paquete se definen todas las actividades que por su comportamiento son necesarias en la arquitectura base. La actividad clave de este marco de trabajo es la representada por la clase ClientActivity que permite definir la navegación de todo el proceso de forma gráfica.

- **Activities:** Son todas aquellas actividades que por su contenido lógico son utilizadas en más de una ocasión, ya sea dentro de un mismo proceso o en procesos distintos.
- **Workflow:** Son todos aquellos procesos definidos dentro de nuestra aplicación. Aquí es donde se define la lógica de negocio de la aplicación, estos a su vez interactúan con los Business Services que son los que contienen la lógica de las funcionalidades.

### 1.6.15.2. Services

En este paquete se encuentran todas las interfaces e implementaciones de servicios pertenecientes a la arquitectura base, como son el servicio de navegación y el servicio de acceso a la información del

proceso. Estos servicios representan un factor clave dentro de la aplicación ya que son los que permiten la interacción con las instancias de proceso creadas. Estos servicios son de tipo runtime.

- Runtime: Son todos aquellos servicios arquitectónicamente importantes.
- Business: Son todos los servicios con los que interactúa un flujo de trabajo para darle solución a una lógica de negocio determinada.

### 1.6.15.3. Hosting

En este namespace<sup>14</sup> se encuentra la clase *BisonHost* la cual contiene la instancia de Runtime que maneja todas las instancias de procesos y servicios. (Proyecto Identificación, 2009)

## 1.7. Consideraciones

La nueva versión del sistema Centinel surgió como propuesta de la DIE, al necesitar nuevas funcionalidades para el buen desempeño del despacho migratorio que se realiza en los aeropuertos internacionales del país. Para lograrlo se investigaron varios sistemas, tantos nacionales como de México y Venezuela, que realizaban el chequeo migratorio, definiéndose una propuesta de solución que proporcionará ventajas significativas.

De igual forma, se llevó a cabo un estudio detallado de las herramientas de desarrollo definidas por el proyecto Identidad, Inmigración y Extranjería de la Universidad de las Ciencias. Se empleó como herramienta de estructuración del diseño los flujos de trabajo, utilizando el *Bison Framework* para la orquestación de procesos de negocio con *Windows Workflow Foundation*, todo integrado en el *Visual Studio Team System*; y como gestor de base de datos el Oracle 11g.

---

<sup>14</sup> Término en inglés que significa: espacio de nombres.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### 2.1. Introducción

En el presente capítulo se hace un estudio de los procesos de negocio que se realizan en la frontera aérea cubana, específicamente en las áreas de chequeo de pasajeros y tripulantes. Además, se comienza la primera fase del desarrollo del software, es decir, se realiza un análisis y se describen los procesos de gestión y control migratorio de la frontera aérea que quedarán actualmente en el sistema a desarrollar para la Dirección de Inmigración y Extranjería del MININT. También se plantean los requisitos funcionales y no funcionales de la aplicación a desarrollar y se modelan los mismos.

### 2.2. Análisis crítico de la ejecución de los procesos

El proceso actual se inicia cuando el pasajero se presenta en las taquillas de Inmigración para realizar el control migratorio, para posteriormente darle acceso al control aduanal en el aeropuerto.

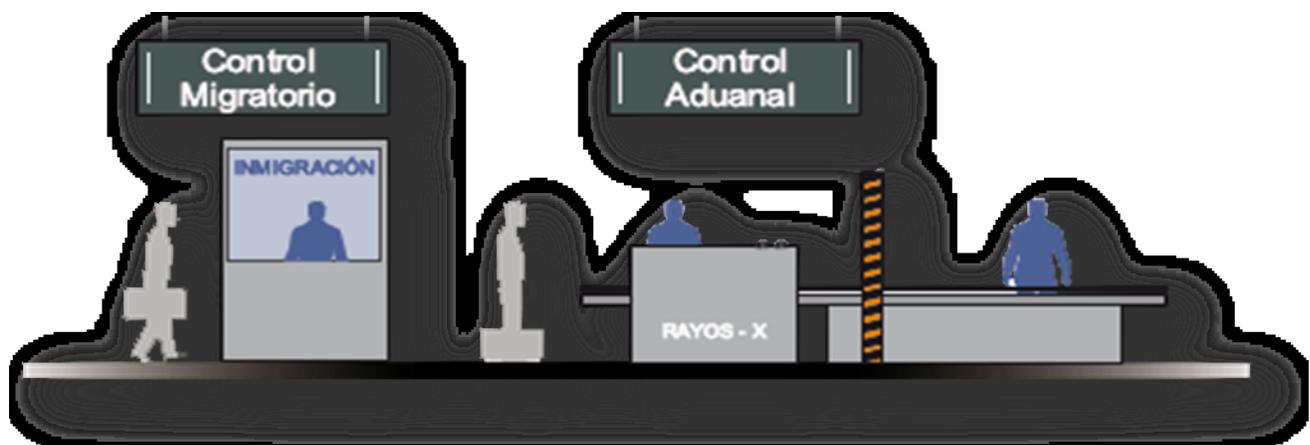


Figura 1. Vista global del Proceso de Entrada y Salida de Pasajeros y tripulantes

En este trabajo solo interesa analizar el control migratorio, describiéndose el proceso en la Figura 1, la cual provee una visión general del flujo por el que transita dicho proceso. Ver Anexo I

En el proceso actual del negocio la información de los pasajeros y tripulantes en el chequeo no está centralizada en una base de datos única a nivel nacional, resultando menos riguroso el control migratorio y en el caso de los pasajeros que vienen encargados de menores no existe en el sistema manera alguna de asociar al adulto con el menor.

Una vez que culmina el chequeo de todos los pasajeros y tripulantes de un vuelo se envían las TED<sup>15</sup> a procesamiento de datos donde se verifican los datos recogidos de los pasajeros y tripulantes en chequeo

<sup>15</sup> Tarjetas de Embarque y Desembarque

con los datos que contienen las TED, si se encuentra algún error en los datos que se encuentran introducidos en el sistema se procede a rectificarlos por las TED, posteriormente se realiza el cierre o inicio de ciclo de los pasajeros y tripulantes según corresponda.

Todas estas deficiencias hacen necesario desarrollar un sistema nuevo que cuente con un módulo donde se realice el chequeo migratorio a los pasajeros y tripulantes, velando que el inspector cuente con la menor cantidad de información posible en cuanto a los registros operativos y de impedimentos y asegure un chequeo más riguroso y seguro en la frontera área.

### 2.3. Modelado de los procesos de negocio

El modelado de los proceso de negocio son una parte esencial de cualquier proceso de desarrollo de software, permitiendo que el analista capture el esquema general y los procedimientos guiados por el negocio. Este modelo provee una vista general de cómo interactúa el sistema Centinel con la estructura organizativa de los aeropuertos internacionales; proporcionando así la justificación de la construcción de un nuevo sistema capaz de mejorar las funcionalidades existentes y automatizar los procedimientos manuales. (Solus S.A., 2007)

#### 2.3.1. Definición de los actores del negocio

Actor	Descripción
Pasajero	Es la persona que se presenta en la taquilla de Inmigración para que se le realice el control migratorio.
Tripulante	Es la persona encargada de la maniobra o servicio de la aeronave que se presenta en la taquilla de Inmigración para que se le realice el control migratorio.

Tabla 1. Actores del negocio

#### 2.3.2. Definición de los trabajadores del negocio

Trabajador	Descripción
Inspector de Inmigración	Es el encargado de realizar el chequeo de inmigración a los pasajeros o tripulantes.
Jefe de Grupo	Encargado de tomar medidas cuando el Inspector de Inmigración detecta algún problema con un pasajero.
Inspector de Circulados	Encargado de recibir todos los circulados que se detecten durante el chequeo y en caso de una circulación de aviso notificarlo vía correo

	electrónico cifrado al órgano que lo circuló.
--	---

**Tabla 2. Trabajadores del negocio**

### 2.3.3. Descripción de los procesos

El chequeo de pasajeros y tripulantes transcurre por 7 subprocesos fundamentales que aseguran de cierta manera el control migratorio en la frontera. Estos procesos son:

1. Entrada y salida de pasajeros. Ver. Anexo II
2. Entrada y salida de tripulantes. Ver. Anexo V
3. Entrevista. Ver. Anexo VIII
4. Registrar incidencia. Ver Anexo XI
5. Procesamiento de tarjetas inicio de ciclo. Ver. Anexo XIV
6. Procesamiento de tarjetas cierre de ciclo. Ver. Anexo XVII
7. Tomar decisión General. Ver. Anexo XX

#### 2.3.3.1. Descripción del proceso Entrada/Salida de Pasajeros

<b>Nombre:</b>	Entrada / Salida de Pasajeros	
<b>Objetivos:</b>	Procesar a un pasajero en la taquilla de Inmigración para permitirle o no, según las reglas definidas, entrar o salir del país.	
<b>Evento(s) que lo generan:</b>	El pasajero se presenta en la taquilla de Inmigración	
<b>Precondiciones:</b>		
<b>Poscondiciones:</b>	Deben quedar modificados y/o guardados los documentos según sea el caso.	
<b>Reglas de Negocio:</b>	RNF 1 , RNF 2 , RNF 3 , RNF 4 , RNF 5 , RNF 6	
<b>Responsable(s):</b>	Inspector de Inmigración	
<b>Cliente(s):</b>	Pasajero	
<b>Rol(es):</b>	Nombre	Función
	Pasajero	Encargado de presentarse en taquilla para que se le realice la inspección para poder entrar o salir del país según sea el caso.
	Inspector de	Es el encargado de realizar el chequeo de inmigración a los

	Inmigración	pasajeros o tripulantes.
	Jefe de Grupo	Encargado de tomar medidas cuando el Inspector de Inmigración detecta algún problema con un pasajero.
	Inspector de Circulados	Encargado de recibir todos los circulados que se detecten durante el chequeo y en caso de una circulación de aviso notificarlo vía correo electrónico cifrado al órgano que lo circuló.
<b>Entradas:</b>	Pasaporte o Documento de Viaje Equivalente, Visa, Pase de Abordar, TED	
<b>Salidas:</b>	Pasaporte o Documento de Viaje Equivalente, Visa, Pase de Abordar, TED	

**Tabla 3. Proceso Entrada/Salida de Pasajeros.**

El proceso Entrada/Salida de pasajeros se inicia cuando el pasajero se presenta en la taquilla de chequeo de cualquier aeropuerto internacional del país. El inspector de inmigración comienza el chequeo comprobando el físico de la persona con la foto de su pasaporte o documento de viaje, cuantas veces sea necesario, para establecer la correcta identificación de la persona. En caso de resultar positiva la comprobación procede a capturar la foto del pasajero, tomando las medidas necesarias para asegurar la calidad de la misma. Después le solicita la documentación al pasajero, dígame el Pasaporte o Documento de Viaje Equivalente, Visa, Pase de Abordar y Tarjeta de Embarque y Desembarque. En caso de que falte algún documento, el inspector le informa al pasajero que no puede viajar sin el Pasaporte o un Documento de Viaje Equivalente y que debe buscarlos, le entrega la documentación recogida anteriormente e informa el caso al Jefe de Grupo, quien toma una medida y registra una incidencia en el sistema en caso necesario. Si por el contrario, el pasajero presenta toda la documentación en orden el inspector selecciona el código de vuelo que le corresponde al pasajero y se dispone a introducir los datos del pasajero en el sistema. Si el pasaporte es de lectura mecánica, lo introduce en el lector mecánico de pasaporte. El sistema carga los datos leídos por el mismo, los guarda en la base de datos y realiza una búsqueda por circulado. Si por el contrario, el pasaporte no es de lectura mecánica, el inspector introduce manualmente los datos del pasaporte y presiona en el sistema la opción de buscar por circulado; el sistema guarda los datos del pasajero y realiza una búsqueda por circulado, mostrándole al inspector datos anteriores del pasajero si viajó por ese aeropuerto y determinadas alertas del sistema.

Una vez realizada la búsqueda por circulado y en caso de coincidir los datos del pasajero en chequeo con alguno en la base de datos, el sistema muestra en el módulo Chequeo que está en taquilla, una

alarma de circulación, si la circulación no es de aviso. Esta alarma también es mostrada en el módulo Monitoreo, a los jefes establecidos y oficiales de monitoreo de cámaras, alertándoles sobre el hecho en cuestión. El sistema muestra un listado con los semejantes encontrados en la base de datos de Circulado y el inspector procede a verificar si los datos del pasajero que tiene delante coinciden con alguno en el listado. En caso de coincidir los datos y el físico el inspector llama al jefe de grupo y le entrega el caso para que este tome la medida correspondiente y le entrega la documentación al pasajero. Si todo está en orden y no es la persona circulada, el inspector comprueba la validez de la. Si los datos están incorrectos, en caso de que la documentación esté incompleta, el inspector procesa la documentación modificando los documentos según el caso especificado en la Regla de negocio de frontera (RNF5); le entrega la documentación modificada al pasajero quien pasa a abordar el vuelo. Si en el proceso descrito, cuando se realiza el físico a la persona, el inspector de chequeo no está seguro de que la persona es la misma que la del pasaporte, entrega el caso al Jefe de Grupo, quien es el encargado de tomar las medidas necesarias.

Paralelamente a todo el proceso, el inspector puede realizar si desea, una serie de preguntas en busca de algún problema con la identidad del pasajero o buscando indicios de actividad enemiga, tráfico de personas, de drogas, contrabando, etc. Luego procede a capturar la barra bidimensional PDF 417, que no es más que una barra de código que posee el pasaporte con los datos del pasajero y otras características biométricas como color de ojos, estatura, color del pelo y la fotografía en escala de grises de forma cifrada, para comprobar la autenticidad del pasaporte; este proceder se realiza solo para los pasaportes cubanos que son los únicos de los cuales se conoce la manera de descifrarlos. Si todo transcurre tranquilamente, el inspector almacena la TED y/o la Visa según sea el caso, acuña la documentación, se le entrega al pasajero toda la documentación y se le permite la entrada o salida del país, culminando el proceso de chequeo de un pasajero.

Para consultar el diagrama de flujo del proceso Entrada/Salida de pasajeros y la descripción textual de las actividades del negocio Ver Anexo II y III.

### **2.3.4. Reglas del negocio**

Las reglas del negocio describen las políticas, normas, operaciones, definiciones y restricciones presentes en una organización y que son de vital importancia para alcanzar sus objetivos. En esta sección se clasificarán las reglas del negocio según las necesidades específicas del módulo Chequeo.

#### **2.3.4.1. Tipo de Reglas**

Para cada regla del negocio se especificará el nombre y una descripción que detalle en qué consiste la regla.

1. **Reglas Textuales:** Contienen "instrucciones", se expresan de forma libre (no estructurada) en lenguaje natural.
2. **Reglas del Modelo de Datos:** Engloba todas aquellas reglas que se encargan de controlar que la información básica almacenada para cada atributo o propiedad de una entidad u objeto sea válida.
3. **Reglas de Relación:** Incluye todas aquellas reglas que controlan las relaciones entre los datos.
4. **Reglas de Derivación:** Es frecuente que a partir de cierta información se pueda derivar otra, este conjunto de reglas especifican y controlan la obtención de información que se puede calcular a partir de la ya existente. (Proyecto de Identificación, Inmigración y Extranjería, 2009)

### 2.3.4.2. Relación de reglas del negocio

No	Tipo	Nombre	Descripción
RNF1.	2	Documentación a recepcionar.	Ver Anexo XXIII.
RNF2.	1	Tipo pasajero.	Ver Anexo XXIII.
RNF3.	2	Comprobar documentación.	Ver Anexo XXIII.
RNF4.	2	Modo de introducir datos del pasaporte.	Ver Anexo XXIII.
RNF5.	2	Salida de cada documentación.	Ver Anexo XXIII.
RNF6.	2	Datos a introducir.	Ver Anexo XXIII.

Tabla 4. Relación de reglas del negocio.

## 2.4. Descripción del sistema

El nuevo sistema Centinel está previsto para que satisfaga una serie de funcionalidades en el chequeo de pasajeros y tripulantes que antes no estaban concebidas. También reestructura el trabajo del aeropuerto al fusionar el módulo Procesamiento de datos con el módulo Chequeo, lo que posibilita realizar un chequeo eficiente y completo puesto que, acciones que se realizaban en el procesamiento de datos una vez despachado el vuelo, como la búsqueda por los registros operativos, ahora se va a realizar en el chequeo en tiempo real, realizando la comparación por datos idénticos y no semejantes, brindando una mayor seguridad en la frontera.

Los logros y funcionalidades que brinda el módulo chequeo son:

- Realiza un control más detallado con los menores, determinando quiénes son los responsables del mismo, para su posterior procesamiento. Para garantizar que no se queden en el país menores de forma ilegal y que no exista el tráfico de los mismos.
- Permite recoger una mayor cantidad de información del pasajero en el Chequeo como es la modalidad de la visa, el motivo de viaje, fecha de la visa de los pasajeros que viajan a los EEUU y el país destino a los cubanos residentes, lo cual permite tener una mayor cantidad de información del pasajero en tiempo real, facilitando el control sobre el mismo.
- Provee en el despacho una serie de fotos como la del pasaporte si es extranjero, y la capturada en el momento de emisión del pasaporte si es cubano, la que se captura en el despacho y la del movimiento migratorio anterior, si lo posee, proporcionándole al inspector de inmigración una mayor información visual a la hora de realizar el físico.
- Este sistema también brinda una mayor confidencialidad de la información sensible, como son los problemas que se le detectan al pasajero, al mostrarse solamente a las personas que tengan acceso a la misma, en este caso al inspector de Despacho Diferido. Y lo único que se le va a mostrar al inspector de chequeo es si está en presencia de una detención para aportar una mayor seguridad en la frontera y que el pasajero no se dé a la fuga.
- Uso de una interfaz gráfica con mayor interactividad con el usuario, permitiendo que estas posean un mayor nivel informativo de datos.
- Integración del nuevo sistema al Sistema de Inmigración, Extranjería y Ciudadanía, sincronizando las bases de datos distribuidas con la base de datos central, para que se actualicen lo más cercano posible al tiempo real.

### 2.5. Descripción de los módulos del sistema

El nuevo sistema Centinel constará con el módulo Chequeo que es el encargado de darle cumplimiento a las funcionalidades anteriormente mencionadas. Este módulo es el encargado de recoger toda la información referente al despacho migratorio que se les realiza a los pasajeros y tripulantes en el cruce de la frontera siguiendo una serie de pasos. Lo primero es realizar el físico y capturar la foto del pasajero y una vez terminado el componente que proveerá CENATAV<sup>16</sup> para la comprobación automática de la

---

<sup>16</sup> Centro de Automatización de Tecnología Avanzada.

fotografía, se tirará la foto y luego se hará el físico de forma automatizada. Luego se capturan los datos personales del viajero a chequear, de forma manual o por la lectura del pasaporte mediante el lector mecánico y se muestran en el sistema tanto estos datos como la foto capturada. Una vez cargados estos datos en el sistema se pasa a realizar la búsqueda de esta persona por los registros operativos para encontrar algún problema o impedimento que dificulte o cancele la entrada o salida del país. En caso positivo el inspector solamente podrá acceder a los problemas que él detecte en ese momento y enviar al pasajero al módulo Despacho Diferido donde se dará solución a su situación. En caso negativo el inspector también podrá seleccionar algún problema si lo detecta y enviarlo al Despacho Diferido si el mismo es grave, o seguir con el despacho normal asociándole un menor en caso que lo tenga. Después de terminado el chequeo se pasa a registrar el pasajero en el sistema y se le permite el cruce de la frontera de forma exitosa.

Para asociar algún menor, el inspector selecciona la opción en el sistema y éste le muestra una interfaz que contiene los datos del mayor y le permite realizar un chequeo al menor que transcurre de igual forma que la descripción anterior sin la opción de asociar menores. Cada vez que registra un menor, se le muestra un listado con los menores y le permite asociar otro. Si decide terminar esta acción, el sistema regresa a la interfaz del mayor mostrándole el listado de los menores que están asociados a él.

## 2.6. Descripción de los roles

Rol	Objetivo
Inspector de Chequeo	Chequear a los pasajeros y tripulantes que entran o salen del país (Ingresar los datos y la foto en el sistema; realizar la búsqueda por los registros operativos; en caso de existir algún problema se envía a la taquilla diferida o de lo contrario se le permite realizar el cruce de la frontera.)

**Tabla 5. Descripción de los roles en el módulo Chequeo**

## 2.7. Diagramas del proceso mejorado Chequeo de pasajeros y tripulantes

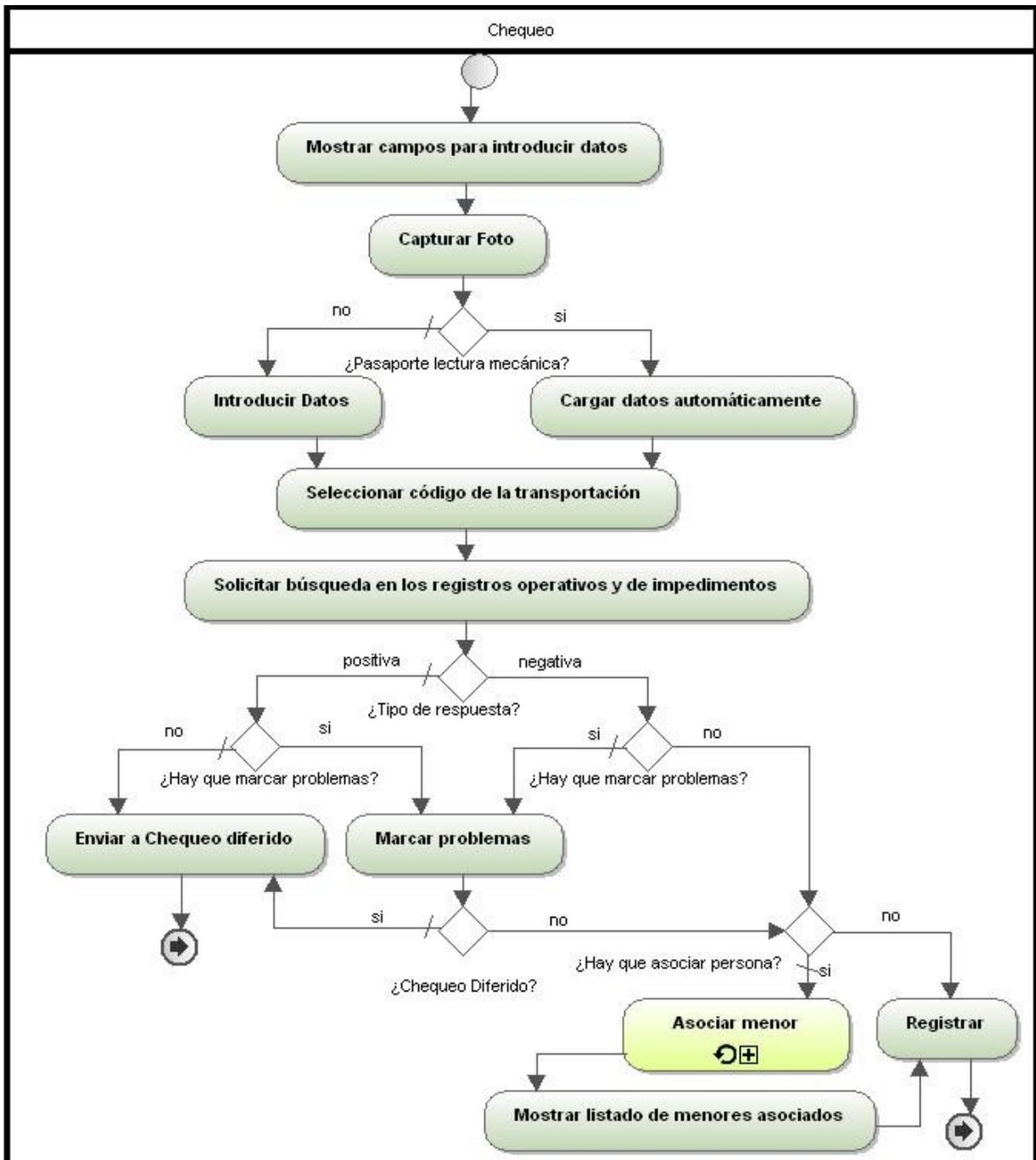


Figura 2. Chequeo de pasajeros y tripulantes.

Para mayor información sobre los diagramas de Proceso Mejorados consultar Anexos: Diagrama del proceso mejorado Asociar Menores. Ver. Anexo XXIV

## 2.8. Especificación de los requisitos de software

Un requerimiento o requisito es una condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente.

### 2.8.1. Definición de los requisitos funcionales

En el proceso de Chequeo Migratorio se definieron 4 requisitos funcionales que debe cumplir el sistema, cumpliendo con lo establecido para la primera iteración y con las funcionalidades que necesitan los clientes. Estos requisitos son:

RF1. Chequear pasajeros y tripulantes en la entrada o salida del país.

RF2. Asociar Menor. Ver Anexo XXV

RF3. Chequear Pasaporte. Ver Anexo XXV

RF4. Realizar búsqueda en los registros operativos y de impedimentos. Ver Anexo XXV

RF5. Mostrar acuerdos de Libre Visado. Ver Anexo XXV. (Proyecto Identificación, 2010)

Para la descripción de los requisitos funcionales dirigirse al Anexo XXV.

### 2.8.2. Descripción de requisitos funcionales

RF1. Chequear pasajeros y tripulantes en la entrada o salida del país.

<b>Propósito</b>	Procesar los pasajeros y tripulantes en la entrada/salida del país.	
<b>Roles</b>	Inspector de Chequeo	
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El inspector de chequeo debe estar autenticado en el sistema.</li> <li>2. Los vuelos deben estar activos.</li> <li>3. El lector de pasaporte y la cámara web deben estar conectados y activados.</li> </ol>	
<b>Entidades tratadas</b>	<b>Entidad</b>	<b>Atributos</b>
	dProcesoMigratorio	<ol style="list-style-type: none"> <li>a. idserialproceso</li> <li>b. horainicio</li> <li>c. horafin</li> </ol>

		<ul style="list-style-type: none"> <li>d. fechaitinerario</li> <li>e. idfechavencimiento</li> <li>f. movimiento</li> <li>g. numerodocumento</li> <li>h. codigovuelo</li> <li>i. idvisamig</li> <li>j. idtipodocmig</li> <li>k. idpersona</li> <li>l. idtipopersonamig</li> <li>m. idmotivoviaje</li> <li>n. idmodalidad</li> <li>o. idciudadania</li> <li>p. paisdestino</li> <li>q. idcategoriaextranjero</li> </ul>
	dVisaMig	<ul style="list-style-type: none"> <li>a. idvisamig</li> <li>b. descripción</li> <li>c. fechavisado</li> <li>d. visaeuu</li> </ul>
	nIdCategoriaExt	<ul style="list-style-type: none"> <li>a. idcategoriaext</li> <li>b. descripción</li> <li>c. fechargistro</li> </ul>
	nModalidad	<ul style="list-style-type: none"> <li>a. idmodalidad</li> <li>b. descripción</li> <li>c. fechargistro</li> </ul>
	nMotivoViaje	<ul style="list-style-type: none"> <li>a. idmotivoviaje</li> <li>b. descripción</li> <li>c. fechargistro</li> </ul>
	nProblema	<ul style="list-style-type: none"> <li>a. idproblema</li> </ul>

		<ul style="list-style-type: none"> <li>b. descripción</li> <li>c. fechargistro</li> </ul>
	nTipoDocMig	<ul style="list-style-type: none"> <li>a. idproblema</li> <li>b. descripción</li> <li>c. fechargistro</li> </ul>
	nTipoPersonaMig	<ul style="list-style-type: none"> <li>a. idproblema</li> <li>b. descripción</li> <li>c. fechargistro</li> </ul>
	dPersona	<ul style="list-style-type: none"> <li>a. idpersona</li> <li>b. idtipopersona</li> <li>c. fechargistro</li> </ul>
<b>Descripción</b>	<p>1.1. Mostrar los campos para los pasajeros y tripulantes:</p> <ul style="list-style-type: none"> <li>a. Para la Salida del país: <ul style="list-style-type: none"> <li>I. Código de la transportación</li> <li>II. Nacionalidad</li> <li>III. Ciudadanía</li> <li>IV. Tipo de documento</li> <li>V. Número del documento</li> <li>VI. Fecha de vencimiento del documento</li> <li>VII. Número de identidad</li> <li>VIII. Apellidos</li> <li>IX. Nombres</li> <li>X. Fecha de nacimiento</li> <li>XI. Sexo</li> <li>XII. Motivo del viaje</li> <li>XIII. Modalidad</li> <li>XIV. País de destino</li> <li>XV. Visa de EEUU</li> </ul> </li> </ul>	

	<p>XVI. Fecha de la visa</p> <p>XVII. Foto del pasaporte</p> <p>XVIII. Foto anterior</p> <p>XIX. Foto actual</p> <p>XX. Sin foto</p> <p>XXI. Problemas detectados</p> <p>b. Para la Entrada del país:</p> <p style="padding-left: 20px;">I. Código de la transportación</p> <p style="padding-left: 20px;">II. Nacionalidad</p> <p style="padding-left: 20px;">III. Ciudadanía</p> <p style="padding-left: 20px;">IV. Tipo de documento</p> <p style="padding-left: 20px;">V. Número del documento</p> <p style="padding-left: 20px;">VI. Fecha de vencimiento del documento</p> <p style="padding-left: 20px;">VII. Número de identidad</p> <p>VIII. Apellidos</p> <p style="padding-left: 20px;">IX. Nombres</p> <p style="padding-left: 20px;">X. Fecha de nacimiento</p> <p style="padding-left: 20px;">XI. Sexo</p> <p style="padding-left: 20px;">XII. Motivo del viaje</p> <p>XIII. Modalidad</p> <p>XIV. Foto del pasaporte</p> <p style="padding-left: 20px;">XV. Foto anterior</p> <p style="padding-left: 20px;">XVI. Foto actual</p> <p style="padding-left: 20px;">XVII. Sin foto</p> <p>XVIII. Problemas detectados</p> <p>1.2. Mostrar la opción "Video Cámara".</p> <p style="padding-left: 20px;">1.2.1. Seleccionar la imagen para la captura de la foto si se desea.</p> <p style="padding-left: 40px;">1.2.1.1. Mostrar la foto capturada en el campo "Foto Actual" si se captura la</p>
--	--

	<p>foto.</p> <p>1.2.1.2. Seleccionar el campo “Sin foto” si no se puede capturar la foto actual.</p> <p>1.3. Leer pasaporte si el pasaporte es de lectura mecánica.</p> <p>1.3.1. Comprobar automáticamente si el pasaporte posee chip.</p> <p>1.3.2. Mostrar los datos leídos del chip del pasaporte si la comprobación fue posible:</p> <ul style="list-style-type: none"><li>a. Nacionalidad</li><li>b. Ciudadanía</li><li>c. Tipo de documento</li><li>d. Número del documento</li><li>e. Fecha de vencimiento del documento</li><li>f. Apellidos</li><li>g. Nombres</li><li>h. Fecha de nacimiento</li><li>i. Sexo</li><li>j. Foto del pasaporte</li></ul> <p>1.3.3. Mostrar los datos leídos de la lectura mecánica si el resultado de la comprobación del chip es negativo:</p> <ul style="list-style-type: none"><li>a. Nacionalidad</li><li>b. Ciudadanía</li><li>c. Tipo de documento</li><li>d. Número del documento</li><li>e. Fecha de vencimiento del documento</li><li>f. Apellidos</li><li>g. Nombres</li><li>h. Fecha de nacimiento</li><li>i. Sexo</li><li>j. Foto del pasaporte</li></ul>
--	--

	<p>1.4. Introducir datos si el pasaporte no es de lectura mecánica:</p> <ul style="list-style-type: none"> <li>a. Nacionalidad</li> <li>b. Ciudadanía</li> <li>c. Tipo de documento</li> <li>d. Número del documento</li> <li>e. Apellidos</li> <li>f. Nombres</li> <li>g. Fecha de nacimiento</li> <li>h. Sexo</li> </ul> <p>1.4.1. Cargar las transportaciones activas en la Operacional.</p> <p>1.4.1.1. Seleccionar el código de la transportación correspondiente.</p> <p>1.5. Seleccionar el motivo de viaje en el campo "Motivo de viaje".</p> <p>1.6. Seleccionar el campo "Modalidad".</p> <p>1.7. Mostrar el campo "País de destino" si es un pasajero, es una salida del país y la modalidad es cubano residente.</p> <ul style="list-style-type: none"> <li>1.7.1. Seleccionar el país de destino si la condición se cumple. <ul style="list-style-type: none"> <li>1.7.1.1. Mostrar el campo "Visa de EEUU." si el país de destino es Estados Unidos.</li> <li>1.7.1.2. Seleccionar la visa de EEUU. si la condición se cumple.</li> <li>1.7.1.3. Mostrar el campo "Fecha de la visa" si el país destino es EEUU.</li> <li>1.7.1.4. Seleccionar la fecha de la visa.</li> </ul> </li> </ul> <p>1.8. Mostrar la opción de "Libre visado". Ver RF5</p> <p>1.9. Mostrar la opción "Chequeo de pasaporte".</p> <ul style="list-style-type: none"> <li>1.9.1. Seleccionar la opción "Chequeo de pasaporte" si se desea comprobar algunas medidas de seguridad del pasaporte.</li> <li>1.9.2. Mostrar campos para chequear el pasaporte. Ver. RF3.</li> </ul> <p>1.10. Mostrar la opción de "Asociar menor".</p> <ul style="list-style-type: none"> <li>1.10.1. Seleccionar la opción de "Asociar menor" si el pasajero viaja, en la entrada</li> </ul>
--	--

	<p>al país, con un pasajero menor de edad no asociado anteriormente. Ver. RF2</p> <p>1.11. Mostrar listado de menores asociados si el pasajero posee menores asociados con los siguientes datos:</p> <ul style="list-style-type: none"> <li>a. Apellidos.</li> <li>b. Nombres.</li> <li>c. Nacionalidad.</li> <li>d. Número del documento.</li> </ul> <p>1.12. Mostrar la opción “Búsqueda en Registros”.</p> <ul style="list-style-type: none"> <li>1.12.1. Seleccionar la opción “Búsqueda en Registros”. Ver. RF4.</li> </ul> <p>1.13. Mostrar los posibles problemas a detectar.</p> <ul style="list-style-type: none"> <li>1.13.1. Seleccionar los campos de los posibles problemas detectados si el inspector de chequeo encuentra problemas a la persona.</li> </ul> <p>1.14. Mostrar la opción “Enviar a Despacho Diferido” si hay problemas en chequeo.</p> <ul style="list-style-type: none"> <li>1.14.1. Seleccionar la opción de “Enviar a Despacho Diferido” si la búsqueda por los registros es positiva y se detectan problemas específicos por el inspector.</li> </ul> <p>1.15. Mostrar la opción de “Registrar”.</p> <ul style="list-style-type: none"> <li>1.15.1. Seleccionar la opción de “Registrar”.</li> <li>1.15.1.1. Guardar todos los datos de la persona y de los menores asociados si posee.</li> </ul> <p>1.16. Mostrar la opción de “Cancelar”.</p> <ul style="list-style-type: none"> <li>1.16.1. Seleccionar la opción de “Cancelar” si se desea.</li> </ul>
<b>Validaciones</b>	<p>Para realizar las operaciones:</p> <ul style="list-style-type: none"> <li>1. Ver validaciones de las entidades tratadas en el Diccionario de datos.</li> <li>2. Comprobar que todos los campos para chequear una persona estén llenos excepto el campo “Fecha de vencimiento del documento”.</li> <li>3. Comprobar que al realizar un registro de un pasajero o tripulante en el sistema no se hayan detectado problemas en los registros operativos o se hayan seleccionado problemas.</li> <li>4. Guardar todos los datos capturados en el despacho excepto “Fecha de</li> </ul>

	vencimiento del documento”.
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>Se registraron los datos de los pasajeros y tripulantes.</li> <li>Se enviaron los pasajeros y tripulantes con problemas y prohibiciones al Despacho Diferido.</li> </ol>
<b>Prototipo</b>	<p>The screenshot shows a web-based interface for checking passengers. It is divided into several sections:</p> <ul style="list-style-type: none"> <li><b>Datos del Pasajero:</b> Includes dropdowns for flight code (cub101), citizenship (Cubana), document type (Pasaporte), and document number. Text fields for surnames, names, and date of birth. A dropdown for nationality (Cubana) and radio buttons for gender (Femenino/Masculino).</li> <li><b>Datos del Viaje:</b> Includes dropdowns for travel mode (CRTA), reason for travel (TP), and destination (Argentina). A dropdown for visa type (EUA) and a date field for visa expiration. Buttons for 'Guardar Información', 'Chequeo Pasaporte', 'Buscar en Registro', and 'Libre Visado'.</li> <li><b>Otros Datos:</b> Four placeholder boxes for 'Foto Pasaporte', 'Foto Anterior', 'Foto Actualizada', and 'Video Cámara'. A checkbox for 'Sin Foto'.</li> <li><b>Problemas Detectados:</b> Checkboxes for 'Pasaporte', 'Violador de Estancia', and 'Otros'.</li> <li><b>Menores Asociados:</b> A table with columns: No, Apellidos, Nombres, Fecha Nac, Ciudadanía, No Documento, and Sexo. Below the table are buttons for 'Asociar Menor' and 'Chequear Menor'.</li> </ul> <p>At the bottom of the interface are buttons for 'Despacho Diferido', 'Registrar', and 'Cancelar', along with two empty text input fields.</p>

Figura 1. Prototipo de Interfaz Chequear pasajeros y tripulantes en la Entrada o Salida del país.

Tabla 6. Descripción del RF1. Chequear pasajeros y tripulantes en la entrada o salida del país

## 2.9. Definición de los requisitos no funcionales

Según Somerville, los requisitos no funcionales definen propiedades y restricciones del sistema que pueden ser más críticos que los requisitos funcionales, ya que son normalmente a los que debe apuntar la arquitectura y si estos no son cumplidos, el software puede no funcionar o el cliente simplemente no aceptar el producto. (Zoftwar, 2006)

A continuación se definen los requisitos no funcionales de software establecidos por las políticas del

proyecto de Identificación, Inmigración y Extranjería.

### **Usabilidad**

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

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

RnF2. El sistema será distribuido en idioma español, aunque estará preparado para soportar el multilinguaje.

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

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

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

### **Fiabilidad**

RnF5. El sistema estará disponible las 24 horas durante los 7 días de la semana.

RnF6. No se realizarán mantenimientos preventivos en horario laboral, deberán ejecutarse en un horario estipulado, para no afectar la disponibilidad del sistema.

RnF7. Las fallas del software se dividirán en dos categorías:

- Simples: la solución y la actualización se realizarán en línea en un período inferior a 10 minutos.
- Complejas: la solución y actualización se realizarán en un tiempo que se definirá posterior a una evaluación detallada.

RnF8. El sistema llevará un sistema de salvadas de errores.

RnF9. Solo se accederá a la base de datos desde la aplicación, nunca directamente desde el gestor de base de datos.

RnF10. Solo podrán acceder a las funcionalidades del sistema los usuarios que posean los permisos

suficientes.

RnF11. Se garantizará la consistencia de los datos, se realizarán comprobaciones y validaciones automáticas en todos los casos posibles.

RnF12. La información manejada por el sistema será eliminada una vez terminada de procesar.

### **Eficiencia**

RnF13. El sistema debe ser capaz de soportar una cantidad escalable de dispositivos de captura de información.

### **Restricciones de diseño**

RnF14. El sistema debe implementarse usando el lenguaje C#, sobre la plataforma ASP.NET.

RnF15. El sistema gestor de bases de datos será Oracle 11g.

RnF16. El sistema debe desarrollarse usando el IDE Visual Studio Team System 2008.

RnF17. Se utilizará el Team Explorer como control de código fuente.

### **Interfaz**

RnF18. Todas las interfaces de usuario que se definan para el sistema respetarán los patrones de diseño establecidos para la organización.

RnF19. Las ventanas del sistema contendrán claro y bien estructurados los datos, y al mismo tiempo permitirán la interpretación correcta e inequívoca de la información.

RnF20. Especificar el diseño de la interfaz de usuario que buscará la ejecución de acciones de una manera rápida, minimizando los pasos a dar en cada proceso.

RnF21. Mostrar todos los textos y mensajes en pantalla en idioma castellano.

RnF22. Diseñar su funcionamiento de modo que sea intuitivo y requiera de información mínima.

### **Interfaces de Comunicación**

RnF23. La comunicación con sistemas externos será a través de los Servicios web de Recepción de Órdenes y Monitoreo.

### **Requisitos de Licencia**

RnF24. 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 2008
- PL/SQL Developer
- Oracle Database Enterprise Edition
- Windows XP Professional SP 2
- Embarcadero/ER Studio 8.0
- Suse Enterprise Edition 10
- Windows Server 2003 Enterprise Edition

### **Requisitos Legales, de Derecho de Autor y otros.**

RnF25. Para el desarrollo del sistema se debe tener en cuenta que el mismo debe cumplir en cada uno de sus procesos con:

- Normas y estándares internacionales para documentos de identificación [ISO/IEC 7810, ISO/IEC e ICAO].
- Regulaciones y reglamentos internos. (Proyecto de Identidad, Inmigración, Extranjería Cuba, 2009)

### **2.10. Conclusiones**

En este capítulo se describió cómo funciona el negocio actual de chequeo migratorio en la frontera aérea de la DIE y se realizó un análisis crítico del mismo, Se detalló el proceso Entrada/Salida de pasajeros, los actores y trabajadores del negocio y se referenciaron las reglas del negocio.

Se describió el nuevo sistema mediante los diagramas del proceso mejorado de chequeo de pasajero y tripulantes en la entrada y/o salida del país, estableciendo el rol que tomara el inspector de chequeo y sus funciones. De igual manera se especificaron y describieron los requisitos funcionales y no funcionales para su mejor entendimiento.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

### 2.11. Introducción

El presente capítulo tiene como propósito general brindar los elementos utilizados en el proceso de desarrollo del sistema, específicamente en la construcción del software. Se hace énfasis en las características fundamentales del sistema y en el cumplimiento de las funcionalidades identificadas en el proceso de mejora. Brinda además una descripción detallada de la arquitectura de la solución y los artefactos que se obtienen con el uso de la metodología.

### 2.12. Arquitectura de la solución

La arquitectura de la solución es de n-capas, representada por 5 capas lógicas que dan un alto nivel de encapsulamiento de las responsabilidades, permitiendo reducir al máximo el acoplamiento y aumentar la reutilización entre las mismas. Esta distribución de las capas permite que se realicen grandes cambios sin tener que realizar cambios en las demás capas. Una vez que estas estén bien definidas la comunicación entre ellas se realizará solo a nivel de interfaces que permiten trabajar de manera transparente a las instancias reales.

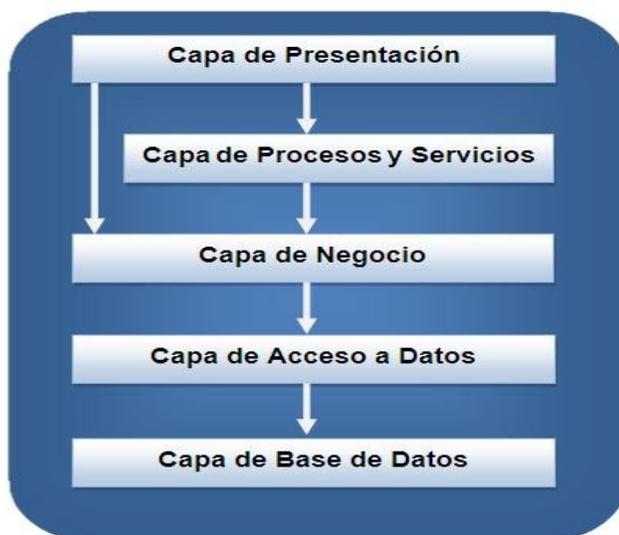


Figura 3. Vista global de la arquitectura de software.

#### 3.2.1. Descripción de las capas

A continuación se describen las distintas capas que posee la arquitectura utilizadas en el desarrollo del módulo Chequeo.

### Capa de Presentación:

Es la responsable de brindar una interfaz de comunicación entre la aplicación y un usuario dado, al estar compuesta por todas las interfaces de usuario representadas por los UserControls que contienen todos los formularios para la inserción de la información y los componentes necesarios para su correcto funcionamiento. Estos elementos pueden ser ficheros Java Script, CSS, servicios consumidos por Java Script, etc.

Al tener todos estos elementos en una capa separada se logra que cualquier cambio en las interfaces no interfiera en el buen funcionamiento de las demás capas.

Esta capa se encuentra representada por el **Proyecto Web** de la aplicación, y tiene interacción directa con la capas de **Procesos y Servicios** y con la de **Negocio**.

### Capa de Procesos y Servicios:

La capa de Procesos y Servicios tiene contenida la lógica de los procesos de negocio representada por flujos de trabajo, las actividades que se encapsulan en una actividad propia del proyecto por su nivel de reutilización o importancia lógica y los servicios del runtime, los cuales son los encargados de la interacción con los flujos de trabajo definidos, estos a su vez interactúan con los servicios de la capa de negocio haciendo uso de una fábrica de servicios que le da la instancia correcta de dicho servicio. Al mismo nivel se encuentra definida una fábrica para los servicios del runtime que le permite a la capa de presentación interactuar con estos sin necesidad de que sepan la instancia concreta del servicio que utilizan. Todos estos elementos se encuentran vinculados directamente con el Bison Framework.

Esta capa se encuentra relacionada con la capa de Negocio de la que consume servicios y con la capa de Presentación a la que le brinda servicios. Los proyectos que se relacionan en la solución son Centinel.Workflows que contiene los flujos de trabajo definidos y las actividades comunes, Centinel.Services donde se encuentran los servicios de los flujos de trabajo y la fábrica correspondiente a estos y el BisonFramework.

### Capa de Negocio:

En esta capa se recogen un conjunto de servicios de negocio que ejecutan todas las funciones presentes en la capa de Procesos y Servicios que no pueden resolverse por el flujo de trabajo y que se agrupan principalmente en tres tipos de clases: *Entidades*, *Controladoras* y *Conectoras* con sus funciones específicas.

➤ **Entidades (Centinel.Entities):** Estas clases describen cada uno de los conceptos del modelo de objeto del negocio y son los contenedores que se utilizan para el transporte de la información por las capas de negocio, la capa de Procesos y Servicios y la capa de Presentación y define tanto las entidades de negocio que pasan a la persistencia como aquellas que están constituidas con composiciones y agregaciones de las antes mencionadas y que solo tienen vida en el transcurso de un proceso.

➤ **Controladoras (Controller):** Contienen las funcionalidades del negocio y para su interacción con la base de datos hacen uso de las clases de tipo Conector. Manejan las entidades del negocio que son necesarias persistir en una fuente de datos y contienen asociada una propiedad del tipo de la interfaz que representará el conector.

➤ **Conector (IConnector):** Son las clases que permiten un alto nivel de abstracción con relación a la base de datos y definen todas las operaciones que debe realizar el conector con la fuente de datos, además de la estructura básica que debe poseer la capa de acceso a datos que le brinde el soporte que necesita.

### **Capa de Acceso a Datos:**

La capa de acceso a datos está directamente relacionada con los servicios definidos en el negocio mediante interfaces de conectores y de la fábrica de conectores que define la capa de negocio de modo tal que los cambios realizados en esta capa no representan ningún cambio para las demás. Su principal función es realizar una implementación de las interfaces definidas en la capa de negocio y al mismo tiempo trabajar directamente con la fuentes de datos establecida.

Es el componente que da soporte a las funcionalidades de la capa de negocio que se encuentran relacionadas con una fuente de datos.

**Conector (Connector):** Implementa cada una de las funcionalidades definidas basándose en la fuente de datos y el mecanismo de comunicación utilizado haciendo uso del LINQ para consultar o buscar información en la fuente de datos definida.

**EntityFramework:** Es el enlace directo entre los conectores y la fuente de datos y contiene todas las clases generadas por el *EntityFramework*.

### **Capa de Base de datos:**

Está constituida por todo el conjunto de tablas y procedimientos que permiten el almacenamiento de la información recolectada y procesada. La única capa que interactúa con esta es *la capa de Acceso a*

*Datos.*

Para más información de los componentes de las capas de la arquitectura, Ver Anexo XXXI

### 3.3. Patrones de diseño

Los Patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objeto. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan.

Es evidente que en diseños de aplicaciones hay problemas que se repiten o que son análogos, es decir, que responden a un cierto patrón. Sería deseable tener una colección de dichos patrones con las soluciones más óptimas para cada caso. En este artículo se presenta una lista con los más comunes y conocidos.

Los patrones de diseño no son fáciles de entender, pero una vez entendido su funcionamiento, los diseños serán mucho más flexibles, modulares y reutilizables.

A continuación una lista con los patrones de diseño a objetos más habituales publicados en el libro "Design Patterns", escrito por los que comúnmente se conoce como GoF (gang of four, "pandilla de los cuatro"). (Gracia, 2005)

Para el desarrollo del sistema en cuestión se han tenido en cuenta un conjunto de patrones que permiten darle flexibilidad y no constituyen cambios grandes en el rendimiento del mismo, además se identificaron un conjunto de patrones específicos para el desarrollo de Flujo de trabajo que brindan claridad y fortaleza a los diseños de estos.

#### 3.4.1. Ejemplos de Patrones

➤ **Encapsulación:** propone esconder algunos componentes, permitiendo sólo accesos estilizados al objeto. Se hace uso de este patrón en casi todas las clases que componen el sistema permitiendo que estas solo posean como elementos públicos aquellos que son exclusivamente necesarios.

➤ **Excepciones:** propone introducir estructuras de lenguaje para arrojar e interceptar excepciones. Se identificaron los diferentes tipos de errores a tratar dentro del sistema creando clases que permitan identificar cada tipo de error en el momento de ejecución.

➤ **Fábrica:** provee de una interfaz para crear familias de objetos relacionados o dependientes sin especificar los tipos concretos de clases. Su uso se encuentra centrado a la creación de los conectores

correspondientes al acceso a datos que se esté utilizando, así como en la obtención de los servicios a utilizar.

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

➤ **GRASP:** el uso de este patrón está totalmente ligado a cada componente desarrollado en el sistema, donde cada uno de ellos posee solo las funcionalidades acorde a las particularidades que lo caracterizan. (Gracia, 2005)

### 3.4.2. Patrones de Flujo de trabajo

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

➤ **Patrones de control de flujo básicos:** Estos patrones están presentes en la mayoría de los lenguajes de Flujo de trabajo, y sirven para modelar procesos secuenciales, paralelos, o aquellos que incluyan alguna decisión.

➤ **Patrón exclusive choice:** Es el punto en el flujo de trabajo donde basado en una decisión, se escoge una de las ramas que siguen en el flujo de trabajo.

➤ **Patrón sequence:** Una actividad en un proceso del flujo de trabajo es habilitada después de terminar otra actividad en el mismo proceso.

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

### 3.4.3. Estándares de codificación y tratamiento de errores

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 estos criterios de denominación deberían eliminar muchas de las preguntas más comunes en el desarrollo de un software.

Para lograr que el software tenga la calidad esperada es de suma importancia tener en cuenta el uso de técnicas robustas de codificación y buenas prácticas de programación. Con el uso de un estándar de

codificación bien definido junto a revisiones del código se logra alcanzar un mejor rendimiento para determinada aplicación y caben muchas posibilidades de que un proyecto de *software* se convierta en una aplicación fácil de comprender y de mantener.

Con el uso de estos estándares se logra elevar la capacidad de mantenimiento del código, sirve como punto de partida para los programadores manteniendo un estilo de programación uniforme y que sea el mismo en todo momento y ayuda a mejorar el proceso de codificación haciéndolo en gran medida eficiente y en muchos casos reutilizables.

Con el objetivo de facilitar el mantenimiento del software se emplearon los siguientes estándares de codificación definidos por el proyecto Identificación, Inmigración y Extranjería:

- Se podrán utilizar 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. Puede utilizar los identificadores de Pascal case en caso de tres o más caracteres.

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

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

- Empleo de comentarios en todas las declaraciones de clases y funciones más complejas para que el código del software sea más accesible a futuros cambios del equipo de desarrollo.

- Utilización de nombres de clases significativos, que expresen total o parcialmente el significado de lo que se realiza.

- No use abreviaturas o contracciones como parte del nombre de un identificador.

- Cuando sea apropiado se utilizará las siglas para reemplazar frases o nombres largos.

- Ejemplo: UI por User Interface

- Los nombres de las clases se escribieron con mayúsculas, estilo Pascal y los atributos con estilo Camello haciendo uso económico de las abreviaturas y bajo ningún concepto hacer uso del underscore (`_`).

- Organización del código de forma estructurada, en bloques de código, para una mejor lectura del mismo.

- Declarar las clases interfaz con la letra I al inicio y utilizar el mismo nombre cuando defina una pareja de clase/interface donde la clase es un estándar de implementación de la interface.
- El identificador de valor de un enumerador (Enum) lo hereda de la clase Enum Class. Utilizar Pascal case para los identificadores Enum y el nombre de sus valores. Sin utilizar la palabra Enum como sufijo en el nombre de los identificadores.
- Utilizar Camel case para los nombres de los parámetros y lograr que sean descriptivos. Los nombres del parámetro deben ser lo suficientemente descriptivo como para que el nombre del parámetro y su identificador puedan ser usados para determinar su significado en la mayoría de los escenarios. Por ejemplo, las herramientas de diseño que ofrecen ayuda en un contexto muestran como identificadores, los parámetros de un método a implementar. Los nombres del parámetro deben ser lo suficientemente descriptivo en este escenario para permitir a los desarrolladores definir los parámetros correctos.
  - Para declarar los métodos es necesario utilizar verbos o frases verbales para declarar métodos con capitalización Pascal case.
    - Dentro de las reglas para los Flujo de trabajos se tienen:
      - Reglas para nombrar las actividades
      - Utilizar un nombre o frase para declarar la actividad con capitalización Pascal case y haciendo uso económico de las abreviaturas
        - El nombre de las actividades se escriben en español.
        - Al HandleExternalEvent se le pone como prefijo “Evento” seguido de NombreActividad.
        - Ex: EventoCapturaDatos.
        - Al Code se le pone el nombre del método.
        - Al AssociationActivity se le pone como prefijo “Asociar” seguido de NombreActividad.
        - Ex: AsociarCapturaDatos.
        - Al WhileActivity se le pone como prefijo “Ciclo” seguido de NombreActividad.
        - Ex: CicloCapturaDatos.
        - Al SequenceActivity se le pone como prefijo “Contenedor” seguido de NombreActividad.
        - Ex: ContenedorCapturaDatos.
        - Al CallExternalMethod se le pone como prefijo “Invocar” seguido de NombreActividad.

- Ex: InvocarCapturaDatos.
- Al StateWFActivity se le pone como prefijo “Estado” seguido de NombreActividad.
- Ex: EstadoCapturaDatos.

Reglas para nombrar los Servicios

- Los nombres de servicios de negocio deben terminar en Service.
- Los nombres de servicios de Runtime deben terminar en RService.
- Pueden estar escritos en inglés o en español dependiendo de la complejidad de los términos, según el negocio estudiado.

En el sistema se adopta una estrategia apropiada para el tratamiento de errores, al garantizar la integridad y confiabilidad de la información que en este se registra y muestra.

Esta estrategia está definida porque es inevitable que los programas tengan errores. No importa cuánto lo pruebes y el cuidado que se tenga en desarrollar la aplicación, siempre salen errores inesperados.

Como personas relacionadas con el software se tiene que saber convivir con ellos y no taparse los ojos intentando ignorarlos. Puesto que los errores forman parte del software y son bastante peligrosos, en cuanto a coste e imagen, por lo que es mejor tenerlos controlados.

Normalmente cuando alguien encuentra un error se comunica con el programador y este lo intenta arreglar. (Gracia, 2004)

En el sistema, cuando se introduce información incorrecta en algunos de los campos, se le notifica al usuario el error con un mensaje con letras rojas debajo del campo que tiene errores. Los mensajes de error que emite el sistema se muestran en un lenguaje de fácil comprensión para los usuarios. Cuando se introduce información en un formulario y por alguna razón no se completan todos los campos el sistema será capaz de enviar una alerta al usuario rellene el formulario y hasta que no se le de solución a esa situación el sistema no permitirá registrar información en la base de datos.

### 2.13. Clases Utilizadas

Con la arquitectura en capas se hace imprescindible separar en clases distintas el negocio y el acceso a datos, así como las clases conectoras que se encargan de acceder directamente a la base de datos y manipular las entidades con independencia plena del gestor de base de datos que se utilice. Esto permite que la realización de cambios en la base de datos solo provoque la actualización de estas clases. Las

clases gestoras utilizan las clases conectoras para recibir la información que estas brindan para conformar las clases del negocio

### 3.5.1. Clases Entidades

Las clases entidades representan información que posee una larga vida a través de atributos que pueden tomar diferentes valores en dependencia del problema y corresponden a los objetos que modelan entidades o acontecimientos del mundo real de los que el software debe utilizar información

Para el desarrollo del módulo Chequeo se tienen que utilizar varias entidades del negocio que aseguren el éxito del proceso, entre ellas se tienen: MigratoryProcess, TravelReason, Modality, entre otras que son de uso común para todo el proyecto

A continuación se muestra la clase entidad MigratoryProcess:

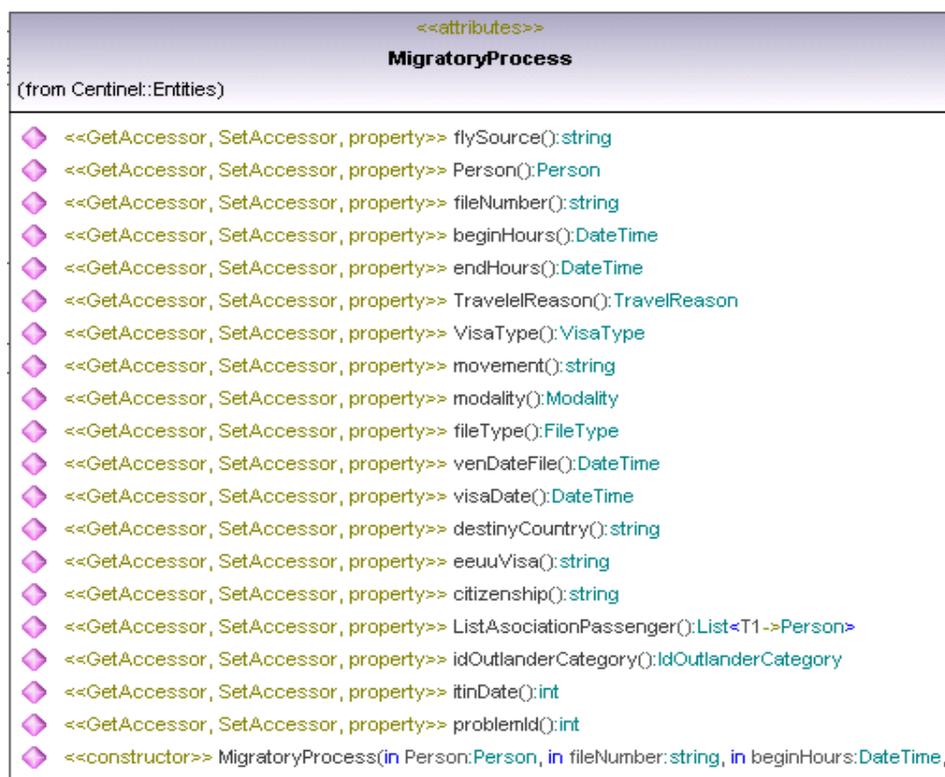


Figura 4. Entidad Proceso Migratorio

La entidad MigratoryProcess (Proceso Migratorio), es la de mayor relevancia en el módulo Chequeo porque es la que agrupa toda la información relacionada con el movimiento migratorio en cuestión, así como los datos del vuelo y de la documentación del pasajero. Contiene además la información de la

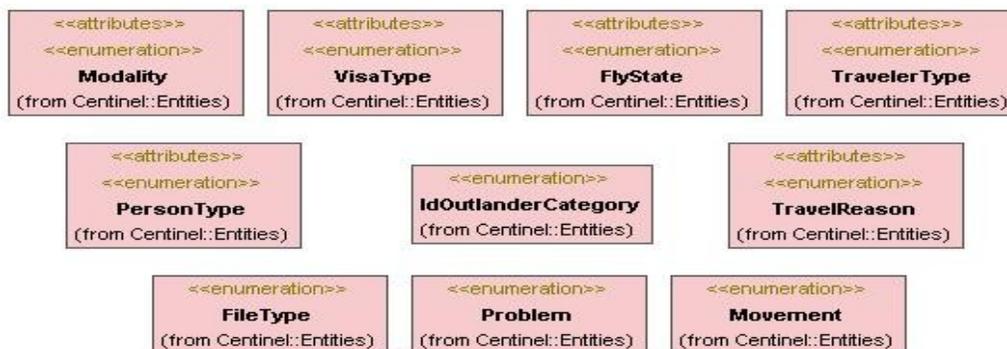
persona que es otra entidad que por ser genérica del proyecto la usan todos los subsistemas y contiene los datos específicos de las personas. Esta entidad es la que permite tener un control del historial migratorio de las personas al registrar todos los movimientos migratorios que ha realizado en un periodo determinado.

Para el buen funcionamiento del módulo, se utilizan otras entidades del negocio que se pueden ver en la siguiente figura:



**Figura 5. Entidad Expediente de pasajero**

Por otra parte la entidad PassengerFyle (Expediente de pasajero) recoge la información de los trámites migratorios que realizan los pasajeros y tripulantes que hacen paso por Cuba.



**Figura 6. Entidades usadas para asegurar el funcionamiento del módulo Chequeo**

Además el sistema trabaja con algunos enumeradores que permiten mayor nivel de información y que son importantes en el proceso de control migratorio, estos son FileType, que no es más que la que recoge los distintos tipos de documentos que pueden poseer los pasajeros y tripulantes cuando realizan algún

viaje. TravelReason que contiene las motivaciones de viajes de los pasajeros. FlyState, que brinda la información del estado de los distintos vuelos para poder trabajar con ellos si dicho estado lo permite y otros enumeradores como: VisaType, Tipo de Visa; TravelerType, tipo de persona desde el punto de vista migratorio, que puede ser pasajero o tripulantes y Modality, que recoge la modalidad del viaje a realizar. De igual manera se tiene la entidad IdOutlanderCategory, que recoge las categorías de extranjero que viajan a nuestro país

Todos estos términos son propios del proceso de control migratorio.

### 3.5.2. Diagrama de clases del diseño

A continuación se muestra el diagrama de clases del diseño del módulo Chequeo:

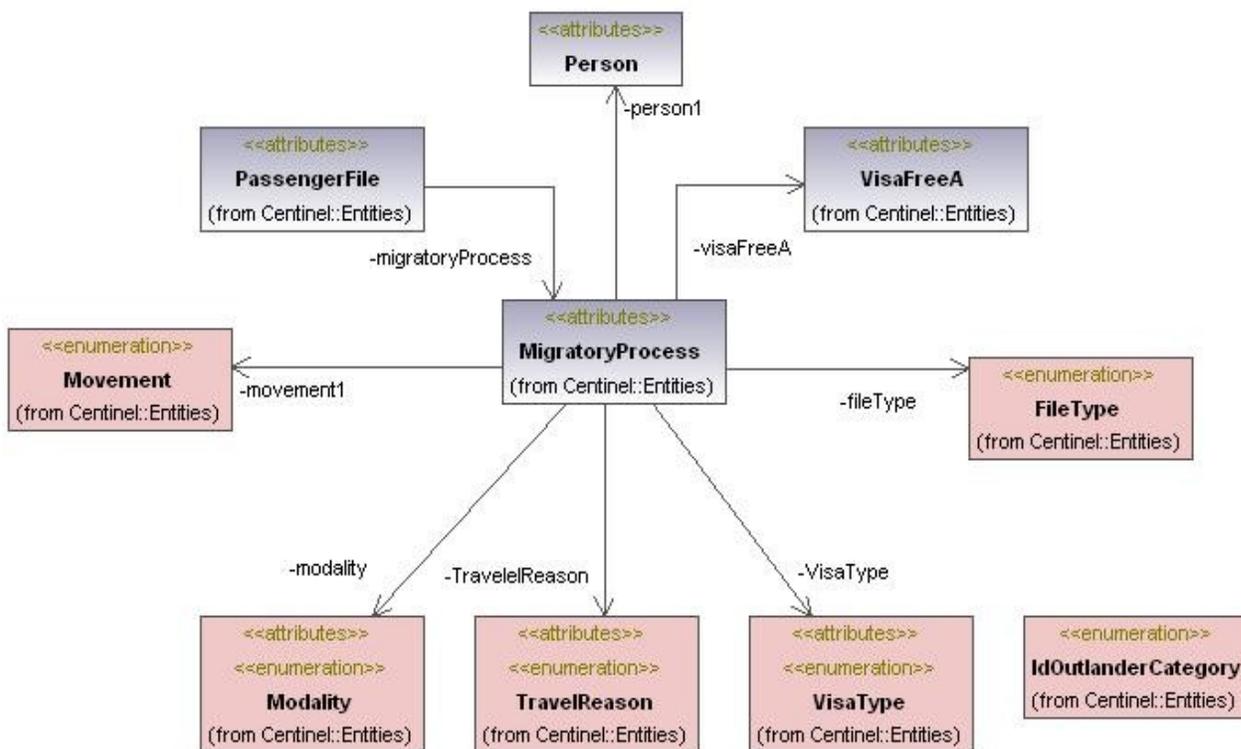


Figura 7. Diagrama de clases del diseño del módulo Chequeo

## 2.14. Servicios del Sistema

Los servicios son pequeños script que realizan acciones sobre una selección haciendo uso de las funcionalidades de la aplicación que los instala o funcionan de forma independiente. Según se instalen a partir de una aplicación o vengan integrados en el sistema tendrán ubicaciones diferentes.

Hay dos tipos de servicios: los que procesan y los que suministran. Los primeros sólo procesan datos para ser usados por otra aplicación y los segundos suministran un resultado para la aplicación que usa dicho servicio. Aunque podemos encontrar servicios que realizan las dos funcionalidades. (Santamaría, 2009)

### 3.6.1. Descripción de los Servicios

Los servicios han sido desglosados, según las funciones que realizan, en dos grupos: Servicios de Runtime y Servicios de Negocio.

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

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

Entre los servicios que utiliza el módulo Chequeo, que son servicios de Runtime, se encuentra el servicio InformationCaptureRService que es el encargado de cargar toda la información que se recoge en el Chequeo. Esta información es la referente a la persona con su fotografía y los datos del Proceso Migratorio que realiza.

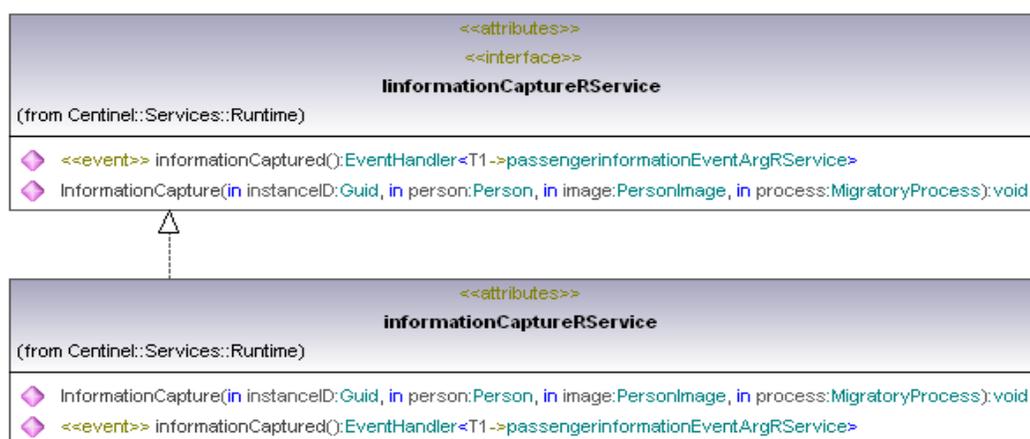


Figura 8. Servicio InformationCaptureRService.

Por otra parte se tiene el servicio `AssociateChildrenRService`, que es el servicio que permite que se asocie un menor de edad a algún pasajero que se hace responsable de su custodia. Este servicio maneja la información relacionada con los datos del menor, así como los argumentos del mayor responsable del mismo, y permite visualizar un listado con todos los menores asociados a determinado mayor.

Para consultar los demás servicios que sustentan el funcionamiento del módulo Chequeo, Ver Anexo XXIX

### 2.15. Diseño de los Flujos de Trabajo

Un flujo de trabajo es un conjunto de actividades guardadas como un modelo que describe un proceso del mundo real. El trabajo pasa a través del modelo desde el principio hasta el final, y las actividades pueden ser ejecutadas por personas o por funciones del sistema y llevan a cabo la manipulación y el procesamiento de la información.

Las actividades utilizadas en el diseño del Flujo de trabajo en el módulo Chequeo se relacionan a continuación:

**CodeActivity:** Permite agregar código de cualquiera de los lenguajes que soporte en el lugar en donde se coloque dentro del flujo de trabajo.

**HandleExternalEventActivity:** Espera porque el evento especificado sea lanzado bloqueando el flujo de trabajo, generalmente se utiliza como fuente de extracción de información con el ambiente que se encuentra fuera del flujo de actividades.

**IfElseActivity:** Ejecuta una de varias bifurcaciones en dependencia de la condición establecida en cada rama del bloque *ifElse*. En caso de que la condición sea positiva se ejecutan las actividades contenidas en la actividad *IfElseBranchActivity* quien representa una bifurcación de una actividad *IfElseActivity*.

**WhileActivity:** Permite la ejecución de una o varias actividades dentro del flujo de trabajo de forma cíclica hasta que se cumpla la condición indicada.

**CallExternalMethodActivity:** Permite la comunicación de entrada y salida con un servicio local externo al flujo de trabajo.

**FaultHandlerActivity:** Se usa para el control de excepciones del tipo especificado.

**ClientActivity:** Permite definir las interacciones del proceso con los usuarios de la aplicación. Es

necesario especificarle la URL del componente representado.

**AssociationActivity:** Permite trasladar la información obtenida en la propiedad *Destiny* a la propiedad *Origin*.

**SavePropertyActivity:** Salva la información contenida en el flujo de trabajo en el cual se coloca.

**Terminate:** Finaliza la instancia del flujo de trabajo.

**StateActivity:** Guarda los estados en los que se encuentra el flujo de trabajo permitiendo recuperar el mismo en un determinado estado.

### 3.6.1. Descripción del flujo de trabajo Chequeo

El flujo de trabajo WFChequeo1 se inicia al asignarle al flujo de trabajo un estado “EstadoChequeo”, luego entra directamente al ciclo CicloCaptarBuscar que está formado por un contenedor de actividades nombrado “ContenedorChequeo”. Este contenedor comienza con un servicio que espera por un evento lanzado por la acción de un botón en la interfaz Chequeo y dirige el flujo hacia una condicional. La primera rama y segunda rama se hacen verdaderas cuando el evento que se lanza es la acción del campo “Buscar por Registros Operativos” de la interfaz principal y la función que realiza la primera rama es lanzar el evento “CaptarInformación”, cuyo objetivo es capturar la información recogida en la interfaz y guardarla en el WF. La segunda rama inicia una actividad paralela del WF llamada “ParaleloBuscarRegistro”, que solicita los servicios a los registros operativos Identidad, Circulado, Persona de Interés Migratorias (PIM, por sus siglas en español) e Impedimentos que tomando los datos que se le envían, devuelven si hay alguna coincidencia. Esta actividad paralela lanza además el servicio “BuscarWFigualEstado”, que verifica que no exista otra persona con el mismo estado para verificar que no existan violaciones en la frontera, y el servicio “BuscarWFDiferentes” para comprobar que el pasajero no sea violador de estancia<sup>17</sup> y devuelve los datos del trámite anterior en caso de existir en la base de datos.

La tercera rama de la condicional se verifica con la acción del campo “Chequeo de Pasaporte”, de la interfaz Chequeo. Su función es mostrarle al inspector una interfaz donde puede realizar la comprobación de la autenticidad del pasaporte. Así como datos biométricos del pasajero para asegurar que sea quien dice. La cuarta rama se comprueba con la acción del campo “Libre Visado”, de la interfaz Chequeo. Su función es mostrarle al inspector una interfaz con los acuerdos de libre visado del país según el filtrado

---

<sup>17</sup> Pasajero que viola el tiempo de estancia en el país, según la modalidad de viaje.

que se le realice a la ciudadanía del pasajero. Por último la quinta rama se hace verdadera cuando se acciona el campo “Registrar” y se registra la información en el WF o envía el pasajero a despacho diferido por presentar algún problema.

Al terminar este ciclo entra en otro ciclo llamado “CicloAsociarChequear”, que tiene en su funcionamiento la condicional “TipoAccionAsociar”, que contiene una sola rama y dirige al flujo de trabajo al nuevo flujo de trabajo “WFAsociar menor”, y espera por una respuesta de este WF.

Para consultar el diseño de los flujos de trabajo del módulo Chequeo, Vea Anexo XXVII.

En la siguiente figura se representa el diseño del flujo de trabajo principal del módulo Chequeo “WFChequeo1”.

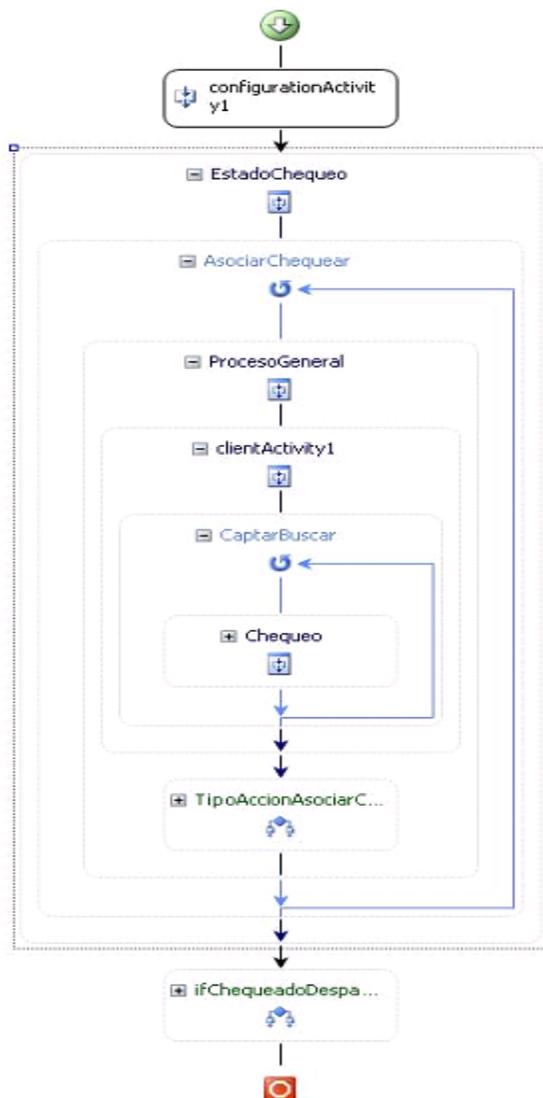


Figura 9. Diseño del flujo de trabajo Chequeo (WFChequeo1).

## 2.16. Modelo de datos

Un modelo de datos es un lenguaje utilizado para la descripción de una base de datos. Por lo general, un modelo de datos permite describir las estructuras de datos de la base de datos (el tipo de los datos que incluye la base de datos y la forma en que se relacionan), las restricciones de integridad (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación de los datos (agregado, borrado, modificación y recuperación de los datos de la base de datos).

Por lo general, un modelo de datos presenta dos sub-lenguajes: un Lenguaje de Definición de Datos o Data Definition Language (DDL, por sus siglas en inglés), cuya función es describir, de una forma abstracta, las estructuras de datos y las restricciones de integridad; y un Lenguaje de Manipulación de Datos o Data Manipulation Language (DML, por sus siglas en inglés), que se orienta a describir las operaciones de manipulación de los datos. A la parte del DML enfocada a la recuperación de datos, se la suele conocer como Lenguaje de Consulta o Query Language (QL, por sus siglas en inglés). (Definición de, 2008)

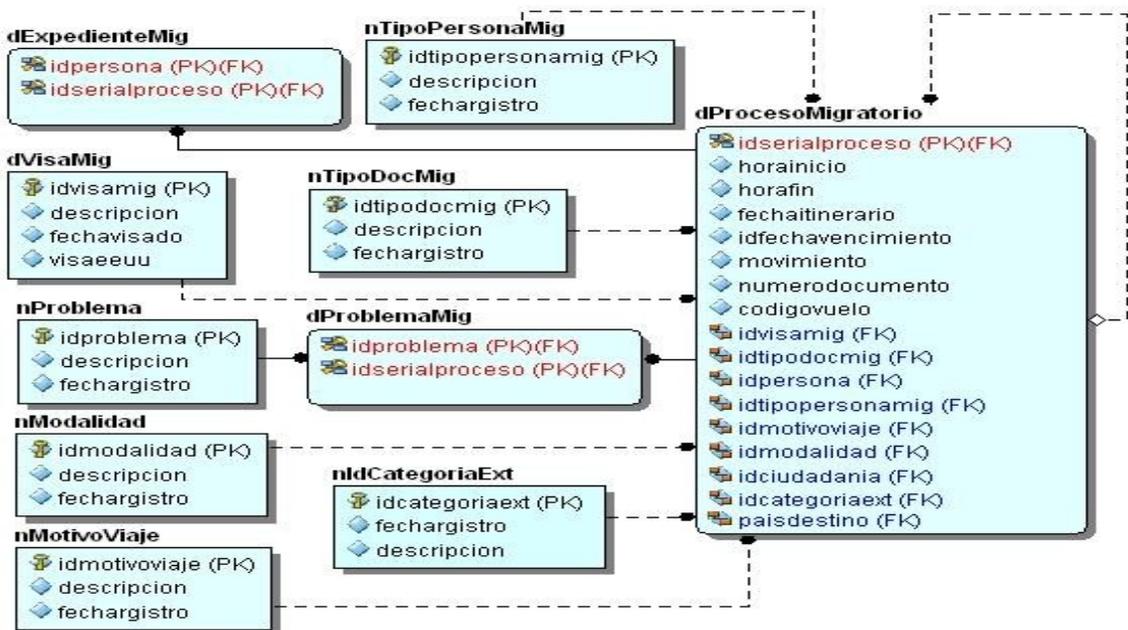


Figura 10. Modelo de Datos

### 3.8.1. Descripción de las entidades fundamentales

A continuación se describen una de las entidades fundamentales para el módulo Chequeo. Para tener mejor dominio de las entidades que intervienen en el modelo de datos ver Anexo XXVI.

Nombre entidad	dProcesoMigratorio			
Definición	Entidad que representa los datos de los procesos de migración.			
Atributo	Dominio	Tipo de dato	Nulo	Definición
idserialproceso	tIdentificadorUnico	UNIQUE ID	N	Identificador único para el proceso migratorio.
horainicio	tFechaHora	DATE TIME	N	Hora de inicio.
horafin	tFechaHora	DATE TIME	N	Hora de fin.

fechaitinerario	tFechaHora	DATE TIME	N	Fecha de itinerario.
idfechavencimiento	tFechaHora	DATE TIME	N	Fecha de vencimiento del pasaporte
movimiento		BIT	N	Si se le dio entrada o no al proceso.
numerodocumento	tNumeroIdentidad	VARCHAR(11)	N	Número del documento de identificación.
codigovuelo	tcadena50	VARCHAR(50)	N	Identificador del vuelo
idvisamig	tIdNomenclador	NUMERIC(4, 0)	N	Identificador de la visa de migración.
idtipodocmig	tIdNomenclador	NUMERIC(4, 0)	N	Identificador para el tipo de documento migratorio.
idpersona	tIdentificadorUnico	UNIQUEID	N	Identificador único de la persona.
idtipopersonamig	tIdNomenclador	NUMERIC(4, 0)	N	Identificador para el tipo de persona migratoria.
idmotivoviaje	tIdNomenclador	NUMERIC(4, 0)	N	Identificador para el motivo de viaje.
idmodalidad	tIdNomenclador	NUMERIC(4, 0)	N	Identificador para la modalidad de viaje.
idciudadania	tIdNomenclador	NUMERIC(4, 0)	N	Identificador para la ciudadanía del pasajero.
paisdestino	tCodigoPais	VARCHAR(3)	N	Identificador del país hacia donde viaja.
idcategoriaext	tIdNomenclador	NUMERIC(4, 0)	N	Identificador para la categoría extranjera.

**Tabla 7. Descripción de la entidad Proceso Migratorio**

## 2.17. Conclusiones

En este capítulo se describió la arquitectura del sistema y las funciones de cada una de sus capas para asegurar que cada nivel se especialice en una actividad específica, evitando la dependencia a una sola tecnología o aplicación. De igual manera se describieron las clases entidades y los servicios que soportan las funcionalidades del módulo Chequeo y que sustentan las especificaciones de los clientes.

Se detalló el diseño del flujo de trabajo obtenido a partir del diagrama de mejora de procesos, donde cada actividad da cumplimiento a requerimientos planteados en el capítulo anterior. Además el modelo de datos que describen las clases de la base de datos; este modelo está definido para cuando se integren todos los componentes del Sistema de Inmigración y Extranjería y se ponga en funcionamiento la base de datos central.

## CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

### 4.1. Introducción

La implementación es el flujo de trabajo en que se implementa el sistema en términos de componentes: ejecutables, ficheros de código fuente, scripts, entre otros. Tiene como objetivo principal desarrollar la arquitectura y el sistema como un todo, así como definir la organización del código. Los diagramas de despliegue y componentes conforman lo que se conoce como un modelo de implementación al describir los componentes a construir y su organización y dependencia entre nodos físicos en los que funcionará la aplicación. Durante esta etapa se obtiene el modelo de implementación que relaciona componentes y subsistemas.

Las pruebas se realizan con el objetivo de garantizar la calidad del producto desarrollado. Estas demuestran hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones y parecen alcanzarse los requisitos de rendimiento. Asimismo, los datos que se van recogiendo a medida que se lleva a cabo la prueba proporcionan una buena indicación de la fiabilidad del software y, de alguna manera, indican la calidad del software como un todo.

### 4.2. Diagrama de Despliegue

El Diagrama de despliegue es un modelo de objetos compuesto por nodos y sus relaciones, que describen la distribución física de un sistema. Muestra las relaciones que se establecen entre los componentes de software y hardware, representándolos mediante nodos estrechamente conectados, para un funcionamiento exitoso de la aplicación.

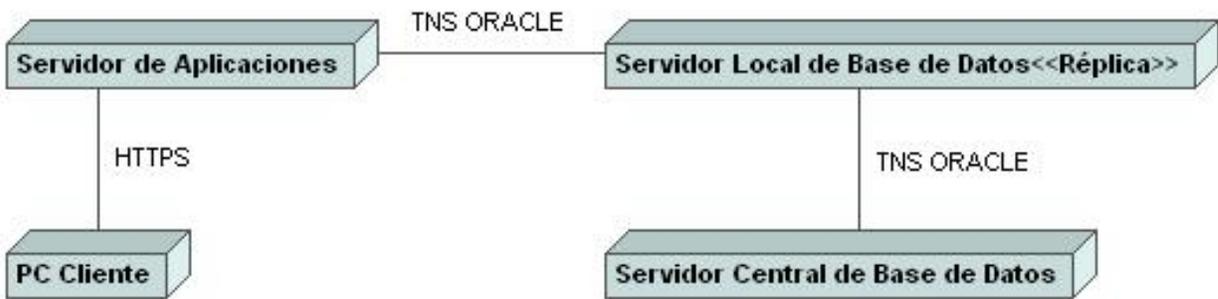


Figura 11. Diagrama de despliegue del módulo Chequeo

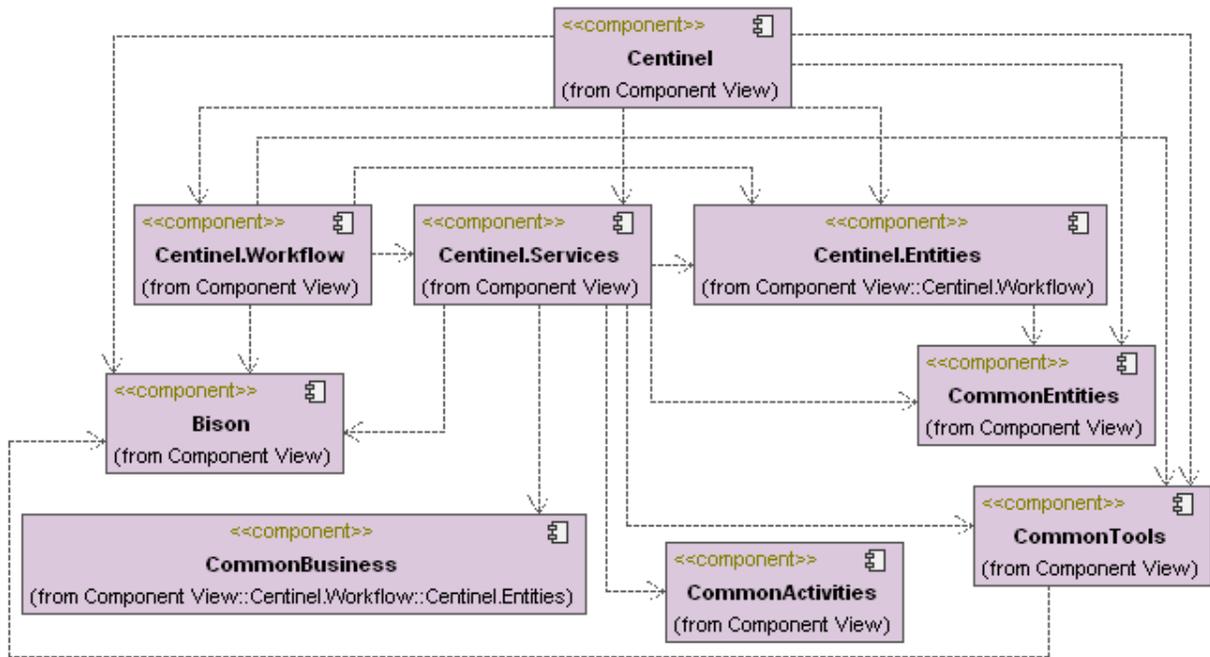
El Diagrama de despliegue del módulo Chequeo muestra la distribución que deben tener los nodos que lo conforman. La PC Cliente representa los puestos de trabajo en las taquillas de inmigración de los

diferentes aeropuertos internacionales del país. Esta PC Cliente se conecta directamente con el servidor de aplicaciones mediante el protocolo HTTPS<sup>18</sup>, y éste a su vez, se conecta al Servidor Local de Base de Datos, que constituye una réplica de la Base de Datos Central y cuando esta base de datos local no disponga de cierta información en la réplica, se conecta al Servidor Central de Base de Datos, ambos mediante el protocolo de comunicaciones de Oracle, Transparent Network Substrate, Sustrato de red transparente( TNS ORACLE, por sus siglas en inglés).

### 4.3. Diagramas de Componentes

Un diagrama de componentes muestra las organizaciones y las dependencias lógicas de componentes de software. Este puede contener paquetes que se utilizan para agrupar elementos del modelo.

En la siguiente figura, se representa el diagrama de componentes referente al módulo Chequeo.



**Figura 12. Diagrama componentes del módulo Chequeo**

El diagrama de componentes del módulo Chequeo muestra la interacción del componente principal del Centinel con los demás componentes comunes del proyecto, de los que utiliza ciertas funcionalidades,

<sup>18</sup> versión segura del protocolo HTTP (HyperText Transferer Protocol) que implementa un canal de comunicación seguro y basado en SSL (*Secure Socket Layers*) entre el navegador del cliente y el servidor HTTP.

como son: Bison, CommonTools, CommonEntities, CommonBussines y CommonActivities. De igual forma se muestra cómo se relaciona con los componentes propios del Centinel, para asegurar el correcto funcionamiento del módulo. Estos son: Centinel.Workflows, Centinel.Services y Centinel.Entities.

#### 4.4. Interfaces del módulo Chequeo

En el funcionamiento del módulo Chequeo se definieron 5 funcionalidades principales, de ellas solo 4 utilizan interfaces gráficas, Chequeo, Asociar menores, Chequeo de pasaporte y Mostrar convenio de libre visado. Para consultar estas interfaces Ver Anexo XXVII.

#### 4.5. Pruebas

Las pruebas son aplicadas para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. Cada una de estas se realizan en determinados momentos del ciclo de vida del software.

Alfredo Weitzenfeld argumenta que existen tres niveles principales para aplicar las diferentes técnicas de pruebas. (Weitzenfeld, 2005.):

*Prueba de unidad:* Mediante esta prueba solamente una unidad es probada como tal, como una clase, un paquete de servicio o un subsistema.

*Prueba de integración:* En ella se verifica que las unidades trabajen juntas correctamente. Ambas pueden ser realizadas mediante casos de uso de pruebas, los cuales pueden ser aplicados a clases, paquetes de servicio, subsistema o el sistema completo.

*Prueba de sistema:* Verifica el sistema completo o su aplicación como tal. Se toma el punto de vista del usuario final y los casos de uso de prueba ejecutan acciones típicas del usuario.

##### 4.6.1. Diseño de casos de pruebas

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. Es la entidad más simple que siempre es ejecutada como una unidad, desde el comienzo hasta el final. Los casos de pruebas deben verificar si el producto satisface los requisitos del usuario, tal y como se describen en las especificación de estos y si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño. (Jacobson, et al., 2000)

A continuación se muestra un ejemplo de un diseño de caso de prueba que responde al requisito de Chequear pasajeros y tripulantes en la entrada o salida del país.

CPR 1: Chequear pasajeros y tripulantes en la entrada o salida del país.

1.1. Condiciones de Ejecución:

Para que el requisito se ejecute:

- El inspector de chequeo debe estar autenticado en el sistema.
- Los vuelos deben estar activos.
- El lector de pasaporte y la cámara web conectados y activos.

**Flujo General del Requisito:**

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
1: <b>Chequear pasajeros y tripulantes en la entrada o salida del país.</b>	E.C 1.1: Mostrar los campos para los pasajeros y tripulantes.	<p><i>Deben mostrarse los campos:</i></p> <p>Para la salida:</p> <ul style="list-style-type: none"> <li>a. Código de la transportación</li> <li>b. Nacionalidad</li> <li>c. Ciudadanía</li> <li>d. Tipo de documento</li> <li>e. Número del documento</li> <li>f. Fecha de vencimiento del documento</li> <li>g. Número de identidad</li> <li>h. Apellidos</li> <li>i. Nombres</li> <li>j. Fecha de nacimiento</li> <li>k. Sexo</li> <li>l. Motivo del viaje</li> <li>m. Modalidad</li> <li>n. País de destino</li> <li>o. Visa de E.U.</li> </ul>	<p>1- Se selecciona la opción <i>Chequeo de pasajeros en el menú ubicado en el panel izquierdo.</i></p>

		<p>p. Fecha de la visa</p> <p>q. Foto del pasaporte</p> <p>r. Foto anterior</p> <p>s. Foto actual</p> <p>t. Sin foto</p> <p>u. Problemas detectados</p> <p>Para Entrada:</p> <p>a. Código de la transportación</p> <p>b. Nacionalidad</p> <p>c. Ciudadanía</p> <p>d. Tipo de documento</p> <p>e. Número del documento</p> <p>f. Fecha de vencimiento del documento</p> <p>g. Número de identidad</p> <p>h. Apellidos</p> <p>i. Nombres</p> <p>j. Fecha de nacimiento</p> <p>k. Sexo</p> <p>l. Motivo del viaje</p> <p>m. Modalidad</p> <p>n. Foto del pasaporte</p> <p>o. Foto anterior</p> <p>p. Foto actual</p> <p>q. Sin foto</p>	
--	--	--	--

	E.C 1.2: Introducir datos si el pasaporte no es de lectura mecánica.	<i>Se introducen manualmente los datos:</i>  i. Nacionalidad. j. Ciudadanía. k. Tipo de documento. l. Número del documento. m. Apellidos. n. Nombres. o. Fecha de nacimiento. p. Sexo. q. Código vuelo. r. Motivo de viaje. s. Modalidad.	1-Se selecciona el código de la <i>transportación correspondiente</i> .  2-Se selecciona el <i>motivo de viaje en el campo "Motivo de viaje"</i> .  3-Se selecciona el <i>campo "Modalidad"</i> .
	E.C 1.3: Mostrar el campo "País de destino"	Si la persona es un pasajero, es una salida del país y la nacionalidad y la ciudadanía del pasaporte es cubana, y la modalidad es cubanos residentes.	1-Se selecciona el <i>país de destino</i> .
	E.C 1.4: Mostrar el campo "Visa de E.U."	Si el país de destino es E.U.	1-Se introduce el <i>tipo de visa</i> .
	E.C 1.5: Mostrar el campo "Fecha de la visa"	Si el país destino es E.U.	1-Se introduce la <i>fecha de la visa</i> .
	E.C 1.6: Mostrar la opción de "Libre visado".	Se debe mostrar la opción " <i>Libre visado</i> "	1- Se selecciona la <i>opción "Libre visado"</i> .  2- Ver caso de prueba " <i>Mostrar acuerdo de libre visado</i> "

	E.C 1.7: Mostrar la opción de “Asociar menor”.	Se debe mostrar la opción “Asociar menor”	1- Se selecciona la opción de “Asociar menor” 2- Ver caso de prueba “Asociar Menores”
	E.C 1.8: Mostrar listado de menores asociados.	Si el pasajero posee menores asociados con los siguientes datos:  e. Apellidos. f. Nombres. g. Nacionalidad. h. Número del documento.	
	E.C 1.9: Mostrar la opción “Búsqueda en Registros”.	Se debe mostrar la opción “Búsqueda en Registros”	1- Se selecciona la opción de “Búsqueda en Registros” 2- Ver caso de prueba “Búsqueda en Registros”.
	E.C 1.10: Mostrar la opción “Registrar”	Si la búsqueda por registro no reportó ningún problema en el pasajero y si el inspector tampoco detectó ninguno.  Se debe mostrar la opción “Registrar”	1- Se selecciona la opción “Registrar”.
<b>Chequear pasajeros y tripulantes en la entrada o salida del país.</b>	E.C 1.10 a: Mostrar la opción “Enviar a Despacho Diferido”	Si la búsqueda por registro reportó algún problema con el pasajero.	1-Se selecciona la opción “Enviar a Despacho Diferido”. 2- Ver caso de prueba “Búsqueda en Registros”

	E.C 1.10 b: Mostrar la opción "Enviar a Despacho Diferido"	Si el inspector seleccionó algún problema que detectó en el pasajero. Se debe mostrar la opción "Enviar a Despacho Diferido"	1- Se seleccionan los campos de los posibles problemas detectados. 2- Se selecciona la opción "Enviar a Despacho Diferido".
	E.C 1.10 c: Mostrar la opción "Cancelar"	Si la búsqueda por registro no reportó ningún problema en el pasajero y si el inspector tampoco detectó ninguno. Se debe mostrar la opción "Cancelar".	1- Se selecciona la opción de "Cancelar" si se desea.

Tabla 8. Caso de prueba del requisito Chequear pasajeros y tripulantes en la entrada o salida del país.

### 4.6.2. Pruebas Unitarias

Las pruebas unitarias son un punto básico para mejorar la calidad del sistema desarrollado. Una prueba unitaria no es más que un código que prueba otro código. Team System facilita enormemente la creación de estas pruebas y el manejo de las mismas. Haciendo pruebas unitarias obtenemos pruebas que podemos repetir. Al modificar código podemos volver a pasarlas de manera automatizada y detectar errores en los cambios que hemos incluido. Seguro que a todos nos ha pasado que hemos hecho un cambio sin importancia que ha provocado efectos no deseados en otros módulos. (Landa, 2009 )

Para el módulo Chequeo se desarrollaron pruebas unitarias a sus métodos entre las que podemos encontrar la realizada al método *VisaFreeAgreemnet* :

Método original

```
public void VisaFreeAgreemnet(Guid instanceID, VisaFreeA visaFree, MigratoryProcess
process, bool visaFreeVisited, bool visaFreeFound)
{
    if (this.FreeVisaChequed != null)
    {
        string conection = @"Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=D:\rep\Centinel\ConvenioLibreVisado.xls; Extended Properties=Excel 8.0";
        OleDbConnection excelConection = new
OleDbConnection(conection); //conexion con el excel
```

```

        excelConection.Open();
        OleDbCommand objCmdSelect = new OleDbCommand("SELECT * FROM [Hoja1$]",
excelConection); //seleccionar la hoja con la que se encuentra la inf en el excel
        OleDbDataAdapter objAdapter = new OleDbDataAdapter();
        objAdapter.SelectCommand = objCmdSelect;
        DataSet objDataSet = new DataSet();
        objAdapter.Fill(objDataSet, "XLData");
        //DataSet obj = RecoverExcelVisaFree(excelConection);
        visaFreeVisited = true;
        visaFreeFound = false;

        for (int i = 0; i < objDataSet.Tables[0].Rows.Count; i++) //Pasar para
la lista lo que se encuentra en el excel
        {
            if (process.citizenship ==
objDataSet.Tables[0].Rows[i].ItemArray[5].ToString())
            {
                string country =
objDataSet.Tables[0].Rows[i].ItemArray[1].ToString();
                string outlanderPass =
objDataSet.Tables[0].Rows[i].ItemArray[2].ToString();
                string cubanPass =
objDataSet.Tables[0].Rows[i].ItemArray[3].ToString();
                string authorizeTime =
objDataSet.Tables[0].Rows[i].ItemArray[4].ToString();
                string citizenship =
objDataSet.Tables[0].Rows[i].ItemArray[5].ToString();

                visaFree = new VisaFreeA(country, outlanderPass, cubanPass,
authorizeTime, citizenship);
                visaFreeFound = true;
                excelConection.Close();
                break;
            }
        }

        FreeVisaEventArgs eventarg = new FreeVisaEventArgs(instanceID, visaFree,
process, visaFreeVisited, visaFreeFound);
        this.FreeVisaChequed(this, eventarg);
    }
}

```

A continuación se muestra el resultado de la prueba desarrollada:

### Método de prueba

```

public void VisaFreeAgreemnetTest ()
{

```

```

        Person p=new Person("86042906177", 0, 0, 0 , "Yuleydis", "", "Pulido",
"Delgado",
        "", "", DateTime.Today, 'F', 0, 0, "", false, "", new Country("", "",
"Venezolana", DateTime.Today), new EyeColor("", DateTime.Today),
        new SkinColor("",DateTime.Today),new HearColor("",DateTime.Today),new
PersonImage(null,DateTime.Today),new OptionalDate("", "", "", "", "", "", "", ""),
        new Address(Guid.Empty, "", ""), new PersonType(DateTime.Today, ""), new
CivilRegister(0, "", DateTime.Today));

        VisaFreeAgreementRService target = new VisaFreeAgreementRService();

        Guid instanceID = Guid.NewGuid();

        VisaFreeA visaFree = new VisaFreeA("", "", "", "", "Venezolano");

        MigratoryProcess process = new MigratoryProcess(p, "464081",
DateTime.Today, DateTime.Today, (TravelReason)Enum.Parse(typeof(TravelReason),
        "Al_Turismo"), (VisaType)Enum.Parse(typeof(VisaType), "E1"), "entrada",
        (Modality)Enum.Parse(typeof(Modality), "ExtranjerosVisitantes"),
        (FileType)Enum.Parse(typeof(FileType), "Pasaporte"), DateTime.Today,
DateTime.Today, "", "", "Venezolano", "esp101");

        bool visaFreeVisited = false;

        bool visaFreeFound = false;

        target.VisaFreeAgreemnet(instanceID, visaFree, process, visaFreeVisited,
visaFreeFound);

    }

```

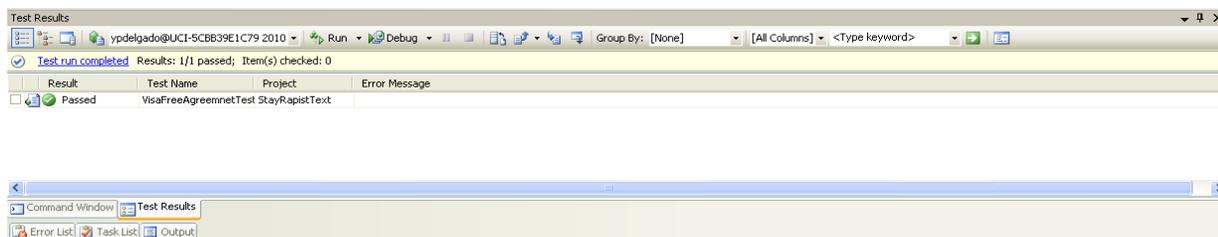


Figura 13. Resultado de la prueba unitaria al método VisaFreeAgreemnet.

### 4.6.3. Pruebas de sistema

**Prueba de sistema:** Verifica el sistema completo o su aplicación como tal. Se toma el punto de vista del usuario final y los casos de uso de prueba que ejecutan acciones típicas del usuario.

Las pruebas del sistema (pruebas de caja negra) se refiere a aquellas que se llevan a cabo sobre la interfaz del software o sea, los casos de prueba pretenden demostrar que las funciones del software son

operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. (SoftwareQATest.com., 2010)

Muchos autores, como Pressman, consideran que estas pruebas permiten encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Se realizaron 3 iteraciones de prueba al proceso Chequear pasajeros y tripulantes en la entrada o salida del país. Para mayor información sobre la descripción de las iteraciones de prueba Ver. Anexo XXX.

El gráfico que se muestra a continuación establece la relación entre las iteraciones de pruebas realizadas y las no conformidades detectadas.



**Figura 22. No conformidades encontradas en el proceso de Pruebas.**

En la figura anterior se pueden apreciar las no conformidades detectadas en las 3 iteraciones de pruebas realizadas al escenario “Introducir datos si el pasaporte no es de lectura mecánica”, demostrando la manera en que fueron disminuyendo las no conformidades, sirviendo esto para corregir los errores que existían. La solución de estas no conformidades ayudó a que la aplicación tuviera una mejor calidad y cumpliera con los requisitos exigidos por los clientes. Las no conformidades que se encuentren en los

demás escenarios se irán mitigando en la medida de que se realicen nuevas iteraciones de pruebas al módulo Chequeo.

### **4.6. Conclusiones**

En este capítulo se trataron temas relacionados con la implementación y las pruebas realizadas al sistema, creando los artefactos que propone la metodología. Se generaron los diagramas de Despliegue y de Componentes encargados de describir la distribución física de los nodos de un sistema y mostrar la organización y dependencias lógicas de los componentes del software respectivamente.

Como parte de las pruebas realizadas al módulo, se diseñaron los casos de pruebas a los requisitos funcionales y se desarrollaron pruebas unitarias y de sistema. Se describieron las iteraciones de prueba efectuadas y se registraron los defectos y dificultades encontradas.

## CONCLUSIONES

Con la realización del presente trabajo de diploma se desarrolló un módulo que perfeccionó el chequeo migratorio en la frontera aérea de la Dirección de Inmigración y Extranjería del Ministerio del Interior al brindar las funcionalidades definidas en los requisitos funcionales, permitiendo así chequear pasajeros y tripulantes en la entrada/salida del país, chequear pasaporte, asociar los pasajeros menores de edad a un mayor responsable, comprobar el convenio de libre visado y registrar una mayor información de los pasajeros en tiempo real.

Para lograr este objetivo se realizó un análisis del marco teórico de la investigación, posibilitando el sustento científico del trabajo. De igual forma se detallaron las herramientas, tecnologías y la metodología propuesta por el proyecto de Identificación, Inmigración y Extranjería. Así como la modelación de los procesos involucrados en el chequeo migratorio que se desarrollan en los aeropuertos internacionales del país y se definieron los requisitos funcionales y no funcionales que sustentaron la aplicación; generándose además, los artefactos definidos por la metodología utilizada y la realización de las pruebas unitarias y de sistema al módulo desarrollado.

## RECOMENDACIONES

Para un óptimo funcionamiento de la solución que se propone en esta investigación se recomienda al equipo de desarrollo:

- Continuar la implementación de las funcionalidades que son necesarias para dar cumplimiento a los requisitos del software.
- Integrar los componentes que necesita el módulo Chequeo para lograr un mayor control en los aeropuertos internacionales.
- Desarrollar los demás módulos del subsistema de control migratorio en la frontera Centinel e integrarlos al Sistema de Inmigración, Extranjería y Ciudadanía.
- Realizar el despliegue del módulo en los aeropuertos internacionales del país.
- Realizar pruebas unitarias y de sistemas al módulo una vez desplegado para proponer mejoras en las próximas versiones.

## BIBLIOGRAFÍA

**ASP.NET, Microsoft. 2010.** Microsoft ASP.NET. [En línea] 2010. <http://www.asp.net/es>.

**Danysoft | Haciendo visible lo invisible. 2009.** Altova UModel. [En línea] 2009. [Citado el: 05 de 12 de 2009.]  
[http://shop.danysoft.com/epages/danyshop\\_com.sf?ObjectPath=/Shops/danyshop\\_com/Products/%22Alto%20UModel%22/SubProducts/%22Altova%20UModel-0001%22](http://shop.danysoft.com/epages/danyshop_com.sf?ObjectPath=/Shops/danyshop_com/Products/%22Alto%20UModel%22/SubProducts/%22Altova%20UModel-0001%22).

**DanySoft. 2008.** DanySoft. [En línea] 2008. <http://www.danysoft.com/bol/visualstudio2008.htm..>

**Danysoft. 2009.** Embarcadero ER/Studio. [En línea] 2009. [http://www.codegear-shop.com/epages/62042259.sf/es\\_ES/?ObjectPath=/Shops/62042259/Products/%22Embarcadero%20ER/Studio%22](http://www.codegear-shop.com/epages/62042259.sf/es_ES/?ObjectPath=/Shops/62042259/Products/%22Embarcadero%20ER/Studio%22).

**Datys. 2009.** Datys, Tecnología y Sistemas. [En línea] 2009. <http://www.datys.cu/home.aspx>.

**Definición.de. 2008.** Definición de modelo de datos. [En línea] 2008. <http://definicion.de/modelo-de-datos/>.

**DesarrolloWeb.com . 2004.** Componentes de la plataforma .NET. [En línea] 26 de Octubre de 2004. [Citado el: 10 de Diciembre de 2009.] <http://www.desarrolloweb.com/articulos/1681.php>.

**garbagecollector. 2004.** Sistema Gestor de base de datos SGBD. [En línea] 01 de Noviembre de 2004.  
[http://www.error500.net/garbagecollector/archives/categorias/bases\\_de\\_datos/sistema\\_gestor\\_de\\_base\\_de\\_datos\\_sgbd.php](http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php).

**Gracia, Joaquin. 2005.** CMM - CMMI Nivel 2. [En línea] 26 de 11 de 2005. [Citado el: 05 de 12 de 2009.] <http://www.ingenierossoftware.com/calidad/cmm-cmmi-nivel-2.php>.

—. **2005.** Patrones de diseño. [En línea] 27 de mayo de 2005.  
<http://www.ingenierossoftware.com/analysisydiseno/patrones-diseno.php>.

—. **2004.** Seguimiento y gestión de errores, Bug Tracking. [En línea] 18 de septiembre de 2004. [Citado el: 10 de febrero de 2010.] <http://www.ingenierossoftware.com/pruebas/gestionerrores.php>.

**Grady Booch, Jim Rumbaugh e Ivar Jacobson. 2004.** UML, El Lenguaje Unificado de Modelado. [En línea] 28 de 10 de 2004. [Citado el: 20 de 11 de 2009.] <http://elvex.ugr.es/decsai/java/pdf/3E-UML.pdf>.

**Hooping.net. 2008.** Diseño de páginas web. *Glosario*. [En línea] 2008. [Citado el: 10 de diciembre de 2009.] <http://www.hooping.net/glossary/aplicaciones-web-146.aspx>.

- Jacobson, I. y Rumbaugh, G. Booch y J. 2000.** *El Proceso Unificado de Desarrollo de software.* 2000.
- José Antonio González Seco.** Tutorial C#. [En línea] <http://www.clikear.com/manuales/csharp/c10.aspx>.
- Jose Luis Olivares, BS-Buffer Sistemas.** Windows Workflow Foundation, Manejo de procesos en los nuevos productos de Microsoft. [En línea] [Citado el: 05 de abril de 2010.] <http://www.mossca.org/Documentos%20compartidos/DESARROLLO%20%28Development%29/Workflow%20>.
- JOURMOLY. 2008.** Sistemas de control de versiones. [En línea] 28 de marzo de 2008. <http://www.jourmoly.com.ar/debes-usar-un-sistema-de-control-de-versiones/>.
- Keeton, Marlys. 2006.** Microsoft Solutions Framework (MSF): A Pocket Guide. s.l. [En línea] 2006.
- Landa, Ibon. 2009 .** Yo sólo pasaba por aquí pero ya que estoy... [En línea] 10 de 3 de 2009 . <http://geeks.ms/blogs/ilanda/archive/tags/Team+System/default.aspx>.
- larevistainformatica.com. 2006.** *Lenguaje de Programación C#.* [En línea] 2006. <http://www.larevistainformatica.com/C1.htm>.
- Latinus.sa. 2009.** Business Process Modeling Notation (BPMN) . [En línea] 2009. [Citado el: 05 de 12 de 2009.] <http://www.latinus.net/paginas/bpmnBpel.html>.
- Mañas, José A. 1994.** *Prueba de Programas.* . 1994.
- Microsoft Corporation. 2006.** The ADO.NET Entity Framework Overview. [En línea] Junio de 2006. [http://msdn.microsoft.com/en-us/library/aa697427\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/aa697427(VS.80).aspx).
- Microsoft. 2010.** Visual Studio Team Foundation Server. [En línea] 2010. <http://emea.microsoftstore.com/es/es-ES/Microsoft/Visual-Studio-Team-Foundation-Server-2010-%28CAL-de-usuario%29>.
- MKM Publicaciones. 2008.** Microsoft Visual Studio 2008. [En línea] 5 de mayo de 2008. [Citado el: 05 de diciembre de 2009.] <http://www.mkm-pi.com/mkmpi.php?article2155>.
- Mossca.rog. . 2010.** Mossca.rog. [En línea] 2010. <http://www.mossca.org/>.
- 2009.** *MSF for CMMI.* 2009.
- Oracle Corporation. 2009.** Oracle. [En línea] 2009. <http://www.oracle.com/technology/products/database/oracle11g/index.html>.
- Pérez, Juan Diego. 2009.** *Notaciones y lenguajes de procesos. Una visión global.* Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla. Sevilla : s.n., 2009. pág. 100.

- producingoss.com. 2009.** Control de Versiones, Capítulo 3. Infraestructura Técnica. [En línea] 2009. <http://producingoss.com/es/vc.html>.
- Proyecto Identidad, Inmigración, Extranjería Cuba. 2009.** *Estándares de Codificación*. Ciudad de la Habana : UCI, 2009.
- Proyecto Identificación, Inmigración y Extranjería. 2009.** Bison Framework. *Ciudad de la Habana*. [En línea] 2009.
- . **2010.** *ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE*. UCI. Ciudad de la Habana, Cuba : s.n., 2010. pág. 111.
- Proyecto identificación, Inmigración y Extranjería. 2009.** *REGLAS DEL NEGOCIO*. UCI. Ciudad de la Habana, Cuba : s.n., 2009. pág. 27.
- Saldaña, Iván. 2004.** EsMas. [En línea] Noticieros Televisa, 28 de julio de 2004. [Citado el: 05 de 12 de 2009.] <http://www.esmas.com/noticierostelevisa/mexico/380786.html>.
- Santamaría, Pedro. 2009.** Servicios del sistema: ¿Qué son? [En línea] 11 de abril de 2009. [Citado el: 10 de marzo de 2010.] <http://www.applesfera.com/applesfera-responde/applesfera-responde-servicios-del-sistema-que-son>.
- Software. [En línea] Danysoft. [Citado el: 18 de enero de 2010.] [http://www.codegear-shop.com/epages/62042259.sf/es\\_ES/?ObjectPath=/Shops/62042259/Products/%22Embarcadero%20ER/Studio%22](http://www.codegear-shop.com/epages/62042259.sf/es_ES/?ObjectPath=/Shops/62042259/Products/%22Embarcadero%20ER/Studio%22).
- SoftwareQATest.com. 2010.** SoftwareQATest.com. [En línea] 10 de Enero de 2010. <http://www.softwareqatest.com>.
- Solus S.A. 2007 .** Guía de Usuario de Enterprise Architect 7.0. [En línea] 2007 . <http://www.sparxsystems.com.ar/download/ayuda/index.html?businessmodelling.htm>.
- Turner, Michael S. V. 2006.** Microsoft Solutions Framework Essentials: Building Successful Technology Solutions. [En línea] 2006. <http://www.microsoft.com/mspress/books/10250.aspx>. als: Building Successful Technology Solutions. ISBN 0-7356-2353-8.
- Weitzenfeld, A. 2005..** *Ingeniería de software orientada a objetos con Java e Internet*. Thomson Paraninfo. 2005. pág. 678. 2005. pág. 678.
- Zoftwar. 2006.** Requisitos No Funcionales - Parte 1. [En línea] 16 de noviembre de 2006. <http://zoftwar.blogspot.com/2006/11/requisitos-no-funcionales-parte-1-con.html>.

## GLOSARIO DE TÉRMINOS.

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

**API:** En español: Información Adelantada de Pasajeros.

**BPMN** (Business Process Management Notation): Notación para el Modelado de Procesos de Negocio. Permite el modelado de asuntos de negocio donde se representan gráficamente las diferentes etapas del proceso. Esta notación coordina la secuencia de eventos que fluyen entre los diferentes procesos y participantes.

**CENATAV:** Centro de Automatización de Tecnología Avanzada.

**DESPACHO O CHEQUEO MIGRATORIO:** Es el acto donde se decide la aprobación del movimiento migratorio de entrada y/o salida del país.

**DIE:** Dirección de Inmigración y Extranjería.

**ECASA:** Empresa Cubana de Aeropuertos y Servicios Aeronáuticos.

**FLUJO DE TRABAJO (Workflow en inglés):** Sistemas que definen, gestionan, controlan y ejecutan flujos de trabajo en el contexto de procesos de negocio, a través de la ejecución de software, cuyo orden de ejecución es controlado por una representación computarizada del proceso de negocio.

**HTTPS:** versión segura del protocolo HTTP que implementa un canal de comunicación seguro y basado en SSL (*Secure Socket Layers*) entre el navegador del cliente y el servidor HTTP.

**IDE:** Integrated Development Environment (Ambiente de Desarrollo Integrado)

**MININT:** Ministerio del Interior. Dentro del estado cubano, cumple con tareas de seguridad y orden interior.

**MSF:** Microsoft Solution Framework, marco de trabajo para soluciones de Microsoft.

**RF:** Requerimiento Funcional.

**RNF:** Regla de Negocio de Frontera.

**RnF:** Requerimiento no Funcional.

**Saime:** Sistema Administrativo de Identificación, Migración y Extranjería de la República Bolivariana de Venezuela.

**SGBD:** Sistema gestor de base de datos.

**TED:** Tarjetas de Embarque y Desembarque

**TNS** (Transparent Network Substrate, Sustrato de red transparente) es una capa de comunicación que utilizan las bases de datos Oracle. TNS Service Name es el nombre por el que se conocen las instancias de una base de datos Oracle en una red. Se asigna un nombre a este servicio cuando se configura la conectividad de la base de datos Oracle. La replicación utiliza el nombre del servicio TNS para identificar al publicador y establecer conexiones.

**VISA:** Permiso de salida que emiten los países con acuerdos migratorios.

**.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.