

Universidad de las Ciencias Informáticas

Facultad 1



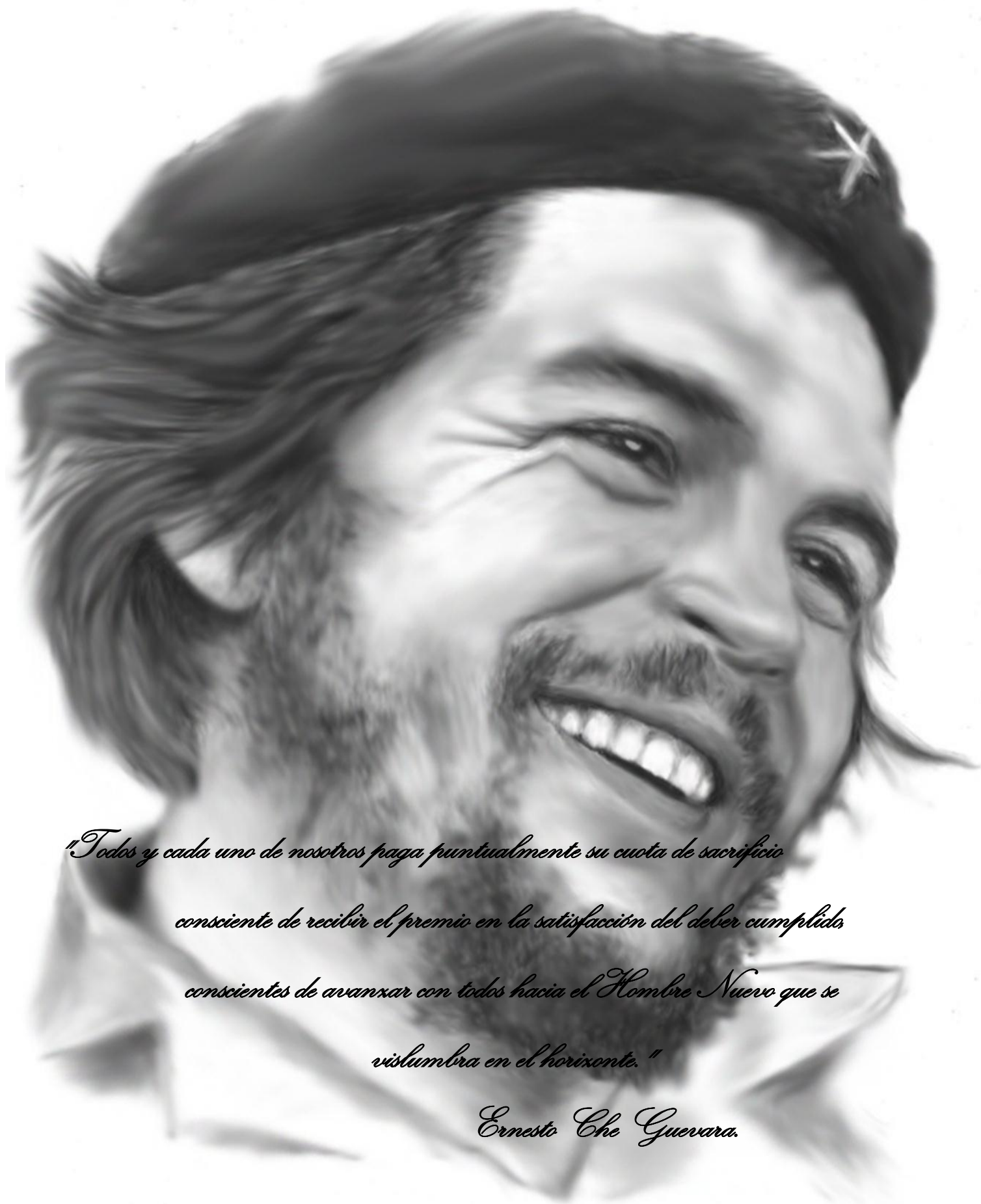
“Flujo de trabajo para personalizar documentos electrónicos de viaje usando el Sistema de personalización del Centro de Identificación y Seguridad Digital”

Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informática

Autores: Yaumara Cruzata Negret
Carlos Alexis Hijuelo García

Tutores: Ing. Dannier Sierra Obregón
Ing. Alexander López Pupo
Co-tutor: Msc. Rubén Cruzata Santos

“La Habana, junio 2013”.



*"Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio
consciente de recibir el premio en la satisfacción del deber cumplido
conscientes de avanzar con todos hacia el Hombre Nuevo que se
vislumbra en el horizonte."*

Ernesto Che Guevara.

Declaración de Auditoría

Declaración de Auditoría

Manifestamos de esta manera ser los autores de la presente tesis y reconocemos a la Universidad de las Ciencias informáticas el derecho a hacer uso de la misma para su beneficio.

Para dejar constancia de lo antes planteado firmamos la presente a los ____días del mes ____del año_____.

Autores

Yaumara Cruzata Negret

Carlos Alexis Hijuelo García

Tutores

Ing. Dannier Sierra Obregón

Ing. Alexander López Pupo

Co-Tutor

Msc. Rubén Cruzata Santos

Síntesis de los Tutores

Ing. Dannier Sierra Obregón: Ingeniero en Ciencias Informáticas. Actualmente se encuentra vinculado al Departamento Tarjetas Inteligentes del Centro de Identificación y Seguridad Digital de la Universidad de las Ciencias Informáticas.

Ing. Alexander López Pupo: Ingeniero en Ciencias Informáticas. Actualmente se encuentra vinculado al Departamento Tarjetas Inteligentes del Centro de Identificación y Seguridad Digital de la Universidad de las Ciencias Informáticas.

Agradecimientos

Yaumara

A quienes considero las mejores personas del mundo, mis padres, por acompañarme en los momentos alegres y tristes, por cada noche que abandonaron el sueño tratando de encontrarle una solución a mis problemas, por haberme guiado día a día hasta llegar aquí. Por demostrarme cada día lo importante que soy en sus vidas. A mi bebe como cariñosamente llamo a mi Julio, quien hace mucho tiempo dejó de ser mi novio para convertirse en alguien que no alcanzo a describir con dos líneas, a ti mi amor te agradezco tu espera y total comprensión. A mi compañero de tesis por hacerme mucho caso y no fajarse conmigo, a los tutores Danniell y Alexander y el co-tutor Rubén, por ayudarnos en la realización del trabajo. A mis hermanos a quienes sin dudas no los cambiaría por nada, pues son insustituibles para mí, a mi cuñi a quien considero otro hermano más. A toda mi familia por su incondicional apoyo en especial a mi abu. A los profesores que me impartieron clases en los 5 años de la universidad en especial a alguien a quien admiro mucho y me brindó su apoyo Alíen. A todos mis compañeros de grupo que aguantaron e sus turnos a una coti, como cariñosamente me dicen muchos de ellos, a los del apartamento, en especial a las cheas Liliet y Aliané, quienes me aguantaron mis malos días y mis cheancias. Mis agradecimientos a cada persona que me apoyó en momentos en que los ánimos no eran los apropiados para poder lograr mi objetivo. A todos los que han estado a mi lado de una forma u otra en algún momento de mi vida, legándome algún consejo o algún gesto de apoyo, para todos mis agradecimientos.

Carlos

Pienso que ésta sección, quizás, no sea suficiente para mencionar a todos aquellos que han puesto un poquito de empeño y su ayuda desinteresada para cumplir éste, mi sueño, y si existe alguien que no sienta su presencia en mis palabras, reciba mis más sinceras disculpas y tenga la certeza que siempre tendrá mi gratitud. Primero que todo agradecer a mis abuelos que me

Agradecimientos

llevaron de la mano mi primer día de escuela y, de cerca o de lejos, han cuidado cada uno de mis pasos hasta aquí. Por el esfuerzo que hacen para que yo sea una mejor persona, le estoy eternamente agradecido y algún día les retribuiré todo lo que han hecho por mí. A mi mamá y padrastro por siempre confiar y creer en mí, por darme todo en mí y estar presente en los momentos necesarios para que yo pudiera lograr mis sueños. A mis primas y primo Gisel, Yanet, Maria Karla y Ernesto por hacer que mi vida se llene de felicidad y estar presente en los momentos más importantes y apoyarme en las decisiones que he tomado a lo largo de esta carrera universitaria. A mi primita Isa que ha sido como la hermanita que no tuve, que ha sido mis ojos por los cuales veo, a pesar de su poca experiencia y edad ha sabido enseñarme y ser como una conciencia en mis acciones. A mi tío Guido por haber sido en sus tiempos de vida mi primer maestro y por haberme soportado las primeras malacrianzas y haberme llenado con sus conocimientos una gran parte de mí. A mis tías y tíos Niurca, Aleida, Esther, Carlitos y Juan Carlos por tener que cargar conmigo en estos últimos años, por su apoyo brindado en la realización de este trabajo y por ayudarme a convertirme en un mejor ingeniero. A mi novia Yaneisys por haber sido más que una novia, haber sido una compañera inseparable, por haberme soportado mis malacrianzas y por haber estado a mi lado en los momentos más difíciles por los que he pasado en esta universidad. Te kiero. A mis tutores Alexander y Dannier les agradezco la ayudada que me han dado para perfeccionar el trabajo de diploma, muchas gracias de verdad. A mis compañeros de clase y apartamento, especialmente a los que estuvieron cerca durante estos cinco años. De manera general a todos que de una manera u otra han estado presentes en mi vida, por todo el apoyo y la ayuda prestada muchas gracias.

Dedicatoria

Yaumara

Primero a mis mayores tesoros quienes se deben sentir igual de emocionados que yo, mis padres, por su comprensión y apoyo incondicional, por confiar en mí. A Julio Iglesias, mi novio quien representa para mi mucho más que eso, es una las personas que llegó a mi vida para quedarse. A mi hermano menor Yohysnel (yoyo), mi príncipe enano que me ve como un ejemplo a seguir, a mi cuñado Alexay y mi hermana Ana Maris ,que sin ellos hubiera resultado un tanto difícil mis últimos año en la universidad y a quien probablemente nunca le dicho que la quiero mucho. A mi familia: mis abuelos(as), mis tíos(as), mis primos(as), mis amistades, y especial a los dos pequeños gigantes que mi hermana me dio por sobrinos, porque sin ellos la vida ya me resultaría la misma, a los vecinos que apoyaron, en especial a mi madrina y mi suegra a quienes nunca les faltó una frase de aliento para mí, quiero agradecer también a quien continua siendo para mi mejor amiga, Dayana y a su familia a quienes considero como mi familia también. A mis amistades que aunque hoy no están aquí en la universidad no han dejado de preocuparse por mí y me continuaron apoyando.

Carlos

A mis abuelos por haberme criado y haber sido mis guías en este camino tan duro que es la vida, mi mamá por haberme traído a este mundo y haberme encaminado en la difícil tarea de crecer y hacerme un hombre de bien y a mi padrastro Humberto por haber sido como un padre para mi desde mis primeros años. Ustedes son el motor impulsor que me renuevan las fuerzas cuando estas parecen haberse agotado. Son mi razón de ser, mi mayor orgullo, mi fuente de inspiración. Les debo todo lo que soy y seré. El haber llegado hasta este punto es fruto de su dedicación y esfuerzo. Muchas gracias mami y papi. Los amo.

Resumen

El presente trabajo recoge los antecedentes y aspectos teóricos y conceptuales relacionados con los documentos electrónicos de viaje y su personalización, centrando la investigación en la personalización de los pasaportes electrónicos en Cuba y el mundo. Se analiza la situación del Sistema de Personalización de Documentos de Identificación (SPDI) que existe en el Centro de Identificación y Seguridad Digital (CISED), con relación a la personalización de pasaportes electrónicos. Abordándose las características actuales de dicho sistema y descartando el problema de este para personalizar documentos como los pasaportes electrónicos. Se propone desarrollar un flujo de trabajo (*workflow*), con el fin de llevar a cabo la personalización de pasaportes electrónicos, de modo que este cumpla con las normas establecidas por la Organización de Aviación Civil Internacional (OACI), y así poder proporcionar mayor funcionalidad al SPDI teniendo en cuenta los avances tecnológicos producidos en los últimos años en los pasaportes. Este documento recoge los resultados de las investigaciones realizadas, donde se describe el diseño de la solución a desarrollar, la arquitectura y los artefactos generados en el proceso de desarrollo. La cual está guiada por la metodología ágil, *XP* por ajustarse a las condiciones del equipo, *Altova UModel* como herramienta de modelado, *.NET Framework 4.0* como plataforma de desarrollo, *Microsoft Visual Studio 2010* como entorno de desarrollo, *Windows Communication Foundation*, *Windows Workflow Foundation* y *C#* como lenguaje de programación.

Palabras Claves: personalización de documentos, pasaporte electrónico, flujo de trabajo.

Índice

Declaración de Auditoría.....	I
Síntesis de los Tutores.....	II
Agradecimientos	III
Dedicatoria.....	IV
Resumen	V
Introducción	1
Capítulo 1.Fundamentación teórica	5
1.1 Introducción.....	5
1.2 Conceptos básicos asociados	5
1.2.1 Workflow	5
1.2.2 Personalización.....	6
1.2.3 Documento electrónico	6
1.2.4 Middleware	6
1.2.5 Applet.....	7
1.3 Definición de Smartcards.....	7
1.3.1 Seguridad de las Smartcards	7
1.3.2 Las Smartcards y su relación con los pasaportes electrónicos.....	8
1.4 Pasaportes electrónicos	8
1.4.1 Organización de los Datos de los pasaportes por Zonas	10
1.4.2 La protección de datos del Pasaporte electrónico	12
1.5 Estándares relacionados con los pasaportes.....	13
1.5.1 Documento de Viaje de Lectura Mecánica (DVLM) 9303	13
1.5.2 ISO 7816.....	13
1.5.3 ISO 14443.....	14

1.5.4 PC/SC.....	14
1.6 Escritura electrónica.....	14
1.6.1 Personalización.....	15
1.6.2 Proceso de Personalización de Documentos electrónicos	15
1.6.3 Organización del proceso de personalización de documentos	16
1.7 Análisis de soluciones existentes	16
1.7.1 Nivel Nacional.....	16
1.7.2 Nivel Internacional	17
1.8 Tecnologías, herramientas y metodología utilizadas.	18
1.8.1 Metodología de desarrollo.....	18
1.8.2 Extreme Programming (XP)	19
1.8.3 Lenguaje de modelación visual	21
1.8.4 Altova UModel.....	21
1.8.5 Lenguaje de programación.....	22
1.8.6 .NET Framework 4.0	23
1.8.7 Windows Communication Foundation	23
1.8.8 Windows Workflow Foundation (WWF).....	24
1.8.9 Bison Framework	24
1.8.10 Visual Studio .NET	25
1.8.11 Developer Suite Gemalto	25
1.9 Conclusiones parciales.....	25
Capítulo 2. Propuesta de Solución.....	26
2.1 Introducción.....	26
2.2 Propuesta de Solución	26
2.2.1 Modelo de Dominio	26
2.2.2 Glosario de conceptos del modelo de dominio	27

2.2.3 Actores del Sistema.....	29
2.2.4 Especificación de Requisitos.....	30
2.2.5 Historias de usuario.....	31
2.2.6 Metáfora.....	35
2.3 Arquitectura.....	35
2.4 Patrones de Diseño.....	36
2.4.1 Patrones Workflow.....	37
2.5 Diagrama de Clases del Diseño.....	38
2.6 Estimación de Tiempo.....	40
2.7 Plan de Iteraciones.....	40
2.8 Plan de Entrega.....	41
2.9 Tarjetas CRC.....	41
2.10 Conclusiones parciales.....	43
Capítulo 3. Implementación y Prueba.....	44
3.1 Introducción.....	44
3.2 Estándares de codificación.....	44
3.3 Tratamiento de errores.....	45
1.4 Vista General del Workflow.....	45
3.5 Diagrama de Componentes.....	47
3.6 Diagrama de Despliegue.....	48
3.7 Fase de producción.....	50
3.8 Pruebas Unitarias.....	50
3.8.1 Prueba de Caja Blanca o Estructurales.....	50
3.8.2 Diseño de Casos de prueba.....	51
3.9 Pruebas Funcionales.....	52
3.9.1 Pruebas de Caja Negra.....	52

3.9.2 Diseño de Casos de prueba.....	53
3.9.3 Resultado de las pruebas.....	54
3.11 Conclusiones Parciales	55
Conclusiones Generales.....	56
Recomendaciones	57
Bibliografía Referenciada.....	58
Glosario de Términos.....	62
ANEXOS.....	64
ANEXO 1.Imágenes de Datos OCR.	64
ANEXO 2. Descripción de la Arquitectura del sistema SPDI.....	65
ANEXO 3.Diagrama de Clases.....	67
ANEXO 4.Tarjetas CRC.	68
ANEXO 5.Vista General del Workflow. Parte1	70
ANEXO 6.Vista General del Workflow. Parte2.....	71

Índice de Figuras

Figura 1.Estructura del chip.	9
Figura 2.Símbolo del pasaporte-e	10
Figura 3.Orden de los datos en la Hoja de Datos.....	10
Figura 4.Construcción de la página de datos del pasaporte.....	11
Figura 5.Ejemplo de escritura electrónica.	15
Figura 6.Fases de un proyecto con XP	20
Figura 7.Flujo del sistema.	26
Figura 8.Modelo de Dominio	27
Figura 9.Arquitectura del Sistema.	35
Figura 10.Ejemplo del Patrón Creador	36
Figura 11.Ejemplo1 de uso de patrones de control de flujos básicos	37
Figura 12.Ejemplo2 de uso de patrones de control de flujos básicos	38
Figura 13.Ejemplo del patrón de ramificación avanzada y sincronización	38
Figura 14.Diagrama de Clases del Diseño	39
Figura 15.Diagrama de la Vista General de Workflow	46
Figura 16.Diagrama de Componente	47
Figura 17.Diagrama de Despliegue.....	49
Figura 18.Ejemplo1 de Datos OCR.....	64
Figura 19.Ejemplo2 de Datos OCR.....	64
Figura 20.Diagrama de Clases con los métodos	67
Figura 21.Vista General del Workflow parte1	70
Figura 22. Vista General del Workflow parte2	71

Índice de Tablas

Tabla 1.Componentes de la ICP (24)	9
Tabla 2.Método de Seguridad básico de los datos electrónicos en el DVLM (26).	12
Tabla 3.Metodos de Seguridad avanzados de los datos electrónicos en el DVLM (26).....	12
Tabla 4.Tabla comparativa de los dos grupos de Metodologías (37).....	19
Tabla 5.Actores del Sistema.	29
Tabla 6.Requisitos no Funcionales	31
Tabla 7.Gestión de la Comunicación.....	31
Tabla 8.Permitir la autenticación	32
Tabla 9.Cargar e instalar el <i>Applet</i>	32
Tabla 10.Creación de los DG	33
Tabla 11.Creación del DG1.....	33
Tabla 12.Creación del DG2.....	34
Tabla 13.Creación del SOD	34
Tabla 14.Estimación de Tiempo de las Historias de Usuario.....	40
Tabla 15.Plan de iteraciones.	41
Tabla 16.Plan de Entregas.....	41
Tabla 17.Tarjeta CRC de la clase SmartCardReader.Client.cs	42
Tabla 18. Tarjeta CRC de la clase ISmartCardReader.Controller.cs	42
Tabla 19. Tarjeta CRC de la clase SmartCardReader.cs	42
Tabla 20. Tarjeta CRC de la clase SPDIWorkflowOrder.....	42
Tabla 21. Tarjeta CRC de la clase SPDIWorkflowRequest.	43
Tabla 22. Tarjeta CRC de la clase IEngraveElectricService.....	43
Tabla 23. .Tarjeta CRC de la clase EngraveElectricService.....	43
Tabla 24.Estándares de codificación.....	45
Tabla 25.Método CreateSecureChannelTest al que se le realizó la prueba unitaria.....	51
Tabla 26.Método TransmitTest al que se le realizó la prueba unitaria.....	52
Tabla 27. Diseño de Caso de Prueba del requisito funcional "Gestionar Personalización".....	54
Tabla 28.Tabla de no Conformidades	55
Tabla 29.Tarjeta CRC de la clase ISmartCardReaderService.....	68
Tabla 30.Tarjeta CRC de la clase SPDIWorkflowSmartCardReaderPersonalization.....	68

Índice de Tablas

Tabla 31. Tarjeta CRC de la clase SmartCardReaderService.....	69
Tabla 32.SmartCardReaderService CPDI.Divices.SmartCardReadersHttpClientWrapper.	69

Introducción

A través de los siglos el hombre ha incursionado en el mundo de lo desconocido, buscando respuestas a los fenómenos que hasta ese momento no tienen una explicación científica o simplemente buscando la optimización de procesos en beneficio social o económico (1). Debido a esto con el transcurso del tiempo se hizo necesario llevar el control de las personas alrededor del mundo.

En función de mejorar el proceso de control de viajes internacionales, la Organización de Aviación Civil Internacional (OACI), creada en 1944 con el fin de hacer más seguro y fácil viajar de una nación a otra (2). Establece un conjunto de normas y regulaciones necesarias para asegurar la eficiencia, seguridad y regularidad del transporte aéreo. Introdujo además requisitos de viaje para garantizar la integridad de la información personal y las relaciones entre los pueblos (3) (4).

El proceso de identificación de las personas foráneas residentes, se ha convertido en un asunto de gran relevancia para lograr mantener identificados a los ciudadanos. Lo anterior, unido al progresivo aumento de la falsificación de identidades y la ocurrencia de actos terroristas, hacen la utilización de las nuevas tecnologías una necesidad vigente en las diferentes etapas de la sociedad actual. El pasaporte constituye sin dudas un documento de identificación a nivel internacional que no está excusado de fraudes, por la seguridad que este documento confiere a los estados (5).

Fue entonces que en 1998, que el grupo de trabajo sobre nuevas tecnologías del Grupo técnico asesor sobre documentos de viaje de lectura mecánica (*TAG/MRTD*) comenzó su labor para establecer el sistema de identificación biométrica más efectivo y los correspondientes medios de almacenamiento de datos para emplear en aplicaciones de documentos de viaje de lectura mecánica (DVLM), concibiéndose de esta manera un nuevo documento, el pasaporte electrónico (pasaportes-e¹). Pero no fue hasta después del 11 de septiembre del 2001, tras el atentado a las Torres Gemelas que los estados asignaron mayor importancia a la seguridad de los documentos de viaje y a la identificación de su titular (6).

El uso de los pasaportes electrónicos se ha ido extendiendo en los últimos tiempos, por lo que se han definido diversos estándares a seguir por los estados que utilicen este tipo de documento, permitiendo así una interoperabilidad² entre todos los sistemas que sigan las especificaciones (7). Actualmente se han desarrollado numerosas soluciones para el manejo de documentos de viaje, dentro de ellas se encuentra el Sistema de Personalización de Documentos de Identificación SPDI, desarrollado en la Universidad de las Ciencias Informáticas (UCI) en el Centro de Identificación y Seguridad Digital (CISED). Es un sistema

¹Conocido también como pasaporte biométrico, es un documento de identidad que contiene un circuito electrónico, se concentra en los aspectos biométricos con relación a los pasaportes de lectura mecánica. Poseen ineficiencia mundial.

² Es la habilidad que posee un sistema o producto que permite interconexión y compatibilidad con otros sistemas.

Introducción

basado en flujos de trabajo que permite llevar a cabo la personalización e impresión de importantes documentos de identidad (carné de identidad y pasaporte). Para ello cuenta con 5 módulos fundamentales: administración, supervisión, impresión, calidad y embalaje que posibilitan el procesamiento organizado y semi-automatizado de grandes solicitudes de impresión de documentos de identidad, este sistema se encuentra desplegado en todo el territorio nacional (8). Hasta este momento sólo personaliza aquellos documentos de lectura mecánica, siendo ésta la limitante fundamental teniendo en cuenta que se ha incrementado el uso de documentos con chip y para este caso específico el SPDI no está capacitado para realizar la personalización de estos tipos de documentos, impidiendo que se pueda llevar a cabo la personalización de documentos como el pasaporte electrónico.

A fin de dar solución a la situación anteriormente expuesta, se plantea el siguiente **problema de investigación** ¿Cómo facilitar el proceso de personalización de documentos electrónicos de viaje de acuerdo a los estándares internacionales establecidos?

Definiéndose como **Objeto de estudio**: El proceso de personalización de documentos electrónicos de viaje. Enmarcado en el **Campo de acción**: El proceso de personalización de documentos electrónicos de viaje basado en flujo de trabajo.

Objetivo General: Desarrollar un flujo de trabajo para la personalización de documentos electrónicos de viaje usando el Sistema de Personalización del Centro de Identificación y Seguridad Digital, de manera que cumpla con los estándares internacionales establecidos.

Objetivos Específicos:

- ✓ Realizar un estudio de soluciones existentes, tecnologías, herramientas y metodología para el proceso de personalización de documentos electrónicos de viaje.
- ✓ Diseñar el flujo de trabajo para el proceso de personalización de documentos electrónicos de viaje.
- ✓ Implementar el flujo de trabajo para la personalización de documentos electrónicos de viaje.
- ✓ Probar el flujo de trabajo, haciendo uso del Sistema de Personalización del centro de Identificación y Seguridad Digital para un *Applet* determinado.

Tareas de la Investigación

- ✓ Análisis del Sistema de Personalización del Centro de Identificación y Seguridad Digital y otros sistemas tanto en el ámbito nacional como internacional. (Yaumara Cruzata Negret y Carlos Alexis Hijuelo García)
- ✓ Análisis de la tecnología *Windows Workflow Foundation 3.5* (Carlos Alexis Hijuelo García)

Introducción

- ✓ Análisis del *Framework* sobre el que se van a desarrollar los flujos de trabajo de personalización del chip. (Yaumara Cruzata Negret y Carlos Alexis Hijuelo García)
- ✓ Descripción de las herramientas, tecnologías y metodologías de desarrollo seleccionadas para el desarrollo del sistema. (Yaumara Cruzata Negret y Carlos Alexis Hijuelo García)
- ✓ Diseño del flujo de trabajo para la personalización del chip, y servicios de negocio, canal seguro, preparación de los datos que generan los grupos de datos y firma del *SOD*. (Yaumara Cruzata Negret y Carlos Alexis Hijuelo García)
- ✓ Implementación de los servicios de negocio para el canal seguro y preparación de los datos que generan los grupos de datos. (Carlos Alexis Hijuelo García)
- ✓ Implementación del flujo de trabajo para la personalización del chip. (Carlos Alexis Hijuelo García)
- ✓ Realización de pruebas de unidad. (Carlos Alexis Hijuelo García)
- ✓ Realización de pruebas funcionales mediante la personalización de un *Applet JavaCard*. (Yaumara Cruzata Negret)

Métodos científicos que sustentarán la investigación:

Teórico:

El método **Histórico-Lógico** permitió adquirir conocimiento sobre la evolución y el desarrollo de los distintos elementos asociados a las tecnologías de tarjetas inteligentes y los pasaportes electrónicos así como el proceso de personalización de este último.

El método **Analítico-Sintético** permitió la comprobación y utilización de fuentes bibliográficas determinadas para sintetizar la información importante.

Se utilizó el método de la **Modelación** para crear diferentes elementos a lo largo de la investigación, como los diagramas de dominio, diagrama de componente y diagrama de despliegue, los cuales constituyeron una parte importante en la comprensión y simplificación del trabajo.

Empírico:

La Entrevista: mediante este método se realizaron entrevistas a personas con un grado considerable de conocimiento y experiencia respecto al tema y mediante el cual se logró obtener más información y se solidificó lo investigado anteriormente.

Justificación de la Investigación:

La UCI cuenta con un sistema de personalización en el CISED, que permite personalizar documentos de lectura mecánica. Debido al avance de las tecnologías en los pasaportes se ha detectado la necesidad de contar con un sistema que sea capaz de personalizar documentos electrónicos de viaje. Para lo cual, lo más conveniente sería la realización de un flujo de trabajo que pueda desarrollar dicho proceso y que pueda ser integrado al SPDI, para lograr obtener un sistema capaz de personalizar todo tipo de documento de viaje. Con dicho flujo para personalizar documentos electrónicos de viaje se estaría contribuyendo a mejorar la confirmación de la identidad de las personas pues se podrán incluir los datos en el chip del pasaporte a partir de su personalización.

Posibles resultados:

Se obtendrá un flujo de trabajo para la personalización documentos electrónicos de viaje, según los estándares internacionales establecidos por la Organización de Aviación Civil Internacional, a partir del Sistema de Personalización del Centro de Identificación y Seguridad Digital. Además la documentación asociada al flujo de trabajo según la metodología de desarrollo seleccionada.

El presente trabajo se encuentra compuesto por 3 capítulos desarrollado a partir del análisis realizado a la información bibliográfica existente. A continuación de la descripción correspondiente a cada uno de ellos:

Capítulo 1. Fundamentación teórica

El capítulo abarca una base teórica para comprender el problema antes mencionado. Asimismo se muestran las principales características de un pasaporte electrónico, algunos de los sistemas relacionados con el tema, así como las herramientas y tecnologías a utilizar en la solución.

Capítulo 2. Propuesta de solución

El capítulo contiene los artefactos definidos por la metodología *XP* para dar solución al problema científico. Se brinda la descripción de la propuesta de solución. De los artefactos logrados, entre los más significativos están las historias de usuarios y el modelo de domino. Se muestran también los requerimientos funcionales y no funcionales.

Capítulo 3. Implementación y Prueba

En este último capítulo se da cumplimiento a los planes trazados a través de las fases: Iteración y Producción, se realizan las pruebas unitarias y funcionales a la solución diseñada e implementada.

Capítulo 1. Fundamentación teórica

Capítulo 1. Fundamentación teórica

1.1 Introducción

El mundo tecnológico se encuentra en constante evolución y la tecnología de tarjetas inteligentes (*Smartcards*)³ es una de las más usadas actualmente. Diversas son las características que definen dicha tecnología de ahí su relación con los documentos de identidad personal, específicamente el pasaporte electrónico (9).

En el presente capítulo se muestra el estudio realizado referente a los pasaportes, exclusivamente a los pasaportes electrónicos, en el cual se exponen variados aspectos de ellos como: el modo de organización de sus datos. Además se hace alusión a varios aspectos de la tecnología de *Smartcards*, y la relación de estas con los pasaportes.

1.2 Conceptos básicos asociados

En el siguiente epígrafe se realizará una explicación de varios conceptos relacionados con el tema, que ayudaran a la comprensión del mismo.

1.2.1 Workflow

Entendido como el flujo de procesos administrativos o de negocio. Es el conjunto de actividades o tareas realizadas en secuencia o en paralelo por dos o más miembros de un equipo de trabajo para lograr un objetivo común siguiendo las reglas de negocio preestablecidas (10).

Conjunto de actuaciones, decisiones, actividades y tareas que se encadenan de forma secuencial y ordenada para lograr un resultado que satisfaga plenamente los requerimientos del cliente al que va encaminado. En otras palabras, un *workflow* no es más que la sucesión de pasos y decisiones que se siguen para realizar una determinada actividad o tarea que, cuando se trabaja desde el enfoque de la eficacia total, deben ir orientados a satisfacer al cliente (11).

Los procesos de *workflow* son propicios al cambio. Esta característica se debe principalmente a la alta participación de personas en tales procesos, las cuales pueden ser reubicadas, ascendidas o suprimidas en el organigrama de una organización, implicando de este modo la variación temporal o definitiva de un determinado proceso (12).

³Smartcards: Estos dispositivos son tarjetas de plástico que contienen un circuito integrado, son semejantes en tamaño y otros estándares físicos a las tarjetas de crédito. Dicho circuito puede ser de solo memoria o contener un microprocesador con un sistema operativo.

Capítulo 1. Fundamentación teórica

De acuerdo a las definiciones anteriores se puede concluir que: *workflow*, es la automatización del flujo de procesos de un proceso de negocio. Desencadena de manera secuencial y ordenada un conjunto de actuaciones, decisiones, actividades y tareas para conseguir un resultado. De manera que este satisfaga plenamente los requerimientos del cliente al que va dirigido.

1.2.2 Personalización

Proceso que permite a los funcionarios de las oficinas emisoras procesar las solicitudes de pasaportes, mediante una secuencia de opciones, que hacen posible la producción del documento (para el pasaporte de lectura mecánica) (13).

Según el Doc. 9303 (OACI), es el proceso por medio del cual se incorporan al documento el retrato, la firma y los datos personales. Para los pasaportes de lectura mecánica (PLM) con circuito integrado sería todo el proceso anterior más la escritura electrónica en el chip del pasaporte-e para la inserción de los datos biométricos, la duplicación de los datos de la zona de lectura mecánica (ZLM) ,así como el objeto de seguridad para la protección de los datos.

1.2.3 Documento electrónico

El término "documento electrónico" es un concepto que ya no es nuevo, se denomina un documento como electrónico si se encuentra físicamente almacenado en un dispositivo electrónico o unidad de almacenamiento de datos, comprensible sin ningún procesamiento adicional, excepto la presentación del monitor o de la página impresa. Cualquier archivo de computadora que tenga un contenido válido para el trabajo de un universo de usuarios, por reducido que sea, se considera un documento electrónico (14).

A los efectos del presente estudio un documento electrónico es aquel documento que se encuentre físicamente almacenando un dispositivo o unidad de almacenamiento electrónico o sea un chip, en el que el contenido está codificado mediante algún tipo de código digital que puede ser leído, interpretado, o reproducido.

1.2.4 Middleware

Un middleware es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción, situada entre la de aplicación y las inferiores (sistema operativo y red) (15). Son usados para relacionar sistemas que precisan intercambio de información, de modo que se pueda realizar la unión a través de interfaces de alto nivel (16).

Entre sus funciones están:

- Transparencia de la heterogeneidad de los componentes de hardware, sistemas operativos y

Capítulo 1. Fundamentación teórica

protocolos de comunicación.

- Hacer más factible el desarrollo de aplicaciones, brindando abstracciones de programación común, mediante el enmascaramiento de la heterogeneidad, la distribución del hardware subyacente y sistemas operativos; además ocultar los detalles de la programación de bajo nivel.
- Actualmente son muy usados para interactuar con las *Smartcards* en los computadores personales ya que hacen función de intermediarios entre diversas aplicaciones y los lectores de tarjetas.
- Suministro de un conjunto de servicios comunes a diversas funciones de propósito general, a fin de evitar la duplicación de esfuerzos y facilitar la colaboración entre las aplicaciones (17).

1.2.5 Applet

Es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web. El *applet* debe ejecutarse en un contenedor que lo provee un programa anfitrión mediante un *plugin* o en aplicaciones como teléfonos móviles que soportan el modelo de programación por *applets* (18).

Según el contexto de trabajo, los *applets* son las aplicaciones que corren internamente en una *JavaCard*. Dichas aplicaciones interactúan en todo momento con el *JavaCardRuntime-Environment* (JCRE) utilizando los servicios que este ofrece e implementan la interfaz definida en la clase abstracta *javacard.framework.Applet* (19).

1.3 Definición de Smartcards

Muchas son las formas de referirse a una *Smartcards*, la más común en documentos técnicos es ICC (*Integrated Circuit(s) Card*) o tarjeta de circuito integrado (TCI)⁴, pero comercialmente es más conocida como tarjeta inteligente, aunque igualmente se les conoce como tarjeta inteligente. Estas son de plástico (polivinilo cloruro o PVC), similar en tamaño y otros estándares físicos a las tarjetas de crédito (20). Contienen un circuito integrado, lo que posibilita la ejecución de cierta lógica programada, de ahí que se utilice el calificativo inteligente en su denominación (21).

1.3.1 Seguridad de las Smartcards

Las *Smartcards* han demostrado ser una de las formas más seguras y fiables para la identificación electrónica. Posee un chip a prueba de falsificación que no puede ser duplicado, el mismo posee capacidad de detección de ataques por rayos X y luz ultravioleta, voltajes inusuales y cambios de frecuencia de reloj (19). Su sistema operativo posee control del acceso a memoria, protección de datos y

⁴(TCI): Tarjeta de Circuito Integrado, tarjeta que contiene una microplaqueta electrónica acoplada a una antena que permite comunicar datos entre la microplaqueta y un dispositivo de codificación /lectura sin necesidad de eléctrica directa.

Capítulo 1. Fundamentación teórica

ficheros. Las medidas de protección parten desde los elementos empleados en el material de su elaboración hasta las características usadas en su impresión gráfica, dificultando su reproducción y alteración (19). Con las *Smartcards*, es evidente que se está ante una tecnología que, con productos diseñados e implantados de la manera apropiada, proporciona una oportunidad incomparable para mitigar y, potencialmente, eliminar algunos de los problemas de fraude con los que día a día se enfrenta el hombre, debido a que la seguridad es una de las características más importantes que ofrece este tipo de tarjetas (20). En ella cada fichero lleva asociadas unas condiciones de acceso que deben ser satisfechas antes de ejecutar un comando sobre ese fichero. La configuración del mecanismo de acceso a los ficheros en la tarjeta puede ser predefinida durante la fabricación de la tarjeta y/o durante la personalización de ese fichero. Todo depende de las características de la aplicación que esté en la tarjeta y de la tecnología usada.

1.3.2 Las Smartcards y su relación con los pasaportes electrónicos

Los avances tecnológicos ocurridos a lo largo de estos años han propiciado un auge en el desarrollo y utilización de las *Smartcards*. Se ha podido presenciar la notable evolución de estas desde el momento en que un circuito integrado fue incluido en ellas. Desde finales de 1999 que salieron al mercado de forma masiva las tarjetas sin contacto; éstas no sólo se manejan en el campo de la telefonía celular o en el comercio electrónico sino también para los pasaportes electrónicos (20). Con la incorporación de la tarjeta inteligentes en los pasaporte se incrementó un poco más a la seguridad en los pasaportes permitiendo que estos no sean falsificados o manipulados.

1.4 Pasaportes electrónicos

Según OACI, es el Pasaporte de Lectura Mecánica (PLM) que contiene una microplaqueta de circuito integrado (CI) sin contacto dentro de la cual, se almacenan datos de la página de datos del PLM. Una medida biométrica del titular del pasaporte y un objeto de la seguridad para proteger los datos con tecnología criptográfica de infraestructura de clave pública (ICP)⁵ (6).

Tomando como referencia la definición de OACI se ha concluido que: se llama pasaporte electrónico, al PLM que incorpora un circuito integrado sin contacto, el cual contiene entre además los datos biométricos que se utilizan para comprobar la identidad de las personas en los puntos de control fronterizos. Este es un chip no contiene fuente de energía por sí solo, el lector es quien provee la energía para comunicarse con el chip; de lo contrario los PLM no tuvieran el tiempo de vida esperado que es de 10 años. Una imagen facial, las huellas dactilares o de iris entre otros, son los datos que se almacenan en el chip, el

⁵(ICP): Metodología de Infraestructura de clave pública para permitir la detención de manipulación indebida de los datos de un pasaporte-e.

Capítulo 1. Fundamentación teórica

cual incluye un mínimo de 32 kilobytes de EEPROM (memoria de almacenamiento) y se ejecuta en una interfaz de acuerdo con la norma ISO/IEC14443 estándar internacional (6).

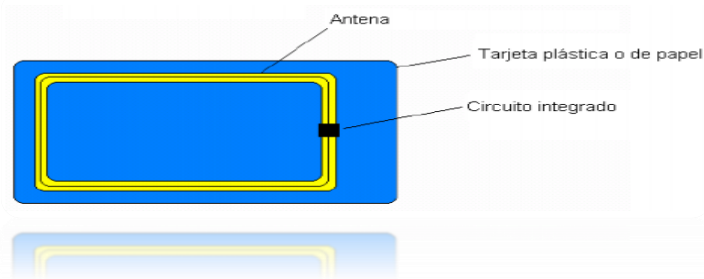


Figura 1. Estructura del chip (22).

Los datos que se almacenan en el chip requieren de una estructura lógica de datos (LDS)⁶ estandarizada, para habilitar la interoperabilidad a nivel global, dicha estructura se encuentra organizada en grupos de datos (DG) (23).

Para mayor seguridad se hace uso de la infraestructura de clave pública (ICP), esta es una combinación de productos de hardware y software, políticas y procedimientos para la implantación y uso de firmas digitales en el documento. La ICP se basa en identidades digitales conocidas como certificados digitales, que actúan como pasaportes electrónicos y vinculan la firma digital del usuario a su clave pública. La ICP debe constar de los siguientes componentes:

Componentes
Una política de seguridad.
Autoridad de certificación (CA).
Autoridad de registro (RA).
Sistema de distribución de certificados.
Aplicaciones habilitadas por ICP.

Tabla 1. Componentes de la ICP (24)

Permite la autenticación de usuarios frente a otros usuarios y usar la información de los certificados de identidad para cifrar y descifrar mensajes, firmar digitalmente cualquier información y garantizar el no repudio de un envío. En este caso, permite autenticar los datos almacenados electrónicamente en el chip del pasaporte, así como para compartir y actualizar el conjunto de llaves públicas para todos los DVLM⁷ que no estén vencidos; de manera que se encuentren en existencia de todos los países participantes en

⁶(LDS): La estructura lógica de datos que describe la forma en que han de ingresarse y formatearse los datos en el pasaportes-e.

⁷ Documento de Viaje de lectura mecánica.

Capítulo 1. Fundamentación teórica

cualquier momento (25). El símbolo que identifica a un pasaporte como portador de un chip es el siguiente:

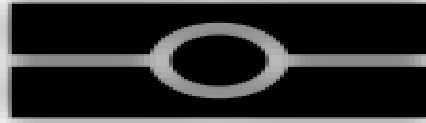


Figura 2. Símbolo del pasaporte-e. (26)

Los datos en el chip son ubicados en el sistema de ficheros definido por la ISO 7816-4. Los ficheros están organizados jerárquicamente en ficheros dedicados (DF) y ficheros elementales (EF). Los DF contienen los ficheros elementales y otros ficheros dedicados. Un fichero maestro (MF), determinado por el sistema operativo que será la raíz del sistema de ficheros (26).

1.4.1 Organización de los Datos de los pasaportes por Zonas

Los datos constarán en los PLM en forma legible, tanto de manera visual como por los métodos de captación óptica de caracteres. Se presentan en 7 zonas que a continuación se enumeran.

1. Zona I Encabezamiento
2. Zona II Datos personales (obligatorios y opcionales)
3. Zona III Datos del documentos (obligatorios y opcionales)
4. Zona IV Firma
5. Zona V Elemento de codificación
6. Zona VI Datos Opcionales
7. Zona VII Zona de lectura mecánica (ZLM) (obligatoria) (6)

Zona de Inspección Visual (ZIV) (Zona I a V)

En la figura 3 se muestra el orden en que se pudieran acomodar los datos en la hoja de datos del PLM.

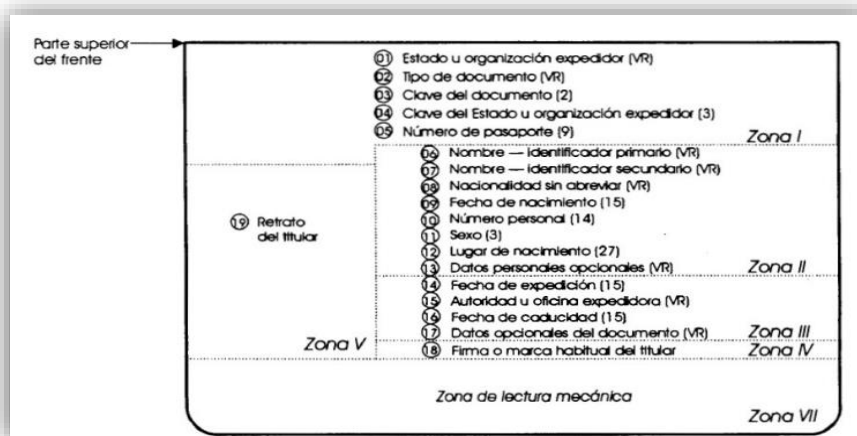


Figura 3. Orden de los datos en la Hoja de Datos (6)

Capítulo 1. Fundamentación teórica

Estos estudios mostraron que una mayor comprensión, más allá de este tamaño, resulta en un reconocimiento del rostro considerablemente menos fiable. No siempre puede conseguirse estos 12 kilobytes debido a que algunas imágenes se comprimen más que otras con la misma relación de compresión, dependiendo de factores como la vestimenta, el color del cabello y el peinado. En la práctica el valor óptimo para utilizar en el pasaporte-e es de tamaño medio de la imagen del rostro está en la gama de 15k-20k (26).

1.4.2 La protección de datos del Pasaporte electrónico

Además de la autenticación pasiva mediante firmas digitales, los estados pueden optar por una seguridad adicional, utilizando formas más complejas de proteger la microplaqueta y sus datos (26).

Método de seguridad básico				
Método	Expedidor	Sistema de inspección	Ventajas	Deficiencias
Autenticación pasiva (5.6.1)	M	M	Prueba que el contenido de la SO _D y la LDS son auténticos y no se han modificado.	No impide una copia exacta o sustitución de plaqueta. No impide acceso no autorizado. No impide despumado.

Tabla 2. Método de Seguridad básico de los datos electrónicos en el DVLM (26).

MÉTODOS DE SEGURIDAD AVANZADOS				
Comparación de ZLM convencional (OCR-B) y ZLM basada en plaqueta (LDS)	N/A	O	Prueba que el contenido de la microplaqueta y el DVLM físico se corresponden	Añade complejidad (menor). No impide una copia exacta de la microplaqueta y del documento convencional.
Autenticación activa (5.6.2)	O	O	Impide la copia del SO _D y prueba que ha sido leído de la plaqueta auténtica. Prueba que la plaqueta no ha sido sustituida.	Añade complejidad. Requiere plaquetas con procesador.
Control de acceso de base (5.7)	O	O	Impide despumado y uso impropio. Impide escucha furtiva de las comunicaciones entre DVLM y sistema de inspección (cuando se usa para establecer un canal de sesión cifrado).	No impide una copia exacta o sustitución de plaquetas (también exige copiar el documento convencional). Añade complejidad. Requiere microplaquetas con procesador.
Control de acceso ampliado (5.8.1)	O	O	Impide acceso no autorizado a las características biométricas adicionales. Impide despumado de características biométricas adicionales.	Exige gestión de claves adicional. No impide una copia exacta o sustitución de plaqueta (también exige copia del documento convencional). Añade complejidad. Requiere microplaquetas con procesador.
Cifrado de datos (5.8.2)	O	O	Protege características biométricas adicionales. No exige microplaquetas con procesador.	Exige gestión de claves de descifrado complejas. No impide una copia exacta o sustitución de la microplaqueta. Añade complejidad.

Tabla 3. Métodos de Seguridad avanzados de los datos electrónicos en el DVLM (26).

Capítulo 1. Fundamentación teórica

1.5 Estándares relacionados con los pasaportes

Este epígrafe recoge algunos estándares que guardan relación con documentos de viaje en este caso con los pasaportes.

1.5.1 Documento de Viaje de Lectura Mecánica (DVL M) 9303

El Doc. 9303 es un estándar para documentos de viajes, cuenta con tres partes y ha sido aprobado por la OACI. El documento propone algunas normas para hacer funcionales y seguros los pasaportes de lectura mecánica y los electrónicos. En él se precisan especificaciones a las que deben ajustarse estos tipos de documentos, en cuanto a los elementos de seguridad física y lógica (26).

En él se describen aspectos para la elaboración de los documentos, la transportación y la personalización de los mismos, además del formato, las dimensiones, la calidad y el posicionamiento de los elementos en el documento. Respecto a los pasaportes electrónicos se proponen algunos métodos de seguridad lógica, entre los que se incluye la infraestructura de clave pública, así como la estructura lógica que deben tener los datos dentro del circuito integrado para que puedan ser leídos desde cualquier sistema que se ajuste a dichas normas (26).

1.5.2 ISO 7816

Con el propósito de producir aplicaciones para *Smartcards* que lograsen funcionar con terminales de diversos fabricantes se instauraron un conjunto de especificaciones las cuales conforman el ISO 7816. En él se definen características físicas, eléctricas, lógicas, de seguridad, estructura de datos y protocolos de comunicación (27). El estándar ISO 7816 está compuesto por 12 partes:

Parte 1: Características físicas

Parte 2: Dimensiones y localización de los contactos

Parte 3: Señales electrónicas y protocolos de transmisión

Parte 4: Comandos interindustriales para el intercambio

Parte 5: Sistema de numeración y procedimiento de registro para los identificadores de aplicación

Parte 6: Elementos de datos interindustriales para el intercambio

Parte 7: Comandos interindustriales para Tarjetas de Lenguaje Estructurado de Consulta (SCQL)

Parte 8: Comando para operaciones de seguridad

Parte 9: Comando para el manejo de tarjetas

Parte 10: Señales electrónicas y respuesta para resetear tarjetas sincronizadas

Parte 11: Verificación personal a través de métodos biométricos

Parte 12: Aplicación de información criptográfica (28)

Capítulo 1. Fundamentación teórica

1.5.3 ISO 14443

El estándar ISO 14443 se relaciona con las tarjetas electrónicas. Este estándar define los parámetros a seguir por las tarjetas de proximidad que utilizan el sistema RFID, los componentes que estas deben incluir, la frecuencia por la que debe operar la antena incluida en el sistema RFID, así como el comportamiento del lector con respecto a la tarjeta (28).

Además cuenta con cuatro partes donde son descritos dos tipos de tarjetas: A y B. Se muestran sus diferencias en cuanto a los métodos de modulación, codificación de los planes o el protocolo de inicialización de los procedimientos. Así como la descripción del protocolo T=CL usado para el intercambio de datos (17).

1.5.4 PC/SC

Personal Computer/SmartCard (PC/SC) son un conjunto de especificaciones para facilitar la comunicación entre las *Smartcards*, los lectores de estas y las computadoras (29). Define una API⁹ de programación que permite a los desarrolladores trabajar con lectores de diversos fabricantes de una misma manera. El interfaz de PC/SC define los interfaces, estándares para una variedad de operaciones inteligentes de la tarjeta. Los más comunes son:

- Enumerando y describiendo a lectores de *Smartcards* adjuntos.
- Petición de la información sobre estados de la tarjeta y del lector.
- Cambio de comandos con las tarjetas (30).

1.6 Escritura electrónica

En este epígrafe se hace una breve historia de la escritura electrónica, y se explica en qué consiste la misma.

Historia de La Escritura Electrónica

En la década de los 90's se dio un paso irreversible, fue la creación por parte de la empresa Intel de un microprocesador que contenía alrededor de 2.000 transistores en miniatura interconectados por medio de la capas de silicio. Estos microprocesadores descartaron las posibilidades que empezaron a ser a partir de ese momento obsoletas, de técnicas manuales para la escritura de planos de procesos y sistemas de información (31).

La escritura electrónica había nacido y junto a ella el hecho de poder almacenar información en circuitos y sistemas que necesitaban máquinas diseñadas para tales propósitos específicamente. Estas máquinas

⁹Una interfaz de programación de aplicaciones o API (en inglés *Application Programming Interface*) es el conjunto de funciones procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Capítulo 1. Fundamentación teórica

computadoras tenían, además, la facultad, por primera vez en la historia, de seguir órdenes que les eran suministradas por escrito y dar cuenta de los resultados que hallaran en dichos procesos de la misma forma; leer y escribir de forma electrónica (31).

1.6.1 Personalización

Escritura electrónica es la forma de registrar información en un medio magnético cuyo lenguaje es el sistema binario: en código de ceros y unos. El proceso se realiza por medio del uso de computadores que a su vez se pueden representar en una pantalla la adaptación de registros en letras o caracteres ideográficos, así como representaciones de imágenes, sonidos, videos, colores, y texturas de superficies (31). En la figura siguiente muestra un ejemplo de este tipo de escritura.

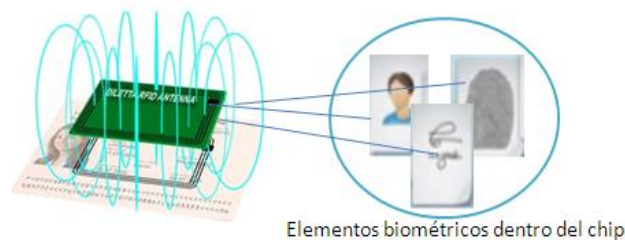


Figura 5. Ejemplo de escritura electrónica.

Concebir un documento electrónico requiere una infraestructura tecnológica necesaria para la captura de datos, imágenes y la validación de identidad de los ciudadanos, así como para efectuar la personalización del documento y su entrega. Los pasos o etapas anteriores son fundamentales para llegar a la emisión de dicho documento. En la personalización guardan gran relación el tipo de documento con los dispositivos de impresión a utilizar.

1.6.2 Proceso de Personalización de Documentos electrónicos

Lograr obtener un documento personal utilizable a partir de un documento impreso, implica un proceso de personalización. Esta etapa es de vital importancia para todo documento en términos de seguridad, calidad y otros tantos aspectos que se aseguran mediante dicho proceso (32). La personalización de documentos cuenta con dos factores:

- Personalización Eléctrica o Electrónica. Se refiere a la escritura de los datos en el chip.
- Personalización Óptica. Se personalizan los datos que se encuentran en la ZIV (22).

Capítulo 1. Fundamentación teórica

Los procesos de personalización se pueden hacer de dos formas diferentes: Personalización centralizada y descentralizada. La OACI sugiere que la personalización sea de forma centralizada, por medidas de seguridad y control del documento.

1.6.3 Organización del proceso de personalización de documentos

En este epígrafe se exponen las dos formas de personalización existentes y se explica en qué consisten cada una de ellas.

Personalización Centralizada

Este tipo de personalización implica el envío de una solicitud al centro donde se realizará la personalización, para la obtención del documento, el ciudadano debe esperar varios días una vez realizada la solicitud para la obtención de un documento. Este tipo de personalización es la opción preferida para muchos países de la Unión Europea y significa que la personalización se realiza en un solo sitio, proporcionando ventajas como la mejora en la calidad y coherencia de la impresión, no se utiliza ningún sistema de logística para la distribución de los documentos vírgenes y el uso de las medidas de seguridad consiguen ser más fuertes a fin de proteger los datos privados de los ciudadanos. Este proceso es usado en países como Alemania, Hungría, Finlandia, Bélgica, Venezuela, entre otros (32).

Personalización Descentralizada

La Personalización Descentralizada se realiza en más de un sitio, la cual posee como ventaja, que el proceso de solicitud del documento sea más rápido y proporciona una oportunidad de vincular al ciudadano con su documento, pero por otra parte muestra desventajas como son: incoherencia entre las oficinas en términos de calidad, personalización y el control de identificación, los riesgos de seguridad son considerablemente más elevados porque la seguridad en la infraestructura tiende a ser menos avanzada que en entornos centralizados, y existe un mayor riesgo en cuanto al desvío de los documentos de identificación vírgenes. Este proceso es usado en países como Italia, Portugal, Grecia, Bulgaria, España entre otros (32).

1.7 Análisis de soluciones existentes

Durante el estudio de las soluciones existentes relacionadas con la personalización de pasaportes electrónicos se identificaron un conjunto de soluciones que a continuación se describen.

1.7.1 Nivel Nacional

SPDI: Garantiza la personalización e impresión de documentos de identidad, probando además la calidad y el proceso de embalaje de estos. Cuenta con un grupo de funcionalidades que posibilitan una mejor

Capítulo 1. Fundamentación teórica

atención a los ciudadanos y un mayor rendimiento y especialización de los funcionarios durante el trámite (33). El sistema se encuentra basado en una solución cliente servidor, desarrollada en la plataforma .Net, con el .Net framework 4.0. El núcleo principal del servidor de aplicaciones es un motor de procesos que utiliza la tecnología *WWF* la cual centra su funcionamiento en una arquitectura orientada a servicios. Es una solución orientada a procesos gestionado por *WWF*. A este son integrados componentes que tienen tanta importancia como el propio sistema ya que estos son los encargados de darle seguridad a los recursos a través de *IDM*¹⁰, de mantener una comunicación flexible por el navegador con los dispositivos haciendo uso del *DGM*¹¹. Cada uno de estos componentes se integra totalmente en el sistema sin que los procesos de negocio se vean afectados directamente, permitiendo el desarrollo en paralelo de todos ellos. Gracias al componente *Bison Framework* que hace función de integrador (34).

Sistema EMIPAS: Software encargado de la emisión de pasaportes en Cuba, trabaja con estándares de seguridad y calidad internacionales, en un ambiente confiable y seguro, tecnológicamente abierto, y posee una interfaz sencilla e intuitiva. Garantiza la personalización de pasaportes de lectura mecánica con código de barra bidimensional para datos biométricos y alfanuméricos. Cuenta con un módulo de control de calidad y supervisión para garantizar que el proceso de personalización se halla efectuado correctamente. El sistema EMIPAS fue desarrollado por *DATYS*¹² y está enfocado en la personalización de pasaportes (13).

1.7.2 Nivel Internacional

Bundesdruckerei:

Este sistema se basa en impresión de la página de datos por sistema *InkJet* y posterior laminación, utiliza equipos multifuncionales o modulares indistintamente. Ha desarrollado más de 135 millones de carnés de identidad y más de 64 millones de pasaportes en la Unión Europea y tarjeta de identificación. La empresa brinda a los clientes nacionales e internacionales documentos de identidad, tarjetas de alta seguridad, hardware de verificación de documentos, software además de servicios de seguridad del Centro de confianza. Además llevó a cabo del sistema de personalización de pasaportes en Inglaterra. Tienen desarrollado el pasaporte electrónico, pero la adquisición del mismo para el país es muy elevada (35).

¹⁰ Identity Manager (Sistema de Administración de identidades).

¹¹ Sistema para la administración de dispositivos en aplicaciones web.

¹² Tecnologías y Sistemas, es una empresa cubana dedicada al desarrollo de aplicaciones informáticas en diversas líneas de negocios, a las que aporta soluciones con sus productos en las áreas de la identificación biométrica, la gestión de contenidos, la minería en textos, la seguridad técnica integral, la gestión de sistemas empresariales, hoteleros y de agencias de viajes.

Capítulo 1. Fundamentación teórica

Thomas De la Rué:

Al igual que sistema de *bundesdruckeri* basa la impresión de la página de datos por sistema *InkJety* con un laminado multicapa que integra varias técnicas entre ellas tinta ópticamente variable y tintas fluorescentes, depositadas sobre la hoja de personalización en capas sucesivas con el uso de una laminadora diseñada a tales fines. Tienen desarrollado el pasaporte electrónico. Su mayor desventaja es que no utilizan dispositivos ópticos de verificación tipo holograma o kinegrama¹³, los cuales son los más fuertes desde el punto de vista de seguridad y son verificables sin medios técnicos. Este sistema se utiliza en México (36).

Resumen referido a los sistemas Homólogos

A partir del análisis realizado a los sistemas estudiados, se puede concluir que estos solo ayudaron a comprender mejor, aspectos teóricos del proceso de personalización. Aunque se pudo apreciar que en el mundo existen muchos sistemas capaces de personalizar un documento electrónico. En cambio el SPDI no se encuentra capacitado para ello, por esta razón se propone integrar un nuevo flujo de trabajo con la capacidad de personalizar un pasaporte electrónico a dicho sistema.

1.8 Tecnologías, herramientas y metodología utilizadas.

En este epígrafe se brinda una breve caracterización de la metodología, tecnologías y herramientas que se utilizan para el desarrollo del flujo propuesto. Algunos de ellos son definidos por políticas del proyecto.

1.8.1 Metodología de desarrollo

El desarrollo un software requiere una guía que permita controlar el proceso de construcción del software. A estas guías se le denominan metodologías de desarrollo de software. La correcta selección de la metodología es un punto fundamental para el éxito de cualquier producto. Estas definen (Quién debe hacer, Qué, Cuándo y Cómo debe hacerlo).

Las Metodologías se clasifican en dos grupos: Metodologías Tradicionales (Pesadas) y Metodologías Ágiles. Las metodologías tradicionales son aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y tienen una rigurosa definición de roles. Las ágiles recalcan una vital importancia en la capacidad de respuesta a los cambios e involucran activamente al cliente en el proceso (37). La siguiente tabla muestra la comparación o diferencia éntrelos dos tipos de

¹³ Imagen óptica tridimensional registrada por medio de rayos láser, sobre una emulsión sensible especial. Procesada e iluminada adecuadamente, la imagen además de en tres dimensiones, aparece saliendo de sus límites, hacia afuera y/o hacia dentro de su marco, variando de perspectiva según sea la posición del espectador, es tan asombroso, que es difícil resistir la tentación de tocarlo.

Capítulo 1. Fundamentación teórica

metodologías anteriormente mencionadas. La siguiente tabla muestra las diferencias entre las metodologías ágiles y las tradicionales.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo de desarrollo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tabla 4. Tabla comparativa de los dos grupos de Metodologías (37).

1.8.2 Extreme Programming (XP)

Es la más preponderante dentro de las metodologías ágiles. *Extreme Programming* es la metodología más apropiada para entornos volátiles. Es una de las metodologías más exitosas en la actualidad utilizadas para proyectos de corto plazo (37).

Extreme Programming es la adecuada para los proyectos con requisitos imprecisos, muy cambiantes y con un riesgo técnico excesivo. Esta metodología tiene la particularidad de tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto (37).

Fundamentación de XP como metodología a utilizar

Los aspectos esenciales que incidieron para la elección de esta metodología son:

- Necesidad de obtener un producto a corto plazo.
- Cantidad de miembros del equipo de desarrollo (dos personas).

Capítulo 1. Fundamentación teórica

- Presencia del cliente en el grupo de desarrollo.
- Asumir una metodología robusta implica una cantidad excesiva de roles y gran volumen de información generada durante todo el ciclo de vida del proyecto. Es por ello que se hace difícil el desarrollo de esta metodología por un equipo pequeño.

Fases de XP

La siguiente figura muestra las fases en las que se subdivide el ciclo de vida de un proyecto de software con XP.

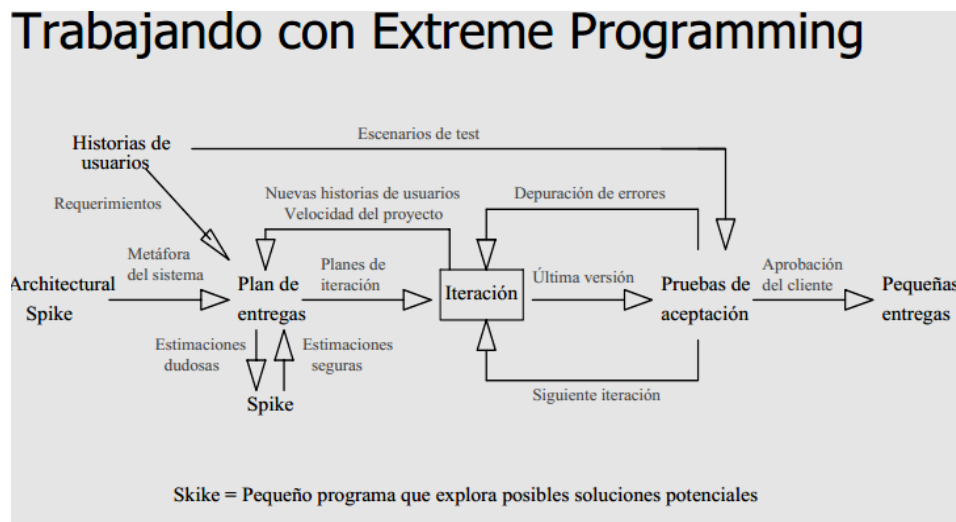


Figura 6. Fases de un proyecto con XP (38).

XP como metodología ágil, enfatiza en el carácter interactivo e incremental del desarrollo. La figura anterior muestra las fases en las que se subdivide el ciclo de vida de un proyecto de software con XP, el cual consta de varias fases:

Exploración

Esta es la fase donde los clientes plantean a grandes rasgos las Historias de Usuario (HU) que son de gran importancia para la primera entrega del producto, así como la confección de la metáfora del sistema que ayuda al equipo a entender las relaciones entre los principales componentes del sistema. Además el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo (37).

Capítulo 1. Fundamentación teórica

Planificación

En esta fase es donde se define la prioridad de las HU y posteriormente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Los artefactos que se generan son el Plan de iteraciones, donde los elementos a tener en cuenta durante su elaboración son: historias de usuario no abordadas, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior (al final de la última iteración el sistema debe estar listo para entrar en producción). Otro artefacto es el Plan de entrega, compuesto por iteraciones de no más de tres semanas (37).

Iteraciones

Fase que incluye varias iteraciones sobre el sistema, primeramente para realizar la entrega. Se diseñan las tarjetas CRC (Clase, Responsabilidad, Colaboración). Una tarjeta CRC representa un objeto. Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior (37).

Producción

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, consecuencia de cambios durante dicha fase. Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana (37).

1.8.3 Lenguaje de modelación visual

El Lenguaje Unificado de Modelado (*UML*, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y usado en la actualidad. Es un lenguaje gráfico para especificar, visualizar, construir y documentar un sistema de software. UML ofrece un estándar para describir un modelo del sistema, abarcando aspectos conceptuales tales como procesos de negocios y funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Está concebido para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de *UML* no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo (39).

1.8.4 Altova UModel

Altova UModel es una herramienta que diseña visualmente modelos de aplicaciones en *UML* y genera

Capítulo 1. Fundamentación teórica

código *Java*, *C#*, o *Visual Basic .NET* y documentación del proyecto. Se utiliza para crear e interpretar diseños de software. Este es utilizado ya que permite realizar la ingeniería inversa de los programas existentes, al final los diseños, los completa con la generación de código. Además con la utilización de *UModel* se puede corregir el código generado o los modelos y completar la ronda produciendo automáticamente nuevos diagramas o regenerando el código (40). Algunas características de *UModel* para el desarrollo de software basado en las capacidades de modelado avanzado son:

- Soporte para los 14 tipos de diagramas *UML*.
- Modelado de esquemas *XML* en diagramas *UML*.
- Generación de código fuente en lenguajes *Java*, *C#*, y *VB.NET*.
- Ingeniería inversa de código fuente y ficheros binarios *Java*, *C#* y *VB.NET*.
- Crea diagramas de secuencia desde el código fuente realizando la ingeniería inversa.
- Generación de documentación personalizable de proyecto (41).

1.8.5 Lenguaje de programación

C# es un lenguaje que adopta mucho de la sintaxis de *C* y *C++*, pero tienen un modelo de objetos más sencillo de la plataforma *.NET* el cual es parecido al de *Java*. *C#* fue diseñado para combinar el control de lenguajes de bajo nivel como el *C* y la velocidad de programación de lenguajes de alto nivel como *Visual Basic* (42) (43).

Es utilizado *C#* como lenguaje de programación, puesto que posee varias características que lo convierten en uno de los lenguajes más utilizados por muchos desarrolladores entre las principales se destacan:

- La plataforma *.NET* provee un colector de basura que es responsable de administrar la memoria en los programas *C#*.
- 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).
- 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 (objeto) (44).

Capítulo 1. Fundamentación teórica

1.8.6 .NET Framework 4.0

- Framework desarrollado por la compañía *Microsoft*. Formado por una serie de librerías y herramientas con las que se pueden crear todo tipo de aplicaciones, utilizado por las posibilidades que este brinda. *.NET Framework 4.0* proporciona nuevas mejoras y características que la hacen diferente a sus versiones anteriores, aunque posee la capacidad de funcionar en paralelo con ellas (43). Entre sus características están:
- Mejoras en *Common Language Runtime* (Lenguaje Común en Tiempo de Ejecución) y la biblioteca de clases base (*BCL*).
- Innovaciones en los lenguajes *Visual Basic* y *CSharp*.
- Mejoras en el acceso a datos y el modelado.
- *Entity Framework* donde se permite a los desarrolladores programar con bases de datos relacionales usando objetos *.NET* y *Language Integrated Query* (*LINQ*, por sus siglas en inglés).
- Servicios de datos de *WCF*, componente que permite crear servicios y aplicaciones que usen protocolo de datos abierto para exponer y usar datos a través de la *Web*.
- Mejoras en *ASP.NET*.
- Más control sobre *HTML*, identificadores de elemento y hojas de estilo personalizadas que facilitan enormemente la creación de formularios que admiten optimización del motor de búsqueda y son conformes a los estándares.
- Nuevas características de datos dinámicos, incluidos nuevos filtros de consulta, plantillas de entidad, mayor compatibilidad con *Entity Framework 4* y características de validación y creación de plantillas que se pueden aplicar fácilmente a formularios existentes.
- Mejoras en *WCF*.
- Compatibilidad con servicios de flujos de trabajo de *WCF*, que permiten programas con actividades de mensajería y correlación.
- Detección de servicios y servicio de enrutamiento.

1.8.7 Windows Communication Foundation

Windows Communication Foundation (*WCF*) es el modelo de programación unificado de *Microsoft* para generar aplicaciones orientadas a servicios. Permite a los programadores generar soluciones con transacción segura y de confianza, que se integren en diferentes plataformas y que interoperen con las versiones existentes (43). *WCF* se ha diseñado para ofrecer un enfoque manejable para la creación de servicios web y clientes de servicios web. Puede ser utilizado en innumerables ocasiones y se indican a continuación algunos escenarios de ejemplo:

Capítulo 1. Fundamentación teórica

- Un servicio seguro para procesar transacciones comerciales.
- Un servicio que proporciona datos actualizados a otras personas, como un informe sobre tráfico u otro servicio de supervisión.
- Un servicio de chat que permite a dos personas comunicarse o intercambiar datos en tiempo real.
- Una aplicación de panel que sondea los datos de uno o varios servicios y los muestra en una presentación lógica.
- Exponer un flujo de trabajo implementado utilizando *WWF* como un servicio *WCF*.
- Una aplicación para sondear un servicio en busca de las fuentes de datos más recientes (45).

1.8.8 Windows Workflow Foundation (WWF)

Windows Workflow Foundation (WWF) es una plataforma que posibilita a los usuarios crear un flujo de trabajo en sus aplicaciones. Puede ser utilizado en escenarios simples, como mostrar controles de interfaz de usuario basados en entradas o escenarios complejos (46). Consiste en un espacio de nombres, un motor de flujos de trabajo en proceso y diseñadores para *Visual Studio*. Es una tecnología extensible para desarrollar soluciones de *workflow* sobre la plataforma *.NET* (43). Provee una Interfaz de programación de aplicaciones *API* y herramientas para el desarrollo y la ejecución de aplicaciones basadas en *workflow*, brindando a los desarrolladores la habilidad de definirlos, compilarlos, instanciarlos, depurarlos y rastrearlos. *WWF* provee la plataforma base donde se pueden desarrollar aplicaciones con disímiles procesos, dicha tecnología complementa al *framework.NET* (47).

1.8.9 Bison Framework

BisonFramework es un *framework* para la organización de procesos de negocio con *Windows Workflow Foundation*. Su objetivo principal es suministrar un componente que permita gestionar las instancias de *workflow*, así como encapsular un conjunto de actividades y servicios que dan mayor dinamismo al desarrollo de sistemas centrado en la orquestación de procesos de negocio con *workflow*.

Además es utilizado por las ventajas que brinda *Bison Framework* dentro de ellas se pueden mencionar las siguientes: proporciona una mayor aproximación a los usuarios de negocio, brinda rapidez y flexibilidad para modelar y cambiar los procesos según las necesidades, aporta escalabilidad o capacidad de crecer, fortifica el puente creado por el *workflow* para la comunicación entre el analista y el desarrollador, propone una arquitectura donde se encuentran bien definidas las capas de presentación y negocio, posee actividades y servicios especializados en la orquestación de interfaces de usuario, que permiten definir su flujo de una manera gráfica dentro del *workflow* (48).

Capítulo 1. Fundamentación teórica

1.8.10 Visual Studio .NET

Es un entorno de desarrollo integrado para sistemas operativos *Windows*. Soporta varios lenguajes de programación como *J#*, *ASP.NET*, *C++*, *C#*, y *Visual Basic .NET*. Otro aspecto significativo de *Visual Studio* es que permite la ejecución de los programas paso a paso e incluir puntos de interrupción diversos, además de poder analizar el contenido de las variables a medida que se está ejecutando la aplicación (49).

1.8.11 Developer Suite Gemalto

Developer Suite Gemalto es una herramienta que ofrece un ambiente apropiado para el desarrollo de *Applets JavaCard*. Es un IDE (Entorno de Desarrollo Integrado), que proporciona un conjunto de herramientas para crear y depurar *applets Java Card*. Brinda un ambiente favorable para el diseño y la implementación de *applets* y posibilita simular las funcionalidades de los *applets* antes de ser instalados en las *Smartcards*. *Developer Suite* a diferencia del *NETBeans* permite desarrollar aplicaciones para la versión de *Java Card* (50).

1.9 Conclusiones parciales

- El análisis de los conceptos relacionados con el proceso de personalización y el análisis de los sistemas homólogos existentes, demostró que existen diferentes sistemas que llevan a cabo la personalización, pero no todas están capacitados para personalizar documentos electrónicos.
- El análisis del SPDI evidenció la imperante necesidad de desarrollar el sistema propuesto para llevar a cabo la personalización de pasaportes electrónicos.
- El análisis de la metodología, tecnologías y herramientas según la necesidad existente, permitió determinar que *XP* es la metodología que guiará el proceso de desarrollo de, *Altova UModel* como herramienta de modelado, *.NET Framework 4.0* como plataforma de desarrollo, *Microsoft Visual Studio 2010* como entorno de desarrollo, *Windows Communication Foundation* y *Windows Workflow Foundation*, *Developer Suite* para creación de *applets*, *C#* como lenguaje de programación y *Visual Studio* como entorno de desarrollo.

Capítulo 2. Propuesta de Solución

Capítulo 2. Propuesta de Solución

2.1 Introducción

El objetivo que se persigue con la elaboración del presente capítulo es mostrar la evolución de la solución durante las fases iniciales de Exploración y Planificación, además de presentar los diferentes artefactos generados en las mismas, los cuales serán elementos indispensables en la entrega de la solución final.

2.2 Propuesta de Solución

Con el propósito de proteger los datos personales contenidos en los pasaportes de lectura mecánica y proporcionar mayor el alcance al SPDI, se ha propuesto desarrollar un *workflow* para personalizar documentos como los pasaportes electrónicos, integrándose este al sistema de personalización de documentos de lectura mecánica quedando finalmente, un sistema basado en flujo de trabajo capaz de llevar a cabo la personalización de todo tipo de documento de viaje, ya sea de lectura mecánica o electrónica. El flujo de trabajo está compuesto por un conjunto de actividades que interconectadas entre sí permiten personalizar un documento electrónico. Estas actividades tienen un orden lógico para poder llevar a cabo la personalización de manera exitosa, agrupadas en dos actividades generales como se muestra en la siguiente figura.



Figura 7. Flujo del sistema.

La primera actividad encierra las sub-actividades encargadas de establecer conexión y cerrar conexión con el pasaporte electrónico, todo a través de un canal seguro. La segunda actividad recoge todas las sub-actividades relacionadas con la escritura en el chip como: obtener los datos personales, crear sistema de ficheros ISO/IEC 7816-4, crear los grupos de datos según OACI (DD1 y DG2), crear el SOD y por último introducir datos personalizados al chip.

2.2.1 Modelo de Dominio

A pesar que el problema a resolver está determinado a partir de procesos de negocio bien definidos que permitan modelar el funcionamiento del mismo, se ha decidido crear un modelo de dominio debido a la metodología utilizada. Este permitirá identificar los principales conceptos asociados al problema en cuestión, y la relación entre ellos. El modelo de dominio muestra (a los modeladores) clases conceptuales

Capítulo 2. Propuesta de Solución

significativas en un dominio del problema; es el artefacto más importante que se crea durante el análisis orientado a objetos (51). A continuación en la siguiente figura se muestra el modelo de dominio correspondiente a la solución.

Diagrama de Clases del Modelo de Dominio

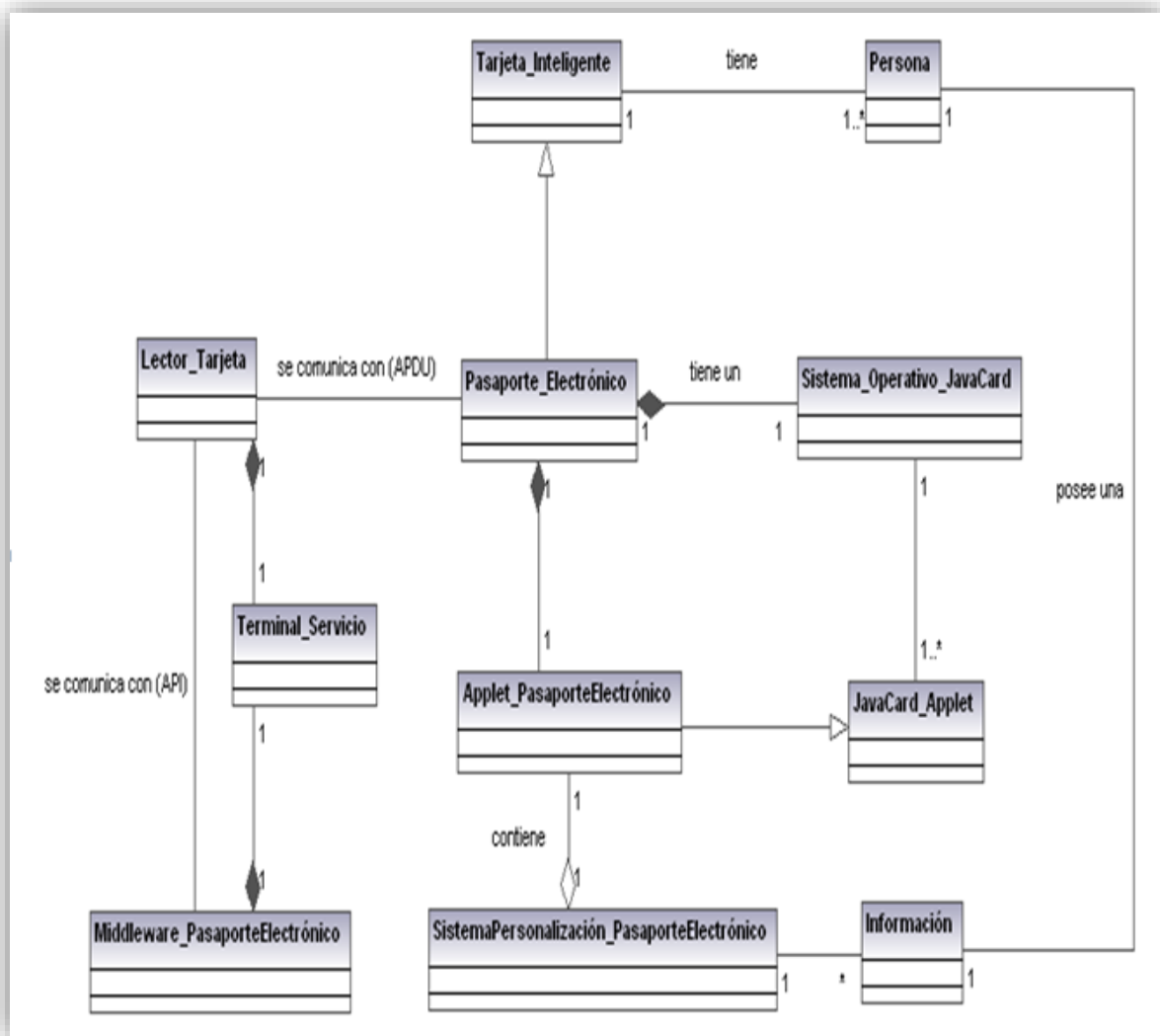


Figura 8. Modelo de Dominio

2.2.2 Glosario de conceptos del modelo de dominio

A continuación los conceptos del modelo de dominio:

Capítulo 2. Propuesta de Solución

Tarjeta Inteligente: Es un dispositivo de PVC¹⁴ o policarbonato similar en tamaño y otros estándares físicos a las tarjetas de crédito. Presenta un circuito integrado, que puede ser de sólo memoria o contener un microprocesador con un sistema operativo que le permita realizar un conjunto de funcionalidades: almacenar información, encriptar información, leer y escribir datos, al igual que un ordenador.

Pasaporte_Electrónico: Es una tarjeta inteligente que contiene tecnología *JavaCard*, la cual contiene datos de identificación de su portador y permite una mayor automatización de los procesos de identificación, logrando una mayor seguridad y efectividad de forma simple.

SistemaOperativo_JavaCard: La tecnología *JavaCard* combina parte del lenguaje de programación Java con un entorno de ejecución optimizado para *Smartcards* y similares. El objetivo de la tecnología *JavaCard* es llevar los beneficios del desarrollo de software en Java al mundo de las *Smartcards*.

JavaCard_Applet: Los *Applets* son las aplicaciones que corren embebidas en la tecnología *JavaCard*. Dichas aplicaciones interactúan en todo momento con el JCRE utilizando los servicios que éste brinda, e implementan la interfaz definida en la clase abstracta *javacard.framework.Applet*.

Applet_PasaporteElectrónico: Es una aplicación implementada en *JavaCard*, instalada dentro de la tarjeta inteligente la cual permite gestionar la información que se almacene dentro del pasaporte electrónico.

SistemaPersonalización_PasaporteElectrónico: Con la información del portador del documento electrónico, se brinda un servicio de personalización para dicho documento, según el estándar ISO/IEC 7816-4 del OACI, se crea un sistema de ficheros en el cual van a estar organizadamente almacenada la información personal del portador del documento.

LectorTarjetas: Es un lector compatible con el estándar PC/SC, el cual define la comunicación entre el *Applet_PasaporteElectronico* y el *Middleware_PasaporteElectronico*, sin realizarle modificaciones a ambas aplicaciones.

Terminal_Servicio: Es un punto en una computadora donde se instalan todas las condiciones para poder interactuar con la solución. Entre los elementos indispensables se puede citar, los lectores PC/SC.

¹⁴ El PVC es un material esencialmente amorfo con porciones sindiotácticas y especial para compuestos flexibles.

Capítulo 2. Propuesta de Solución

Middleware_PasaporteElectrónico: Es un componente que funciona como capa de traducción entre el *Applet_Pasaporte_Electronico* y otro sistema, permitiendo una mejor comprensión de las respuestas obtenidas por la comunicación establecida con la aplicación instalada en el documento electrónico de viaje (*Applet*).

Persona: Portador del documento electrónico de viaje.

Información: Son datos referentes al portador del documento electrónico de viaje y son almacenados en el mismo.

2.2.3 Actores del Sistema.

Los roles o actores son personas con privilegios de interactuar con el sistema simplemente en la parte que le es asignada. Las personas relacionadas con el sistema en este caso son algunos de los actores que intervienen en el SPDI, puesto que el flujo generado formará parte del sistema antes mencionado, en el cual intervienen los siguientes actores.

Rol	Objetivo
Administrador	Es el encargado de realizar el diseño de las plantillas.
Jefe de área del SPDI	Solicitar lotes al sistema Insumo y asignar estos a los puestos de impresión. Generar reportes generales.
Jefe de turno del SPDI	Generar reportes generales.
Funcionario de Impresión del SPDI	Recepcionar y controlar las órdenes de trabajo, listado de documentos a personalizar; así como los documentos vírgenes.
Funcionario de calidad del SPDI	Ejecutar el control de calidad de la impresión. Controlar los documentos con errores de impresión.
Funcionario de Ensamblaje del SPDI	Encargado de preparar y enviar los lotes listos al almacén.

Tabla 5. Actores del Sistema.

Capítulo 2. Propuesta de Solución

2.2.4 Especificación de Requisitos

En el actual epígrafe se realiza la definición y especificación de cada uno de los requisitos necesarios para el desarrollo de la aplicación, el desarrollo de los mismos es llevado a cabo a través de la tecnología de *workflow*.

Requerimientos Funcionales

Describen las funciones que el software va a ejecutar; por ejemplo, ajustarse a un formato de texto o modular una señal. Se conocen también como capacidades del sistema (52).

RF1: Gestionar comunicación

RF1.1: Detectar el lector conectado

RF1.2: Establecer conexión con el pasaporte electrónico

RF1.3: Cerrar conexión con el pasaporte electrónico

RF2: Gestionar personalización

RF2.1: Obtener los datos personales

RF2.2: Crear sistema de ficheros ISO/IEC 7816-4

R F2.3: Crear los grupos de datos según OACI

RF2.3.1 Crear el DG1

RF2.3.2 Crear el DG2

RF2.4: Crear el SOD

RF2.5: Introducir datos personalizados al chip

Requerimientos no funcionales

Son los que actúan para obligar a funcionar la solución. Se conocen también como apremios o requisitos de calidad. Pueden ser clasificados más a fondo en: requisitos de funcionamiento, requisitos de seguridad, requisitos de confiabilidad o uno de muchos otros tipos de requisitos del software (52). A continuación se muestran los requisitos no funcionales por los que se rige el sistema de personalización.

Requisitos no Funcionales	
Soporte	Se redactará un manual de usuarios para documentar el uso del producto desarrollado.

Capítulo 2. Propuesta de Solución

Seguridad	Solo debe tener acceso a la tecnología y a los datos para la personalización el personal autorizado.
Requerimientos de software	Tener instalados en la PC cliente los drivers del lector de tarjetas.
	Tener instalados el Framework.NET 4.0.
	Sistemas operativos, cualquier versión de Microsoft Windows.
	IDE: Visual Studio 2010, Gemalto Developer Suite.

Tabla 6.Requisitos no Funcionales

2.2.5 Historias de usuario

Las historias de usuario son utilizadas en XP para especificar los requisitos del software desde el punto de vista del cliente. A continuación de muestran las tablas correspondientes a las historias de usuarios adecuadas a la solución.

HU Gestión de la comunicación

Historia de Usuario	
Número: HU1	Usuario: Desarrollador
Nombre historia: Gestión de la comunicación	
Prioridad en negocio: Alta	Riesgo en desarrollo: bajo
Puntos estimados: 1	Iteración asignada: 1
Descripción: Permitirá la comunicación mediante el estándar PC/SC, obteniendo los lectores disponibles conectados a la terminal y estableciendo la conexión con el documento electrónico de viaje. Permitirá, además, enviar comandos APDU ¹⁵ y obtener los APDU de respuesta del pasaporte electrónico.	

Tabla 7.Gestión de la Comunicación.

¹⁵APDU (Application Protocol Data Units) - Unidades de Datos en la Capa de Aplicación. Los APDU pueden contener un mensaje de comando o un mensaje de respuesta.

Capítulo 2. Propuesta de Solución

HU Permitir la autenticación

Historia de Usuario	
Número: HU2	Usuario: Desarrollador
Nombre historia: Permitir la autenticación	
Prioridad en negocio: Alta	Riesgo en desarrollo: bajo
Puntos estimados: 1	Iteración asignada: 1
Descripción: Permitirá mediante especificaciones del OACI la autenticación en la tarjeta, estableciendo un canal seguro a través de un <i>Basic Controller Access (BAC)</i> y creando una sección para la transmisión segura de los datos	

Tabla 8. Permitir la autenticación

HU Cargar e instalar JMRTD

Historia de Usuario	
Número: HU3	Usuario: Desarrollador
Nombre historia: Cargar e instalar el <i>Applet</i>	
Prioridad en negocio: Alta	Riesgo en desarrollo: bajo
Puntos estimados: 2	Iteración asignada: 1
Descripción: Se cargará un <i>.CAB</i> ¹⁶ , el cual contendrá una serie de datos en tecnología <i>JavaCard</i> con un formato para personalizar documentos electrónicos de viajes.	

Tabla 9. Cargar e instalar el *Applet*

¹⁶Abreviatura de Cabinet, es el formato nativo de archivo comprimido de Microsoft Windows. Soporta compresión y firma digital

Capítulo 2. Propuesta de Solución

HU Creación de los DG

Historia de Usuario	
Número: HU4	Usuario: Desarrollador
Nombre historia: Creación de los DG	
Prioridad en negocio: Alta	Riesgo en desarrollo: bajo
Puntos estimados: 3	Iteración asignada: 1
Descripción: Crear un sistema de ficheros, identificados por la ISO/IEC 7816-4 del ICAO, donde se almacenarán datos personales referente al portador del documento. Este sistema está organizado jerárquicamente en archivos elementales y dedicados. Los dedicados contienen archivos elementales y otros archivos dedicados.	

Tabla 10.Creación de los DG

HU Creación de los DG1

Historia de Usuario	
Número: HU4_T1	Usuario: Desarrollador
Nombre historia: Creación del DG1	
Prioridad en negocio: Alta	Riesgo en desarrollo: bajo
Puntos estimados: 1.5	Iteración asignada: 1
Descripción: Etiqueta de la Información de la Zona de Lectura Mecánica, '61'; Obligatorio. Este EF (Archivo Elemental) contiene la información obligatoria de la Zona de Lectura Mecánica (MRZ) para el documento en la plantilla '61'. La plantilla contiene un objeto de datos, '5F1F'. El objeto de datos del MRZ es un Dato Elemental compuesto por la misma información de OCR-B MRZ (Reconocimiento Óptico de Caracteres - Binario) que fue impreso en el documento de viaje.	

Tabla 11.Creación del DG1

Capítulo 2. Propuesta de Solución

HU Creación de los DG2

Historia de Usuario	
Número: HU4_T2	Usuario: Desarrollador
Nombre historia: Creación del DG2	
Prioridad en negocio: Alta	Riesgo en desarrollo: bajo
Puntos estimados: 1.5	Iteración asignada: 1
Descripción: estipulado por la ISO/IEC 7816-4 que representa la característica biométrica de interfuncionamiento mundial, donde se almacena una imagen del rostro del titular como elemento de entrada al sistema de reconocimiento del rostro. La cual se obtendrá de la fotografía utilizada para crear el retrato exhibido impreso en la página de datos del pasaporte-e y se codificara a la ISO.	

Tabla 12.Creación del DG2

HU Creación del SOD

Historia de Usuario	
Número: HU5	Usuario: Desarrollador
Nombre historia: Creación del SOD	
Prioridad en negocio: Alta	Riesgo en desarrollo: bajo
Puntos estimados: 2	Iteración asignada: 2
Descripción: Cuando se crea cada grupo de dato también se almacenan de forma independiente un hash, es decir, un resumen de dicho grupo de datos; estos hash son integrados al SOD. Posteriormente se crea un SOD para todos los datos de manera general.	

Tabla 13.Creación del SOD

Capítulo 2. Propuesta de Solución

2.2.6 Metáfora

El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema (37). Una metáfora para el sistema básicamente es la historia que todo el mundo puede contar acerca de cómo funciona el mismo. Elegir una metáfora para el sistema permite mantener la coherencia de nombres de que todo aquello que se va a implementar es muy importante. El Flujo de trabajo para llevar a cabo la personalización en chip del pasaporte, cumple con los estándares de los pasaportes electrónicos de OACI, la ISO 7816-4 y sus complementos. Lo primero que se debe lograr es la autenticación. Luego se podrán realizar operaciones básicas como reconocer los lectores conectados a la máquina y establecer una conexión con el pasaporte electrónico, cerrar conexión con el pasaporte electrónico, gestionar personalización que encierra: obtener los datos personales, crear sistema de ficheros ISO/IEC 7816-4, crear los grupos de datos según OACI en te caso los DG1 y DG2 y su correspondiente SOD. Para finalizar introducir datos personalizados al chip y el datos de seguridad. Todas las operaciones serán implementadas en el *workflow* que se encargará de enviarla al Middleware, este a su vez se lo enviará al *applet* del documento al que se le está efectuando la personalización.

2.3 Arquitectura

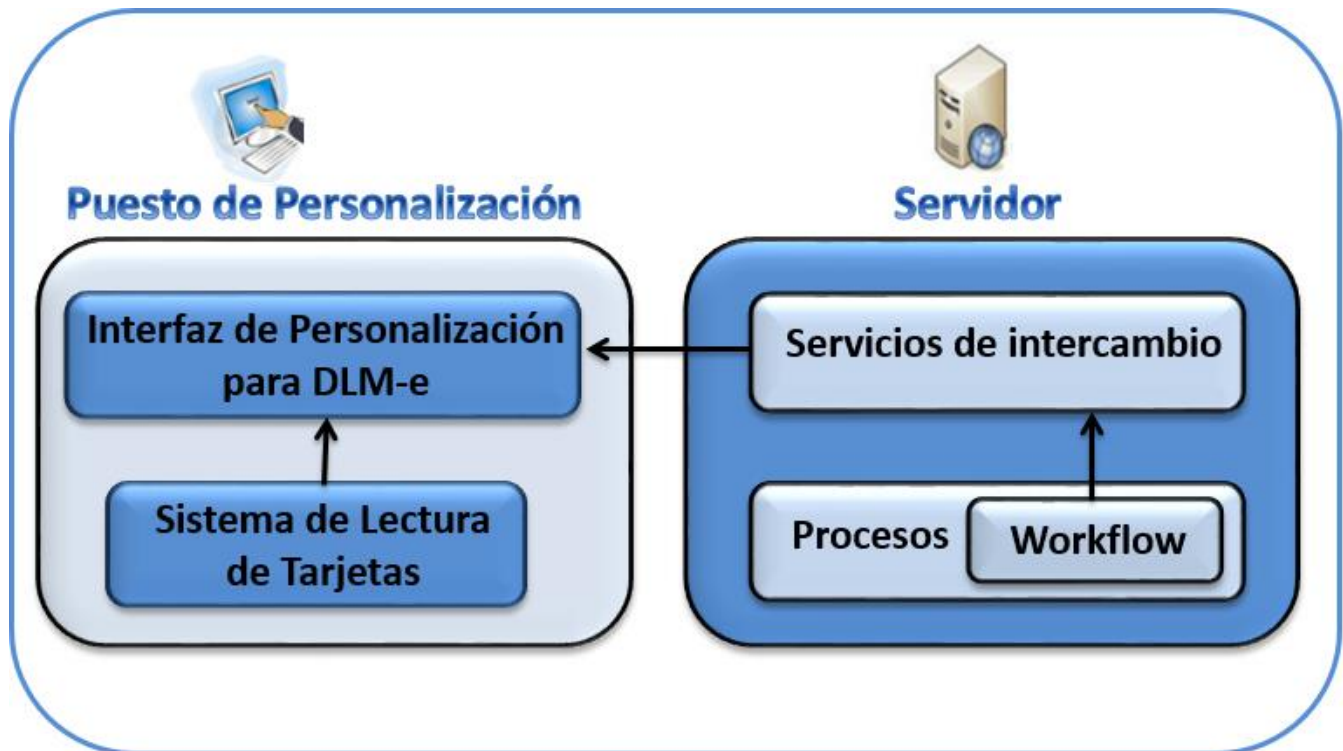


Figura 9.Arquitectura del Sistema.

Capítulo 2. Propuesta de Solución

Descripción de la Arquitectura

La arquitectura del sistema de personalización para los pasaportes electrónicos basa su implementación en una arquitectura cliente/servidor. La capa cliente se encuentra dividida en dos subcapas, la interfaz de personalización para pasaporte-e, mediante la cual hace uso de los servicios implementados en el *workflow* para que este los ejecute y la subcapa de sistema de lectura de tarjetas que es la encargada de la conexión con los dispositivos que se utilizarán para la personalización de pasaportes electrónicos. En segundo lugar se tiene una capa servidor, en la que se encuentra la subcapa de servicios encargada de brindar los servicios de intercambio con el puesto de impresión eléctrica, además consta con una subcapa llamada procesos que contiene los *workflow* encargados de manejar el proceso de negocio del sistema de personalización. Para revisar la arquitectura completa del sistema SPDI y ver la descripción de la capa de procesos-servicios ([Ver anexo2](#)).

2.4 Patrones de Diseño

Los patrones de diseño permiten dar solución a problemas comunes que se presentan dentro del ciclo de vida del proceso de desarrollo de software, describiendo dichos problemas de forma sencilla e indicando los pasos necesarios para resolverlos (53).

Se utilizaron para el diseño los patrones **GRASP**, los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos expresado en forma de patrones. Algunos de los patrones que abarca *GRASP* son: Experto, Creador, Alta Cohesión, Bajo Acoplamiento y el patrón Controlador. A continuación se describe uno de ellos.

Creador: Permite decidir cuáles serán las clases creadoras de otras clases. La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia, es útil contar con un principio general para la asignación de las responsabilidades de creación. Ejemplo: la clase *SmartCardReaderService* contiene objetos de la clase, *CPDI.Devices.CardReaders.HttpClientWrapper*, ésta, asume la responsabilidad de crear las instancias de la misma.

```
public class SmartCardReaderService : ISmartCardReaderService
{
    public SmartCard.Devices.CardReaders.ISmartCardReader client = new HttpCardReaderClient();
    public ICAOLDSSecureChannel _secureChannel;

    public void Connect(string Reader, SHARE ShareMode, PROTOCOL PreferredProtocols) ...

    public void Disconnect(DISCONNECT Disposition) ...
}
```

Figura 10. Ejemplo del Patrón Creador

Capítulo 2. Propuesta de Solución

Otros patrones empleados son los patrones **Gang of Four** más conocidos como patrones **GoF**. En la arquitectura n-capas hay cerca de cien patrones que son variantes del patrón básico de capas. Los patrones de uso común relativos al estilo son Fachada, Adaptador, Puente y Estrategia entre otros (54). Los patrones anteriormente mencionados se evidencian en el SPDI debido a la arquitectura adopta, estructurada en n-capas. El flujo está ubicado en la capa de procesos-servicios y para su desarrollo se usó el patrón Estrategia, le cual se explica más adelante.

Estrategia (*Strategy*): Es un patrón de comportamiento. Define una familia de algoritmos encapsulando por separado cada uno de ellos y haciéndolos, por tanto, intercambiables. Esto permite a los algoritmos variar con independencia de los clientes que los usan (54) . Se utiliza cuando muchas clases relacionadas sólo se diferencian en su comportamiento Ejemplo: las clases *SmartCardReader.Client*, *ISmartCardReader.Controller* y *SmartCardReader.Common*.

2.4.1 Patrones Workflow

Los patrones para el diseño de workflow van desde los más simples como el patrón secuencial hasta los más complejos, por ejemplo, el patrón de sincronización. A continuación se muestran los patrones utilizados para el diseño del workflow.

Patrones de control de flujos básicos: Sirven para modelar procesos secuenciales, paralelos o aquellos que incluyan alguna decisión. Los diagramas de proceso de *workflow* implementan este patrón por defecto en cada actividad. O sea, por defecto, los conectores entre tareas implican precedencia, salvo que se explicita lo contrario (55). Las figuras que se muestran a continuación ilustran su aplicación:

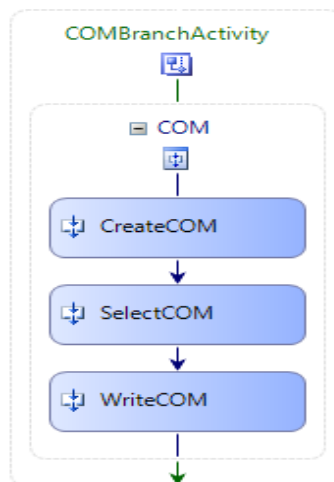


Figura 11.Ejemplo1 de uso de patrones de control de flujos básicos

Capítulo 2. Propuesta de Solución

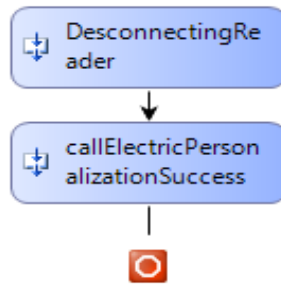


Figura 12. Ejemplo 2 de uso de patrones de control de flujos básicos

Patrones de ramificación avanzada y sincronización: Como se muestra en la figura éstos superan a los de control de flujo básico al permitir tipos avanzados de bifurcación y sincronización.

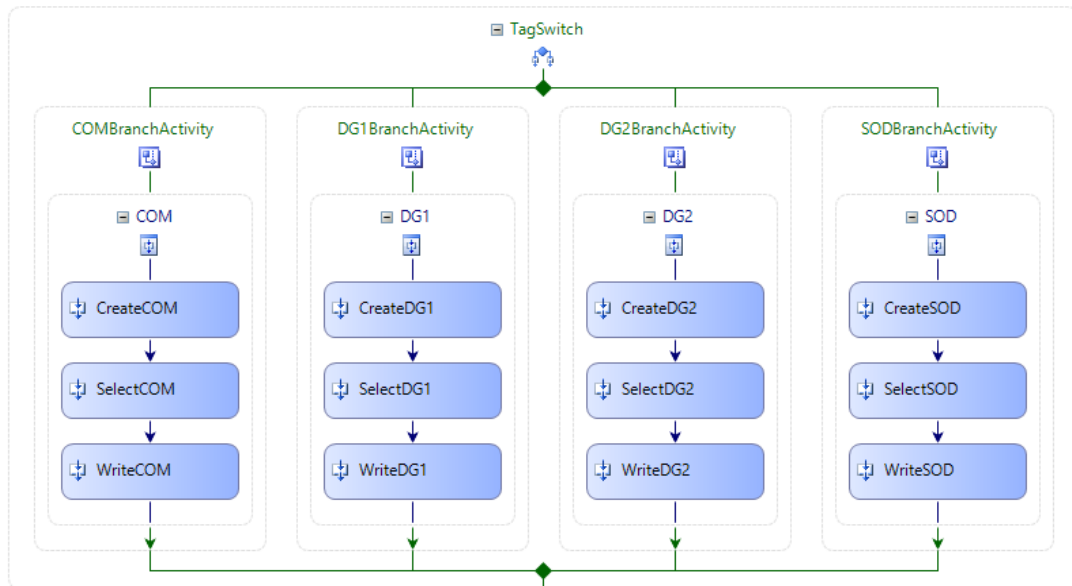


Figura 13. Ejemplo del patrón de ramificación avanzada y sincronización

2.5 Diagrama de Clases del Diseño

El diagrama de clases captura la estructura lógica del sistema y las clases que constituyen el modelo. Es un modelo estático, que describe las clases que existen y qué atributos y comportamiento tienen. Los diagramas de clases son los más útiles para ilustrar las relaciones entre las clases e interfaces. Las generalizaciones, las agregaciones y las asociaciones son todas valiosas para reflejar la herencia, la composición o el uso y las conexiones respectivamente (56). Son utilizadas durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema y las clases que se encargarán del funcionamiento así como la relación entre ellas. A continuación

Capítulo 2. Propuesta de Solución

se muestra el diagrama de clases del sistema y para ver el diagrama de clase más completo ([Ver anexo 3](#)).

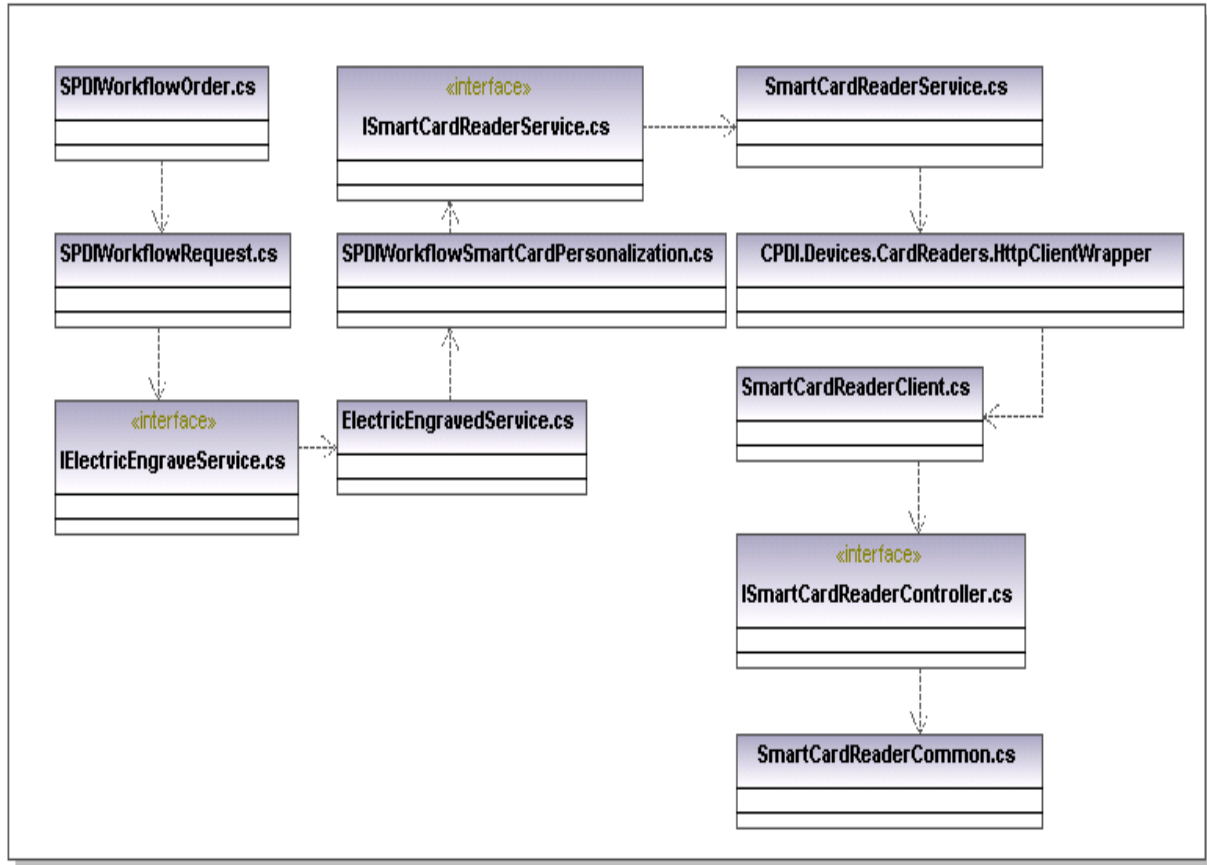


Figura 14. Diagrama de Clases del Diseño

Descripción de las Clases del Sistema.

El sistema comienza en el *SPDIWorkflowOrder* que es la clase encargada de gestionar las órdenes, las cuales contienen un conjunto de solicitudes. Cada solicitud es recibida por la clase *SPDIWorkflowRequest*; en esta clase se manejan los estados de las solicitudes, es decir, se efectúa dicha solicitud según su descripción. Cuando entra al estado de *ElectricEngraveService* se crea una instancia de *SPDIWorkflowSmartCardPersonalization* mediante la interrelación de la interfaz *IElectricEngraveService* implementada por la clase *ElectricEngraveService*. La clase *SPDIWorkflowSmartCardPersonalization*: es la encargada de llevar a cabo el flujo de personalización electrónica haciendo uso de los métodos del servicio *SmartCardReaderService* quien crea una instancia de la clase *CPDI.Devices.CardReaders.HttpClientWrapper* que implementa métodos como *Connect* y *Disconnect*, con el fin de establecer la conexión remota con el lector conectado a la PC donde se llevará a cabo la

Capítulo 2. Propuesta de Solución

personalización; otro de los métodos que implementa es el *Transmit*, método encargado de enviar *APDUCommand* al chip, mediante una conexión directa con el *Device Grid Manager (DGM)*, el mismo contendrá tres clases fundamentales *SmartCardReaderClient*, *ISmartCardReaderController*, *SmartCardReaderCommon*. La primera de ellas, hace uso de una instancia de la segunda y mediante los métodos *Connect*, *Disconnect* y *SendCommand* que se comunica directamente con el lector de la tarjeta, con el propósito de invocar los servicios del lector. Estos métodos están implementados en la clase *SmartCardReaderCommon* y, responden con comandos *APDUResponse*, los cuales van devuelta al servidor y son tratados por el *SPDIWorkflowSmatCardPersonalization*.

2.6 Estimación de Tiempo

Los desarrolladores estiman cuánto tiempo es necesario para implementar cada una de las HU. Si es el tiempo para una HU supera las tres semanas, ésta se debe desglosar en otras más pequeñas. El valor del tiempo se expresa en semanas y en un principio no es totalmente exacto, pero con el transcurso de las iteraciones y el desarrollo del producto se irá acercando a la realidad. Esto se hace concluida la fase de exploración. A continuación la tabla correspondiente a la estimación de tiempo.

Historia de Usuario	Estimación
Gestión de la comunicación	1
Permitir la autenticación	1
Cargar e instalar el <i>Applet</i>	2
Creación de los DG	3
Creación del SOD	2

Tabla 14. Estimación de Tiempo de las Historias de Usuario

2.7 Plan de Iteraciones

Como parte del ciclo de vida de un proyecto usando la metodología XP se realiza un plan de duración de las iteraciones donde se muestra el orden y la duración de la implementación de cada HU dentro de cada iteración. Para la solución se han definido 7 HU divididas en 2 iteraciones, para una duración total del proyecto de 18 semanas. A continuación la tabla correspondiente a al plan de iteraciones.

Capítulo 2. Propuesta de Solución

Iteración	No.HU	Historia de Usuario	Duración en(Semanas)
Iteración1	HU_1	Gestión de la comunicación	8
	HU_2	Permitir la autenticación	
	HU_3	Cargar e instalar el <i>Applet</i>	
Iteración2	HU_4	Creación de los DG	10
	HU_5	Creación de los DG1	
	HU_6	Creación de los DG2	
	HU_7	Creación del SOD	

Tabla 15.Plan de iteraciones.

2.8 Plan de Entrega

Inmediatamente de tener concretadas las HU es necesario crear un plan de publicación. Éste es el resultado de una reunión entre los desarrolladores y los clientes. En él se recogen los tiempos de implementación ideales de las HU y la prioridad con la que serán implementadas. Para finalizar se propone una fecha para entregar el producto. Con el tiempo esta fecha puede variar, pero lo más seguro es que se acerque mucho a la planificación. De esta manera queda listo el cronograma de entregas en el cual también se deciden las fechas para liberar las versiones funcionales del producto. A continuación la tabla correspondiente al plan de entregas.

Entregable	Fin Iteración 1	Fin Iteración 2
Sistema Personalización de DLM-e	Enero 2013	mayo 2013

Tabla 16.Plan de Entregas

2.9 Tarjetas CRC

Tarjetas CRC correspondiente a la *HttpCardReaderClient.cs* del diagrama de clases del diseño. A continuación las CRC de la solución. Para consultar el resto de las CRC ([Ver anexo 4](#)).

Capítulo 2. Propuesta de Solución

Nombre de la Clase: SmartCardReader.Client	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. Connect 2. Disconnect 3. SendCommand 	<ol style="list-style-type: none"> 1. ISmartCardReader.Controller

Tabla 17. Tarjeta CRC de la clase SmartCardReader.Client.cs

Nombre de la Clase: ISmartCardReader.Controller	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. Connect 2. Disconnect 3. SendCommand 	<ol style="list-style-type: none"> 1. SmartCardReader.Common 2. SmartCardReader.Client

Tabla 18. Tarjeta CRC de la clase ISmartCardReader.Controller.cs

Nombre de la Clase: SmartCardReader.Common	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. Connect 2. Disconnect 3. SendCommand 	<ol style="list-style-type: none"> 1. ISmartCardReader.Controller

Tabla 19. Tarjeta CRC de la clase SmartCardReader.cs

Nombre de la Clase SPDIWorkflowOrder	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. DecrementElectricEngraveJob_ExecuteCode 2. IncrementElectricJobDefault_ExecuteCode 3. CA_UpdateOrderElectricWorkstation_ExecuteCode 4. On_OrderElectricAssignment_e1Property 5. On_OrderElectricAssignment_e1 6. UpdateElectricJob_ExecuteCode 7. On_RequestElectricEngravedDone_e1Property 8. On_RequestElectricEngravedDone_e1 	<ol style="list-style-type: none"> 1. SPDIWorkflowRequest.

Tabla 20. Tarjeta CRC de la clase SPDIWorkflowOrder.

Capítulo 2. Propuesta de Solución

Nombre de la Clase SPDIWorkflowRequest	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. SmartCardWFTType 2. ElectricPersonalizationSuccess 3. CA_ElectricEngravedSuccess_ExecuteCode 	<ol style="list-style-type: none"> 1. IEngraveElectricService 2. SPDIWorkflowOrder

Tabla 21. Tarjeta CRC de la clase SPDIWorkflowRequest.

Nombre de la Clase IEngraveElectricService	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. PrintRequests 2. Print 3. CancelPrint 4. EngravedStarted 5. EngravedDone 6. RequestEngravedDone 7. OrderEngravedDone 	<ol style="list-style-type: none"> 1. EngraveElectricService 2. SPDIWorkflowRequest.

Tabla 22. Tarjeta CRC de la clase IEngraveElectricService.

Nombre de la Clase EngraveElectricService	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. PrintRequests 2. Print 3. CancelPrint 4. EngravedStarted 5. EngravedDone 6. RequestEngravedDone 7. OrderEngravedDone 	<ol style="list-style-type: none"> 1. IEngraveElectricService 2. SPDIWorkflowSmartCardReaderPersonalization

Tabla 23. .Tarjeta CRC de la clase EngraveElectricService

2.10 Conclusiones parciales

- La profundización del objeto de estudio permitió la definición de los requisitos funcionales con los que debe cumplir el flujo.
- La especificación de los requisitos que debe cumplir el sistema permitió definir el uso del patrón arquitectónico basado en la arquitectura cliente/servidor para describir la vista lógica del sistema.
- La Estimación del tiempo que lleva la realización de la solución, permitió una mejor definición del tiempo empleado hasta la culminación del mismo.

Capítulo 3. Implementación y Prueba

Capítulo 3. Implementación y Prueba

3.1 Introducción

Después de las fases de exploración y planificación, la metodología XP define las fases de iteraciones y producción. En la Planificación se definieron las iteraciones y en cada iteración se diseñan, prueban y codifican cada una de las HU. El objetivo que se persigue en este capítulo es mostrar la evolución de la solución, explicar su diseño, así como los principales componentes y las relaciones que existen entre ellos, a través de diferentes diagramas los cuales ofrecen una panorámica del funcionamiento del sistema.

3.2 Estándares de codificación

Los estándares de codificación se definen por el equipo de desarrollo para lograr generalización en la programación del software y evitar errores comunes en cuanto al uso de las mayúsculas y al mismo tiempo asegurarse de que los programadores del proyecto trabajen de forma coordinada. Cuando el proyecto de software incorpore código fuente previo o bien cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación debería establecer cómo operar con la base de código existente. Un método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones del código de rutinas. A continuación se muestran algunos estándares de codificación.

Nombre	Descripción
Pascal	La primera letra en el identificador y la primera letra de cada subsiguiente palabra concatenada se capitalizan. Puede utilizar los identificadores de <i>Pascal case</i> en caso de tres o más caracteres. Por ejemplo: <code>BacInputData</code>
Evitando confusión de nombre y tipo	Utilizar nombres que describan a sus identificadores

Capítulo 3. Implementación y Prueba

Sensibilidad a mayúsculas	<ul style="list-style-type: none">• No se debe utilizar nombres o identificadores que requieran ser sensibles a las mayúsculas. No crear funciones con nombres de parámetros que se diferencian solo en el uso de la mayúscula.• No se debe crear <i>namespaces</i> con nombres de clases que se diferencien solo en el uso de las mayúsculas.• No crear clases con propiedades que se diferencien solo en el uso de las mayúsculas.• No crear clases con métodos que se diferencien solo en el uso de las mayúsculas.
Otros criterios	<ul style="list-style-type: none">• No debe dejarse de poner en el principio de cada método comentarios estándar, repetitivos, que indiquen el propósito del mismo.• Cuando se modifique el código, se mantendrán actualizados los comentarios pertinentes al cambio realizado.

Tabla 24. Estándares de codificación

3.3 Tratamiento de errores

En el diseño del sistema no solo debe tener en cuenta lo que debe ocurrir sino que, además, debe analizar las diferentes situaciones que se puedan presentar y que sin duda constituyen un error dentro del sistema.

Las excepciones son parte del diseño cuando aún la gran mayoría de ellas son identificadas en el proceso de implementación, no obstante es necesario definir con antelación un mecanismo efectivo para su tratamiento. En el caso del sistema de personalización es necesario llevar una traza de todas aquellas excepciones que se lancen y, en muchos casos, es necesario que el usuario que está interactuando con la aplicación especifique alguna información que tribute a la futura validación o corrección de la excepción lanzada. Pueden considerarse como parte del flujo de negocio en muchos procesos, por lo cual es necesario lograr identificar cada tipo de excepción para darle un tratamiento a cada una de ellas. En la solución se dio tratamiento a las excepciones identificadas en cada situación evitando errores dentro del flujo de trabajo.

1.4 Vista General del Workflow

Para ver de forma ampliada y más detallada todas sus actividades de la Vista General del *workflow* ([Ver Anexo 5 y 6](#)).

Capítulo 3. Implementación y Prueba

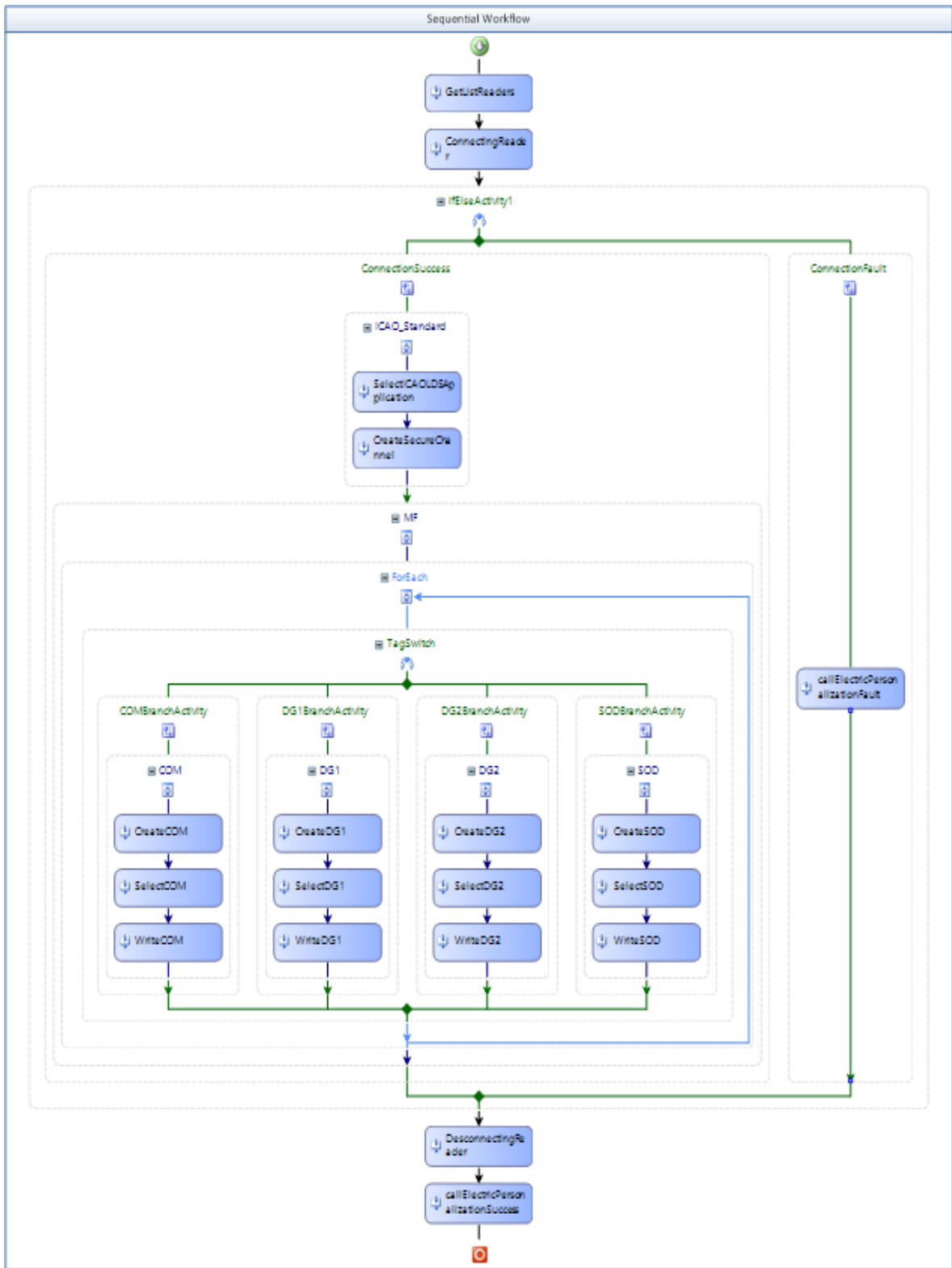


Figura 15. Diagrama de la Vista General de Workflow

Capítulo 3. Implementación y Prueba

3.5 Diagrama de Componentes

Un diagrama de componentes muestra las dependencias lógicas entre componentes software, sean éstos componentes fuentes, binarios o ejecutables.

Se consideran en este tipo de diagramas solo tipos de componentes y es representado como un grafo de componentes software unidos por medio de relaciones de dependencia, (generalmente de compilación), como se muestra en la Fig19.

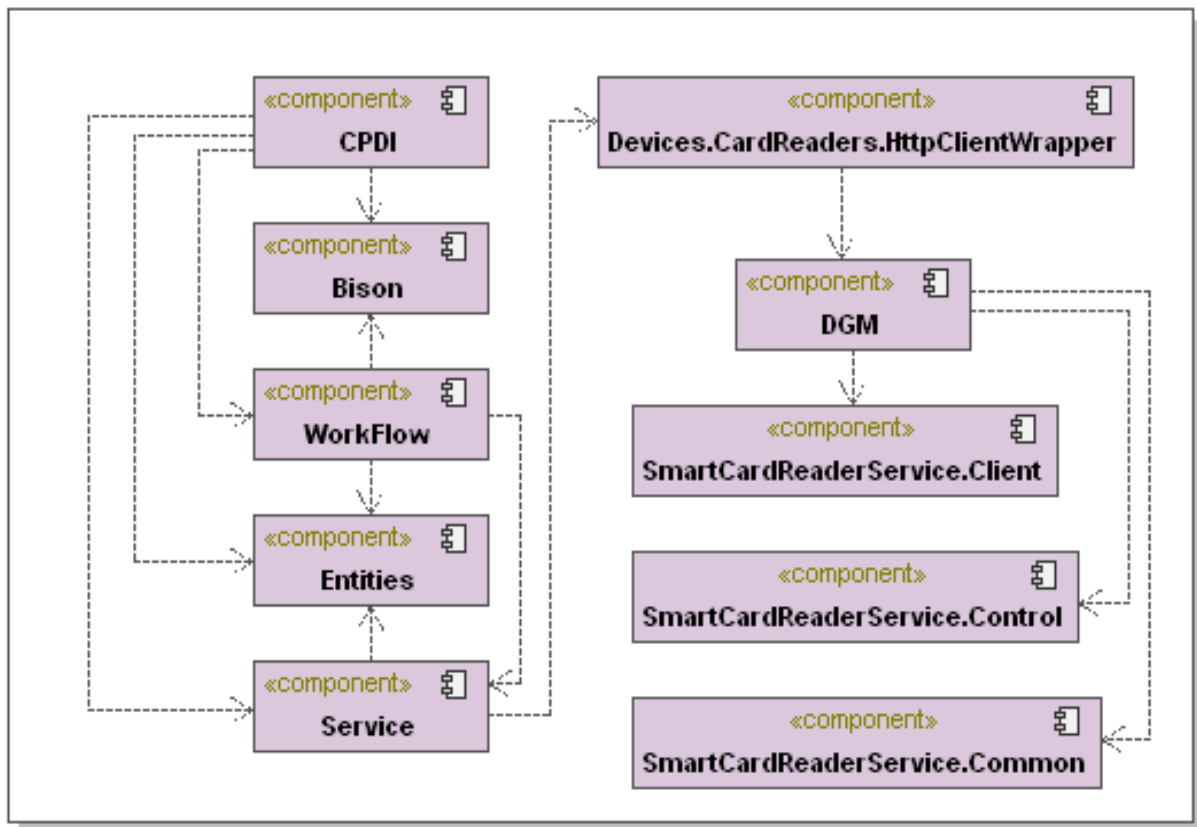


Figura 16. Diagrama de Componente

Puede mostrar también relación entre componentes software e interfaces soportadas. El diagrama de componentes se genera a partir del diagrama de clases. Por lo general los diagramas de componentes contienen:

- componentes
- interfaces
- Relaciones de dependencia, generalización, asociación y realización
- Paquetes o subsistemas (57).

Capítulo 3. Implementación y Prueba

Descripción del Diagrama de Componente

En el diagrama de componentes del flujo de trabajo para personalizar pasaportes-e, usando el sistema de personalización SPDI del Centro de Identificación y Seguridad Digital, refleja todos aquellos componentes que conforman la solución propuesta.

Se muestra la conformación del sistema por proyectos, que están constituidos por el componente: <<CPDI>>, el cual contiene todas las interfaces de usuario. Este componente hace referencia a otros tales como <<Bison.dll>> del CPDI que es la librería que gestiona y extiende el funcionamiento del Workflow, de ahí la referencia del mismo hacia el <<Bison.dll>>, a otro componente al que le hace referencia es el <<CPDI.Workflow.dll>> quien conoce la lógica de negocio para la personalización y facilita la realización de futuros cambios y, hace uso del componente <<CPDI.Service.dll>> que contiene los servicios necesarios para establecer la comunicación con el <<DGM>> que contiene el SmartCardReaderService.Client.dll, SmartCardReaderService.Common.dll, SmartCardReaderService.Control.dll y estos envían respuesta a la componente que lo invocó, acción que se efectúa mediante el componente <<CPDI.Devices.CardReaders.HttpClientWrapper.dll>>. Las entidades están contenidas dentro del componente <<CPDI.Entities.dll>> y éstas son invocadas por el componente <<CPDI.Service.dll>>.

3.6 Diagrama de Despliegue

Es un tipo de diagrama que muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Los diagramas de despliegue son el complemento de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema (58). Describen la topología del sistema la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP. En la siguiente figura se muestra el diagrama de Despliegue del sistema.

Capítulo 3. Implementación y Prueba

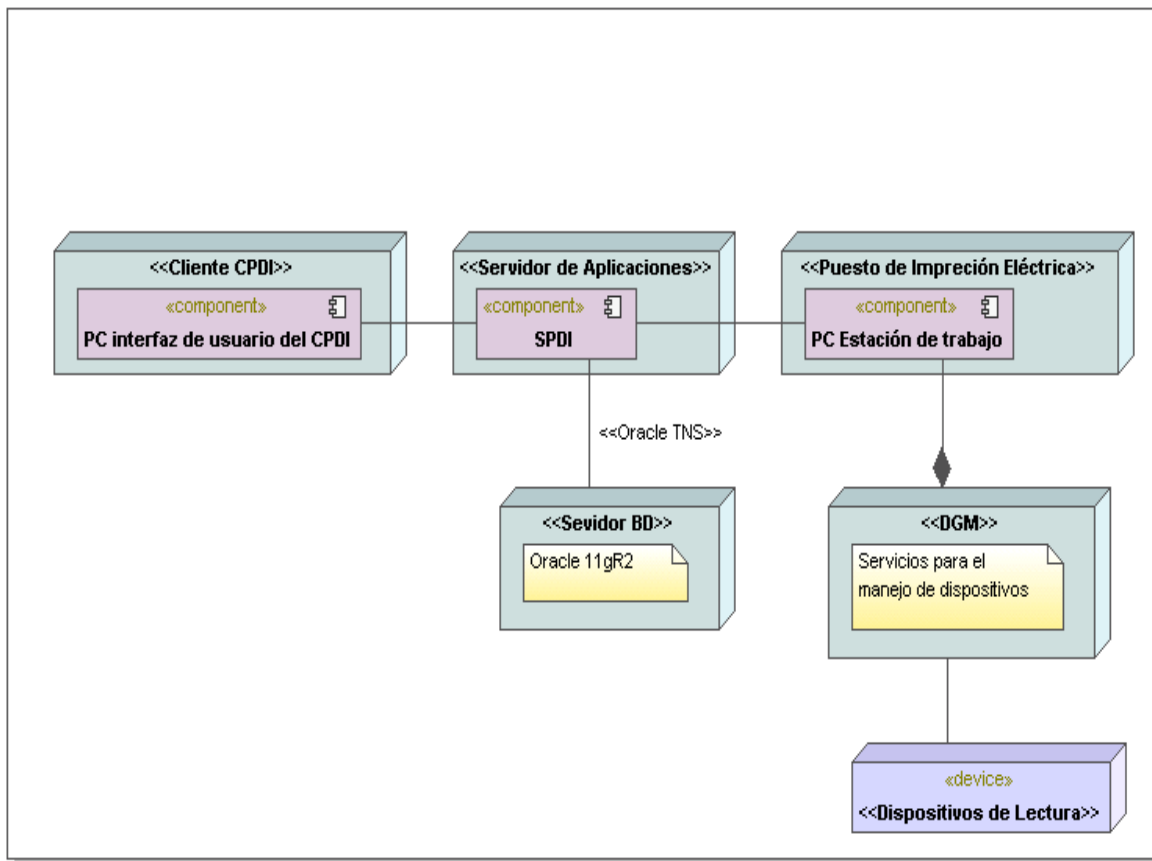


Figura 17. Diagrama de Despliegue

Descripción del Diagrama de Despliegue

El proceso de personalización de documentos se lleva a cabo mediante órdenes que arriban desde el «Cliente CPDI» al sistema a través un servicio web a la interfaz del SPDI, utilizando para la comunicación, el protocolo de «HTTP», dicho servicio es quien permite la comunicación con sistemas externos.

El «Puesto de Impresión Eléctrica» en el cual está instalado el DGM, que se encarga de la administración y manejo de los servicios del dispositivo de impresión eléctrica. Una vez recibidas las órdenes, se obtienen de los datos estableciendo una conexión con el servidor de base de datos a través del protocolo de conexión «Oracle TNS >>, proceso que se encuentra ya definido en el sistema existente. Acto seguido a esto se les inicia el proceso de personalización a cada orden recibida, procesando cada una de las solicitudes.

En el proceso de personalización de documento electrónico no solo realiza la personalización de la hoja de datos, esta incluye además la personalización del chip contenido en el documento, o sea efectúa la

Capítulo 3. Implementación y Prueba

personalización eléctrica y personalización gráfica. En este caso solo se ha enfocado la personalización del chip. En el estado de impresión se realizan la impresión grafica luego de haber finalizado este proceso se procede a la impresión eléctrica, mediante las estaciones de trabajo con los dispositivos adecuados para cada tipo de impresión.

3.7 Fase de producción

En la fase de producción el equipo de desarrollo se concentra en lo solicitado por el cliente, refinando el diseño y el código consecutivamente. El perfeccionamiento de código fue logrado a través de un grupo de pruebas unitarias automatizadas que ratificaron la ejecución correcta del sistema en todo el período de desarrollo.

El único instrumento adecuado para determinar el estado de la calidad de un producto de software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de verificar que el código está correcto y medir hasta qué grado el mismo cumple con los requerimientos. Pruebas que deben ser efectuadas varias veces puesto que se ven afectadas por los cambios que se van produciendo conforme se va desarrollando (59). Para la validación la propuesta de solución se aplicaron pruebas unitarias conocida (pruebas de caja blanca), así como las pruebas Funcionales o (prueba de caja negra).

3.8 Pruebas Unitarias

Las pruebas unitarias son llevadas a cabo por los programadores antes de empezar a codificar lo cual hará más sencillas y efectivas las pruebas finales. Se realizan reiteradamente a lo largo de todo el proyecto, asegurando siempre el funcionamiento correcto de cada componente por individual antes de realizar su integración. De este modo se evitan las ambigüedades y los requerimientos quedan perfeccionados en la prueba.

Sirven para utilizar otro código fuente, llamando directamente a los métodos de una clase pasando los parámetros apropiados y comparando los valores que se generan con los valores esperados. Los métodos de pruebas unitarias residen en clases Test¹⁷, que se almacenan en archivos de código fuente (60).

3.8.1 Prueba de Caja Blanca o Estructurales

Es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para llevar a cabo la derivación de los casos de prueba. Dichas pruebas pretenden garantizar que:

- Se ejecutan al menos una vez todos los caminos independientes de cada módulo.

¹⁷ Clase generada por la herramienta Visual Studio 2010 al realizarle pruebas unitarias a cualquier funcionalidad.

Capítulo 3. Implementación y Prueba

- Se utilizan las decisiones en su parte verdadera y en su parte falsa.
- Se ejecuten todos los bucles en sus límites.
- Se utilizan todas las estructuras de datos internas (61).

3.8.2 Diseño de Casos de prueba

En las figuras siguientes se muestran 2 funcionalidades del sistema a las que se les realizaron las pruebas.

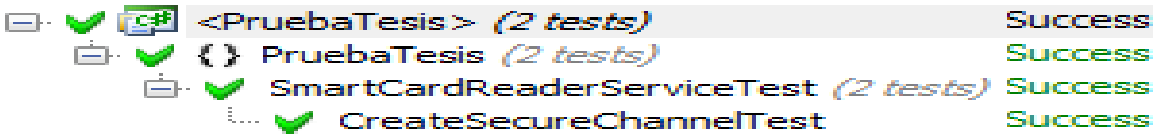
Prueba de Unidad		
NombrePrueba: CreateSecureChannel_Test.		
Estado: Satisfactoria.	Tipo: Caja Blanca.	ÚltimaEjecución: 25/05/2013
Ejecutada por: Yaumara Cruzata		
Descripción: Para que la ejecución de esta prueba tenga resultados satisfactorios se deben proporcionar los parámetros necesarios.		
Entrada: bacimputData		
Criterio de aceptación: Establecer un canal seguro		
Resultado:		
		

Tabla 25. Método **CreateSecureChannelTest** al que se le realizó la prueba unitaria.

Prueba de Unidad		
NombrePrueba: TransmitTest.		
Estado: Satisfactoria.	Tipo: Caja Blanca.	ÚltimaEjecución: 25/05/2013
Ejecutada por: Yaumara Cruzata		
Descripción: Para que la ejecución de esta prueba tenga resultados satisfactorios se deben proporcionar los parámetros necesarios.		

Capítulo 3. Implementación y Prueba

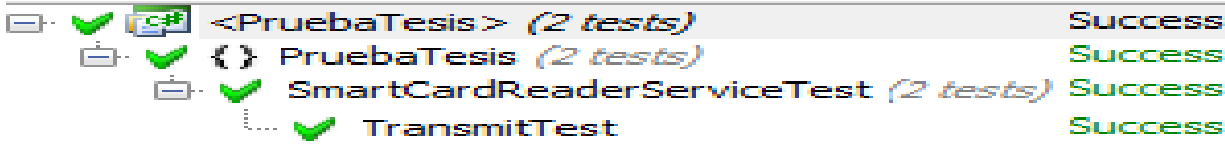
Entrada: comando APDU
Criterio de aceptación: Se crea el DG correspondiente
Resultado:  <p>The screenshot shows a test runner interface with a tree view of test results. At the top level, there is a folder icon with a green checkmark and the text '<PruebaTesis> (2 tests) Success'. Below it, there is a sub-folder icon with a green checkmark and the text 'PruebaTesis (2 tests) Success'. Underneath that, there is another sub-folder icon with a green checkmark and the text 'SmartCardReaderServiceTest (2 tests) Success'. Finally, at the bottom level, there is a test icon with a green checkmark and the text 'TransmitTest Success'.</p>

Tabla 26. Método **TransmitTest** al que se le realizó la prueba unitaria

3.9 Pruebas Funcionales

Las pruebas de aceptación buscan discrepancias entre el programa y sus objetivos o requerimientos, enfocándose en los errores hechos durante la transición del proceso al diseñar la especificación funcional. Constituyen un proceso vital dentro de las revisiones efectuadas a un sistema que significan la satisfacción del cliente con el producto desarrollado. Las pruebas del sistema tienen un propósito particular: comparar y asegurar que las funcionalidades del sistema cumplen con lo que se espera de ellas (62).

Estas pruebas son creadas a partir de las HU. El cliente o usuario especifica los aspectos a testear cuando una historia de usuario ha sido correctamente implementada. La HU puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento y no se considera completa hasta que no supera sus pruebas de aceptación. Es responsabilidad del cliente verificar la corrección de las pruebas y tomar decisiones acerca de las mismas.

3.9.1 Pruebas de Caja Negra

Las pruebas de caja negra se efectúan sobre la interfaz del sistema, observando las respuestas ante determinadas acciones y los datos de salida para comprobarlos con los datos de entrada, obviando el comportamiento interno y la estructura del programa. Este tipo de pruebas pretenden encontrar estos tipos de errores:

- Funciones incorrectas o ausentes
- Errores en la interfaz
- Errores en estructuras de datos o en accesos a bases de datos externas
- Errores de rendimiento

Capítulo 3. Implementación y Prueba

- Errores de inicialización y de terminación (39).

Dentro del método de caja negra la técnica de partición de equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software. Descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición de equivalencia se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así en número de clases de prueba que hay que desarrollar. A continuación se muestra un caso de prueba utilizando la técnica de partición de equivalencia.

3.9.2 Diseño de Casos de prueba

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. Los casos de prueba deben verificar si el producto satisface los requisitos del usuario, tal y como se describen en la especificación de estos y si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño (63).

A continuación el Caso de Prueba del Requisito funcional Gestionar Personalización que encierra obtener los datos personales, crear sistema de ficheros ISO/IEC 7816-4, crear los grupos de datos, según OACI, crear el DG1, crear el DG2, crear el SOD e introducir datos personalizados al chip.

CPR 1 Gestionar Personalización

Condiciones de Ejecución

El usuario debe haber adquirido los permisos necesarios para la realización de las operaciones.

ID del Escenario	Tipos de Clases	Clases	Resultado Esperado	Resultado de la Prueba	Observaciones
Obtener los datos personales	Válidas	Seleccionar operación Obtener los datos personales	Se realiza la obtención de los datos a personalizar en el documento.	Datos obtenidos.	Satisfactorio.
	Inválidas	No permite obtener los datos personales	No se realiza la obtención de los datos a personalizar en el documento.		

Capítulo 3. Implementación y Prueba

Crear sistema de ficheros ISO/IEC 7816-4	Válidas	Seleccionar operación crear el sistema ficheros de donde se almacenaran los datos.	Mostrar los datos del sistema de fichero.	Se mostraron los datos satisfactoriamente.	Satisfactorio.
	Inválidas	No permite crear el sistema de ficheros	No muestra los datos para del sistema de fichero.		
Crear el GD1	Válidas	Se realiza la creación del GD1	Permite crear el GD1	Se mostraron los datos satisfactoriamente	Satisfactorio.
	Inválidas	No permite crear el GD1	No permite crear el GD1		
Crear el GD2	Válidas	Se realiza la creación del GD2	Permite crear el GD2	Se mostraron los datos satisfactoriamente	Satisfactorio.
	Inválidas	No permite crear el GD2	No permite crear el GD2		
Introducir datos personalizados al chip	Válidas	Se realiza la inclusión de los datos personalizados en el chip	Se realiza la personalización del documento	Documento personalizado	Satisfactorio
	Inválidas	No se realiza la inclusión de los datos personalizados en el chip	No se realiza la personalización del documento		

Tabla 27. Diseño de Caso de Prueba del requisito funcional "Gestionar Personalización".

3.9.3 Resultado de las pruebas

El resultado de las pruebas del sistema es una parte primordial en el desarrollo porque brinda la posibilidad de ver que tan bien está el desarrollo del sistema con respecto al cumplimiento de los requerimientos funcionales especificados.

Capítulo 3. Implementación y Prueba

Se realizaron tres iteraciones de pruebas con el objetivo de encontrar la mayor cantidad de no conformidades existentes y darle solución a las mismas. Durante la primera iteración fueron encontradas un total de 14 no conformidades, las cuales se subsanaron en esa etapa. Seguidamente se realizó la segunda iteración donde se comprobó que todos los errores de la iteración anterior habían sido corregidos, además esta iteración permitió encontrar siete no conformidades que no habían sido detectadas. Posteriormente se llevó a cabo la tercera y última iteración en la cual fueron encontradas otras tres no conformidades las cuales fueron tratadas, con lo cual se evidenciaba la disminución de las no conformidades.

A continuación se muestra cómo se fueron corrigiendo las no conformidades detectadas en las iteraciones. Donde cada iteración subsana las no conformidades de la iteración anterior.

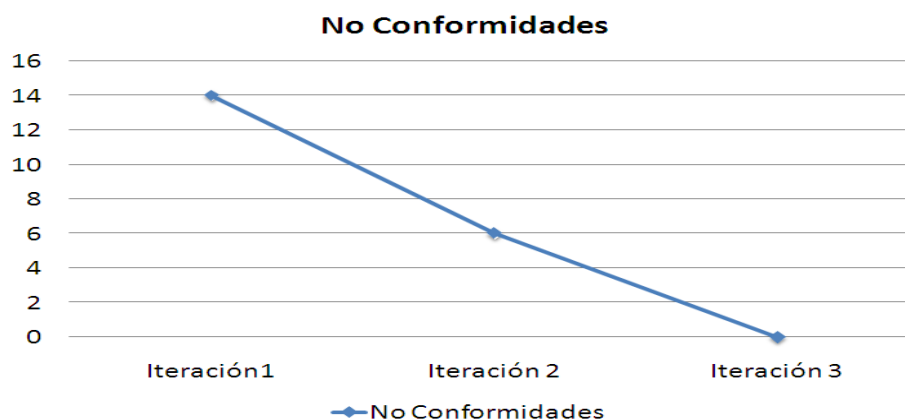


Tabla 28. Tabla de no Conformidades

3.11 Conclusiones Parciales

- La realización de los diferentes diagramas permitió obtener un enfoque detallado del sistema, el diagrama de despliegue, permitió visualizar las dependencias lógicas de los componentes así como la distribución física de los nodos del sistema
- Las pruebas unitarias posibilitaron evaluar las respuestas de funcionalidades implementadas, en tanto las pruebas de caja negra permitieron comprobar la correcta ejecución de los requerimientos funcionales previamente definidos, posibilitando de esta manera localizar y subsanar las no conformidades detectadas dentro del sistema.

Conclusiones Generales

Conclusiones Generales

Con el desarrollo de este trabajo se ha podido demostrar cómo se han cumplido los objetivos principales definidos para la creación de un flujo de trabajo que permita la personalización de documentos electrónicos de viaje integrable al sistema de personalización de documentos de lectura mecánica; proporciona a CISED un sistema para la personalización más completo. Algunas de las conclusiones arribadas luego de terminada la primera versión de este producto son:

- La aplicación de los métodos teóricos y el análisis bibliográfico permitieron concretar el marco teórico con el uso actual de los pasaportes-e.
- El análisis del SPDI evidenció la imperante necesidad de desarrollar el sistema propuesto.
- El análisis de las tecnologías según la necesidad existente, permitió determinar que la utilización de la tecnología de *workflow* es la más adecuada para el desarrollo de la solución.
- El análisis de las soluciones existentes y la profundización del objeto de estudio permitió la definición de los requisitos funcionales con los que debe cumplir el flujo.
- La ejecución de las pruebas posibilitó la validación del correcto funcionamiento del sistema.

Recomendaciones

Recomendaciones

Al finalizar la presente investigación, se plantean las siguientes recomendaciones para las siguientes versiones de la solución:

- Integrar el flujo de trabajo para la personalización de documentos electrónicos de viaje con el Sistema de Administración de Llaves del centro CISED.

Bibliografía Referenciada

Bibliografía Referenciada

1. **Krostanje, Maximiliano.** *APORTES DE LOS VIAJES A LAS CIENCIAS SOCIALES.* Universidad de Católica de Argentina Buenos Aires, Argentina Gestion Turística : s.n., 2008. No 8ISSN 0717-1811.
2. **C., De I. Byron Puga.** *IAEN, Implementación de de una fabrica de pasaporte de lectura mecánica y/o de pasaporte electrónicos en el Instituto Geográfico Militar.* República de Ecuador : s.n., 2009.
3. **Naciones Unidas-Centro de Información (Cuba ,México y Republica Dominicana).** *Naciones Unidas-Centro de Información (Cuba ,México y Republica Dominicana).* [Online] [Cited: mayo 17, 2013.] <http://www.cinu.org.mx/onu/estructura/organismos/oaci.htm>.
4. **OACI. Organizacion de Aviacion Civil Internacional.** *Organizacion de Aviacion Civil Internacional.* [Online] [Cited: mayo 17, 2013.] <http://www.monografias.com/trabajos7/oaci/oaci.shtml> .
5. **Alina Surós Vicente, Reynier Lester Claro Escalona.** *Directorio de llaves públicas de la OACI.* 2010.
6. **OACI.** *Documentos de viaje de lectura mecánica, Parte 1 Volumen 1.* s.l. : Sexta Edición, 2006.
7. **Nelson Hernández Guerra, Adrián Martínez Pérez.** *Applet y su componente middleware para la gestión de información en los pasaportes electrónicos.* Ciudad de La Habana : s.n., 2011.
8. **SPDI.** *SPDI-Resumen.* Ciudad de La Habana : s.n., 2013.
9. **Areitio, Javier y Teresa.** *Análisis en torno a la tecnología biométrica para los sistemas electrónicos de identificación y autenticación.* Facultad de Ingeniería. ESIDE. Universidad de Deusto Universidad del País Vasco (UPV/EHU) : s.n., 2007.
10. **Aalast W., Hee K.** *Workflow Management Models, Methods, and Systems MI TPress.* 2002.
11. **Mira, José Joaquín, Gómez, José María y Blay, Inma.** *La gestión de procesos.* 2006.
12. **Morales, Ing. Pablo.** *Arquitectura de procesos para modelos de Workflow.* 2002.
13. **DATYS.** *EMIPAS .Sistema Cubano de Emisión de Pasaporte.* 2013.
14. **Pérez1, Alexeis García.** *ASIMED, La gestión de documentos electrónicos como respuesta a las nuevas condiciones del entorno de información.* Ciudad de La Habana : s.n., 2001.
15. **Dr. Víctor J. Sosa Sosa.** *MIDDLEWARE: Arquitectura para Aplicaciones Distribuidas.* 2007.
16. **Luis E. Mendoza, LISI,** Universidad Simón Bolívar. *Estudio de Tecnologías Middleware para Sistemas Peer-to-Peer* lisi.usb.ve/.../07%20integracion%20de%20sistemas/integracion_17.pdf. 2011.
17. **Pereda Viñolo Katerina, Fernández Santana Vismar.** *Plataforma para el desarrollo de servicios en línea .* Ciudad Habana : s.n., 2010.
18. **Torres, Jorge.** *ViaFirma.MAGAZINE .Firma electronica en dispositivos móviles.* 2008. Numero 1.

Bibliografía Referenciada

19. Daniel Perovich, Leonardo Rodríguez y Martín Varela. *Proyecto de Taller V (2000) Programación de JavaCards -Instituto de Computación Facultad de Ingeniería Universidad de la República. 2001.*
20. Alva, Luis Roberto Santizo. *LLAVE PÚBLICA EN TARJETA INTELIGENTE Universidad de San Carlos de Guatemala ,Facultad de Ingeniería ,Escuela de Ingeniería en Ciencias y Sistemas. Guatemala : s.n., 2005.*
21. Pozo, Luis Salvador. *HERRAMIENTAS DE VALIDACIÓN DE CERTIFICADOS EN PKI CON TARJETA INTELIGENTE -INGENIERÍA SUPERIOR DE TELECOMUNICACIÓN. 2009.*
22. Vicente, Alina Surós. *El pasaporte electrónico -Electronic passport.*
23. ORGANIZATION, INTERNATIONAL CIVIL AVIATION. *DEVELOPMENT OF A LOGICAL DATA STRUCTURE - LDS For OPTIONAL CAPACITY EXPANSION TECHNOLOGIES. 2004.*
24. Europa, Seguridad documental Firma digital y DNI en. *Seguridad documental Firma digital y DNI en Europa -www.ksitdigital.com/documentacion/descargas/dlibroksi_small. 2008.*
25. Tom A.F. Kinneging for ICAO-NTWG, PKI Task Force. *PKI Digital Signatures For Machine Readable Travel Documents. 2004.*
26. Iternacional, Organización de Aviación Civil. *Documentos de viaje de lectura mecánica Parte 1-Volumen 2. s.l. : Sexta edición, 2006.*
27. ISO/IEC. *ISO/IEC 7816-4 INTERNATIONAL STANDARD. 2005.*
28. Angamarca, Walter Pupiales. *Diseño de un Sistema de Control de acceso utilizando la tecnologia de identificacion FRID para la empresa de soluciones Gcuatro del Ecuador CIA.LTDA. Quito : s.n., 2009.*
29. Alexandro. *Tarjetas Inteligentes. www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.../A5.pdf. 2006.*
30. Junction, Princeton. *Acceso Lógico Seguro: El Papel de las Tarjetas Inteligentes en una Autenticación Más Sólida. 2004. ID-04002.*
31. Fajardo, DiegoTadeo. *Socialización por medio de texos electrónicos. Bogotá -Colombia : s.n., 2003.*
32. Bundesdruckerei. *A Bundesdruckerei Pocket Guide To ePassport Systems. 2007.*
33. Fernández, Yanisleivi Valdés. *MANUAL DE USUARIO-Sistema de Personalización de Documentos de Identificación (SPDI). 2011.*
34. Zambrano, Ing. Reynier Blanco. *Documento de Arquitectura-Identificación, Inmigración y Extranjería de la República de Cuba. 2009.*
35. ID Systems. Bundesdruckerei. 2006 : s.n.
36. UCI, Proyecto Identidad Venezuela. *Proyecto para la transformación y modernización del sistema de Identificación, Inmigración y Extranjeria de la República Bolivariana de Venezuela. 2005.*
37. Carmen, Penadés Patricio Letelier y M^a. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Universidad Politécnica de Valencia : s.n., 2004.*

Bibliografía Referenciada

38. Escribano, Gerardo Fernández. *Introducción a Extreme Programing -Ingeniería de SoftwareII*. 2002.
39. Pressman, Roger S. *Ingeniería del Software. Un enfoque práctico* ISBN:0-07-709677-0. 2002.
40. Altova. *Altova UModel*. 2010.
41. Herramienta CASE para diseño con UML y desarrollo de software. [Online] [Cited: mayo 17, 2013.] www.sparxsystems.com.ar/products/ea/index.html.
42. Iglesias, Raul Fernández y Josefa. *Grupo B. Sistemas y Tecnologías de la información - Instituto* - www.ign.es/ign/resources/acercaDe/aig/B.pdf.
43. Dariel Ricardo Pérez Bandomo, Onel Salinas Peña y José Antonio Ballate Oquendo. *Soluciones informáticas -Modernización del proceso tramitación de carné de extranjeros de la Dirección de Inmigración y Extranjería de la República de Cuba*. 2012.
44. Internelia. Canal Visual Basic.net. [Online] [Cited: Enero 12, 2013.] <http://www.canalvisualbasic.net/manual-net/c-sharp/#cSharp>.
45. MSDN. [MSDN.Microsoft.com/es-es/library/.../ms735119\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/.../ms735119(v=vs.90).aspx) -España. [Online] [Cited: Enero 13, 2013.]
46. Danysoft, Servicios profecionales. *Catálogo Cursos Microsoft*. 2010.
47. Mylopoulos, A. Gal and J. *Supporting Distributed Autonomous Information Services Using Coordination.International Journal of Cooperative Information*. 2009.
48. Proyecto, Arquitectura de. *Sistema Único de Identificación Nacional de la Población de la República de Cuba*. 2009.
49. Vallejo, Marco Salazar y Daniel. *ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DEL CATASTRO TURÍSTICO PARA LA EMPRESA PÚBLICA METROPOLITANA DE GESTIÓN DEL DESTINO TURÍSTICO DE LA CIUDAD DE QUITO*.
50. Gemalto. Gemalto.2010. DeveloperSuite. [Online] 2010. [Online] 2010. [Cited: Enero 14, 2013.] [http://www.gemalto.com/..](http://www.gemalto.com/)
51. Hall., Craig Larman. Prentice. [Online] 2003. [Cited: abril 24, 2013.] [.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf](http://ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf).
52. Ing.Sooft. *Cuerpo del conocimiento de la Ing.Sooft. Capítulo 2 Requerimientos de Software*. 2006.
53. Superior, Escuela Politécnica. *Patrones de Diseño, TACCII*. Curso 2008/2009.
54. Kiccillof, Carlos Reynoso y Nicolás. *Estilos y Patrones en la Estrategia de arquitectura de Microsoft*. 2004.
55. Urudata. *How to Q-flow Patrones básicos de Workflow Versión : 2 .0 m*. 20 11.
56. Milviuris Villariño Pelier, Erislandys Premiot Laffita. *Sistema de autenticación de usuarios por medio de tarjetas inteligentes y códigos de identificación personal* . Ciudad de la Habana : s.n., 2010.

Bibliografía Referenciada

57. Cruz Quispe Víctor Fabio, Gutiérrez Mamani Ever Dino y Mendivil Torrico Luís Briam. Universidad Sales Iana Bolivia. [Online] [Cited: abril 2, 2013.] virtual.usalesiana.edu.bo/web/practica/archiv/componen1.ppt.
58. Bolivia, Universidad Sales Iana. [Online] [Cited: abril 2, 2013.] virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc.
59. Pruebas de Software. [Online] [Cited: abril 1, 2013.] <http://www.pruebasdesoftware.com/laspruebasdesoftware..>
60. MSDN, Microsoft Corporation. Microsoft Corporation. MSDN. [Online] [Cited: abril 2, 2013.] [msdn.microsoft.com/es-es/library/ms182532\(v=vs.80\).aspx](http://msdn.microsoft.com/es-es/library/ms182532(v=vs.80).aspx) - España.
61. Almería, Universidad de. Universidad de Almería. [Online] [Cited: abril 1, 2013.] <http://indalog.ual.es/mtorres/LP/Prueba.pdf..>
62. Dayvis Malfará, Diego Cukerman, Fernando Cócaro, Juan Pablo Cassinelli y Renzo Séttimo. [Online] [Cited: abril 22, 2013.] www.fing.edu.uy/inco/cursos/gestsoft/.../Testing%20en%20XP..
63. Ivar Jacobson, Booch Grady y James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*. s.l. : s.l. : Addison Wesley, 2000.
64. Obregón, Linnett Galens Ameneiro y Dannier Sierra. *Sistema de Personalización de Documentos de Identificación de la República de Cuba*. Ciudad de La Habana : s.n., 2010.

Glosario de Términos

Glosario de Términos

APDU: Unidades de Datos en la Capa de Aplicación. Los APDU pueden contener un mensaje de comando o un mensaje de respuesta.

Applet: Aplicación que se ejecutan dentro de las *Smartcards* y gestiona la información almacenada en ellas.

DVLM: Documento de viaje de lectura mecánica.

DF: Fichero dedicado.

DG: Data Group.

EF: Fichero elemental.

ICP: Infraestructura de clave pública para permitir la detención de manipulación indebida de los datos de un pasaporte-e.

ISO: (*International Organization for Standardization*): La Organización Internacional para la Normalización es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional.

LDS: La estructura lógica de datos que describe la forma en que han de ingresarse y formatearse los datos biométricos en los pasaportes-e.

MF: Fichero maestro.

Pasaportes electrónicos: Pasaporte de Lectura Mecánica (PLM) que contiene una microplaqueta de circuito integrado (CI) sin contacto dentro de la cual se almacenan datos de la página de datos del pasaporte-e, elementos biométricos del titular del pasaporte y un objeto de la seguridad para proteger los datos con tecnología criptográfica de infraestructura de clave pública ICP y que se ajusta a las especificaciones del Doc9303, Parte1.

Personalización: Proceso que permite a los funcionarios de las oficinas emisoras procesar las solicitudes de pasaportes, mediante una secuencia de opciones, que hacen posible la producción del documento.

Glosario de Términos

Portador: Persona que un porta un pasaporte-e o identificación al reclamar una identidad legitima o falsa.
Persona que interactúa con un sistema biométrico para inscribirse o para que se verifique su identidad.

SPDI: Sistema de Personalización de Documentos de Identificación.

TCI: Tarjeta de Circuito Integrado, tarjeta que contiene una microplaqueta electrónica acoplada a una antena que permite comunicar datos entre la microplaqueta y un dispositivo de codificación /lectura sin necesidad de eléctrica directa.

Workflow: Es el conjunto de actividades o tareas realizadas en secuencia o en paralelo por dos o más miembros de un equipo de trabajo para lograr un objetivo común siguiendo unas reglas de negocio preestablecidas.

ZIV: Las partes del DVLM (páginas de datos en el caso del PLM), o sea, el frente y el reverso (cuando corresponda), que no se definen como ZLM.

ZLM: Área de dimensiones fijas situada en la página de datos del DVLM que contiene los datos obligatorios y opcionales ordenados de forma que puedan ser leídos mecánicamente con métodos OCR.

ANEXOS

ANEXO 1.Imágenes de Datos OCR.



Figura 18.Ejemplo1 de Datos OCR

<p>OCR-A</p> <p>1234ABCD</p> <p>Alphanumeric (+4 currency char.)</p>	<p>OCR-B</p> <p>1 2 3 4 A B C D</p> <p>Alphanumeric (+4 currency char.)</p>
<p>MICR E-13B</p> <p>1 2 3 4 0 1 2 3 4 5 6 7 8 9</p> <p>Numeric (+4 special char.)</p>	<p>SEMI M12</p> <p>1234ABCD</p> <p>Alphanumeric (+4 currency char.)</p>

Figura 19.Ejemplo2 de Datos OCR

ANEXO 2. Descripción de la Arquitectura del sistema SPDI.

Capa de Presentación

Está compuesta por todas las interfaces de usuario y los componentes necesarios para su correcto funcionamiento. Estos elementos pueden ser ficheros JavaScript, CSS, servicios web, interfaces de usuario, etc. Esta capa se encuentra representada por el proyecto web de la aplicación, y tiene interacción directa con la capa de Procesos y Servicios y con la de Negocio.

Capa de Procesos y Servicios

Tiene contenida la lógica de los procesos de negocio representados por los *workflows*, las actividades que por su nivel de reutilización o importancia lógica se encapsulan en una actividad propia del proyecto y los servicios de tiempo real, los cuales son los encargados de la interacción con los *workflows* definidos, estos a su vez interactúan con los servicios de la capa de negocio haciendo uso para ello de una fábrica de servicios que le devuelve la instancia requerida. Al mismo nivel se encuentra definida una fábrica para los servicios de tiempo real, que le permite a la capa de presentación interactuar con estos sin necesidad de que se sepa la instancia del servicio que utilizan. Todos estos elementos se encuentran vinculados directamente con el BisonFramework.

Capa de Negocio

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

Capa de Acceso a datos

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

Capa de Base de Datos

Por la importancia que posee el desarrollo de procedimientos y vistas dentro del negocio, se ha decidido separar la base de datos en una nueva capa, donde se concentra una pequeña parte de la lógica de las funcionalidades dentro de la aplicación. La única capa encargada de interactuar con esta es la de Acceso a datos (64).

ANEXO 3. Diagrama de Clases.

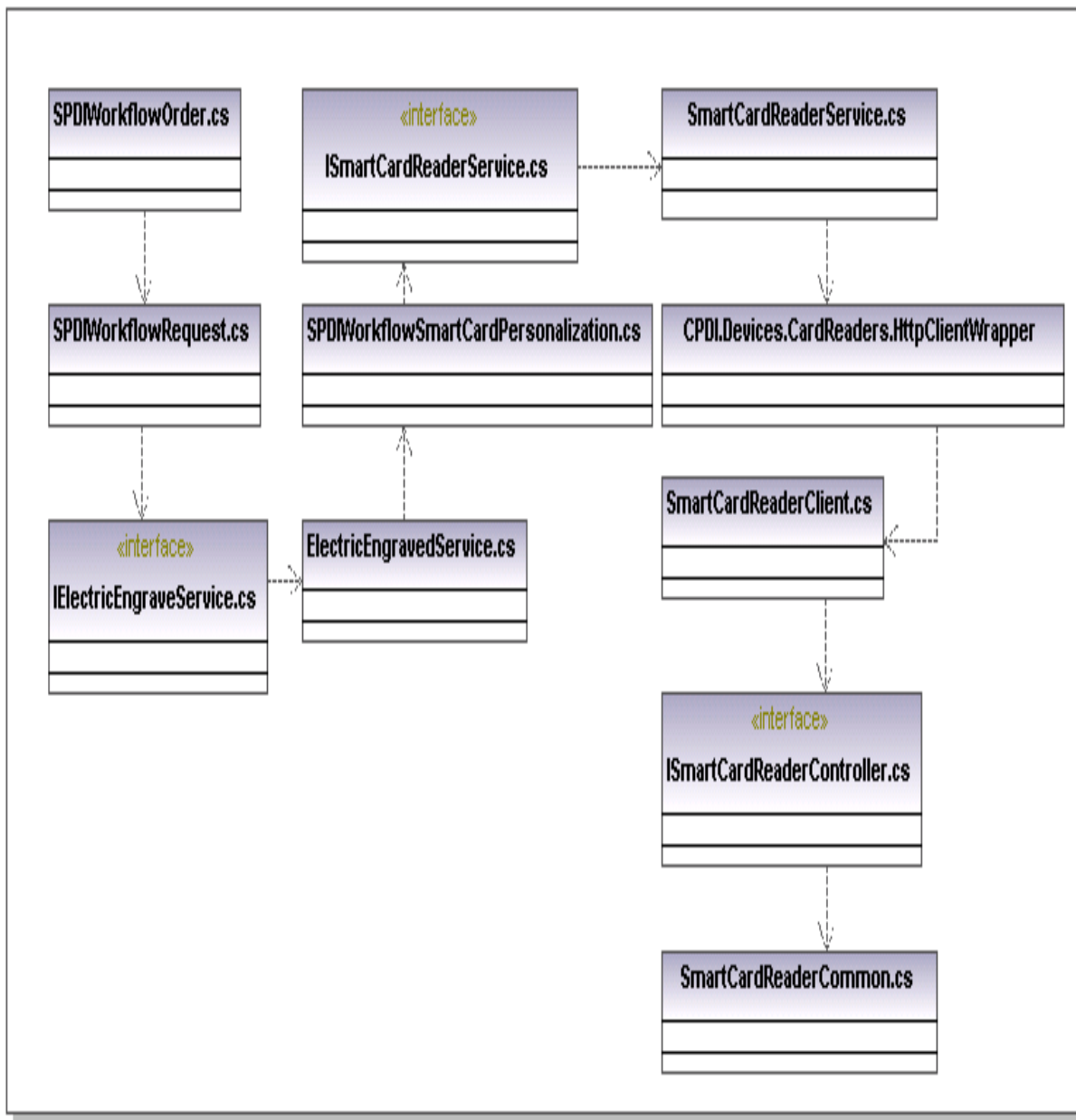


Figura 20. Diagrama de Clases con los métodos

ANEXO 4.Tarjetas CRC.

Nombre de la Clase ISmartCardReaderService	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. Connect 2. Disconnect 3. Transmit 4. ListReaders 5. CreateSecureChannel 6. SendSecureAPDU 7. StartPersonalization 8. ElectricPersonalizationSuccess 9. ElectricPersonalizationFault 	<ol style="list-style-type: none"> 1. SPDIWorkflowSmartCardReaderPersonalization 2. SmartCardReaderService

Tabla 29.Tarjeta CRC de la clase ISmartCardReaderService.

Nombre de la Clase SPDIWorkflowSmartCardReaderPersonalization	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. ApduCommandSelectICAOLDSApplication 2. ApduResponseSelectICAOLDSApplication 3. ApduCommandCreateCOM 4. ApduResponseCreateCOM 5. ApduCommandSelectCOM 6. ApduResponseSelectCOM 7. ApduCommandWriteCOM 8. ApduResponseWriteCOM 9. ApduCommandCreateDG1 10. ApduResponseCreateDG1 11. ApduCommandSelectDG1 12. ApduResponseSelectDG1 13. ApduCommandWriteDG2 14. ApduResponseWriteDG2 15. ApduCommandCreateDG2 16. ApduResponseCreateDG2 17. ApduCommandSelectDG2 18. ApduResponseSelectDG2 19. ApduCommandWriteDG2 20. ApduResponseWriteDG2 21. ApduCommandCreateSOD 22. ApduResponseCreateSOD 23. ApduCommandSelectSOD 24. ApduResponseSelectSOD 25. ApduCommandWriteSOD 26. ApduResponseWriteSOD 	<ol style="list-style-type: none"> 1. EngraveElectricService 2. ISmartCardReaderService

Tabla 30.Tarjeta CRC de la clase SPDIWorkflowSmartCardReaderPersonalization.

Nombre de la Clase SmartCardReaderService	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. Connect 2. Disconnect 3. Transmit 4. ListReaders 5. CreateSecureChannel 6. SendSecureAPDU 7. StartPersonalization 8. ElectricPersonalizationSuccess 9. ElectricPersonalizationFault 	<ol style="list-style-type: none"> 1. ISmartCardReaderService 2. CPDI.Divices.SmartCardReadersHttpClientWrapper

Tabla 31. Tarjeta CRC de la clase SmartCardReaderService.

Nombre de la Clase CPDI.Divices.SmartCardReadersHttpClientWrapper	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. HttpCardReaderClient 2. BuildUri 3. InvokeWebService 4. BeginTransaction 5. EndTransaction 6. GetAttribute 7. ListReaders 8. StartCardEvents 9. StopCardEvents 10. Connect 11. Disconnect 12. Transmit 	<ol style="list-style-type: none"> 1. SmartCardReaderService 2. SmartCardReader.Client

Tabla 32. SmartCardReaderService CPDI.Divices.SmartCardReadersHttpClientWrapper.

ANEXO 5.Vista General del Workflow. Parte1

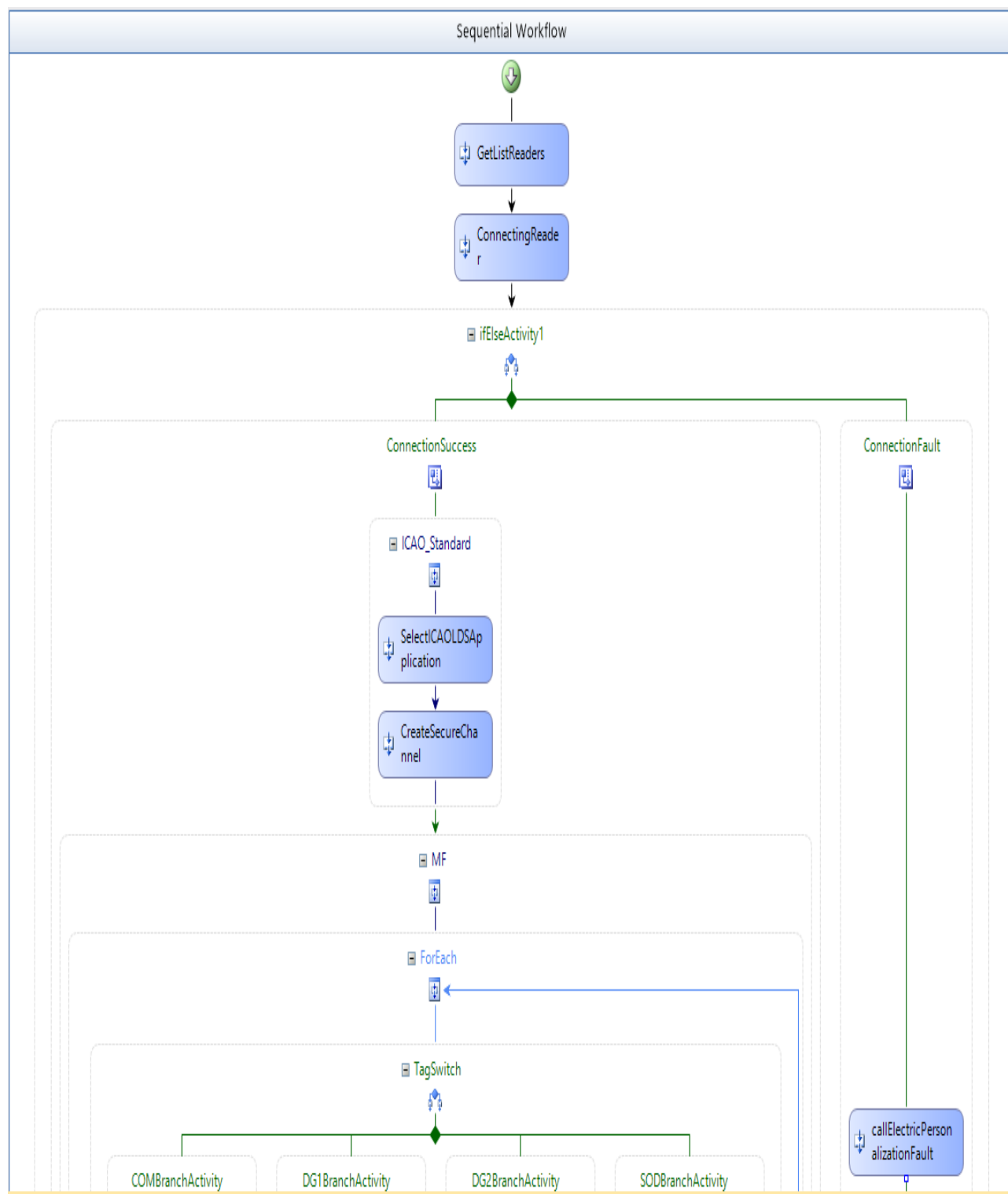


Figura 21.Vista General del Workflow parte1

ANEXO 6.Vista General del Workflow. Parte2

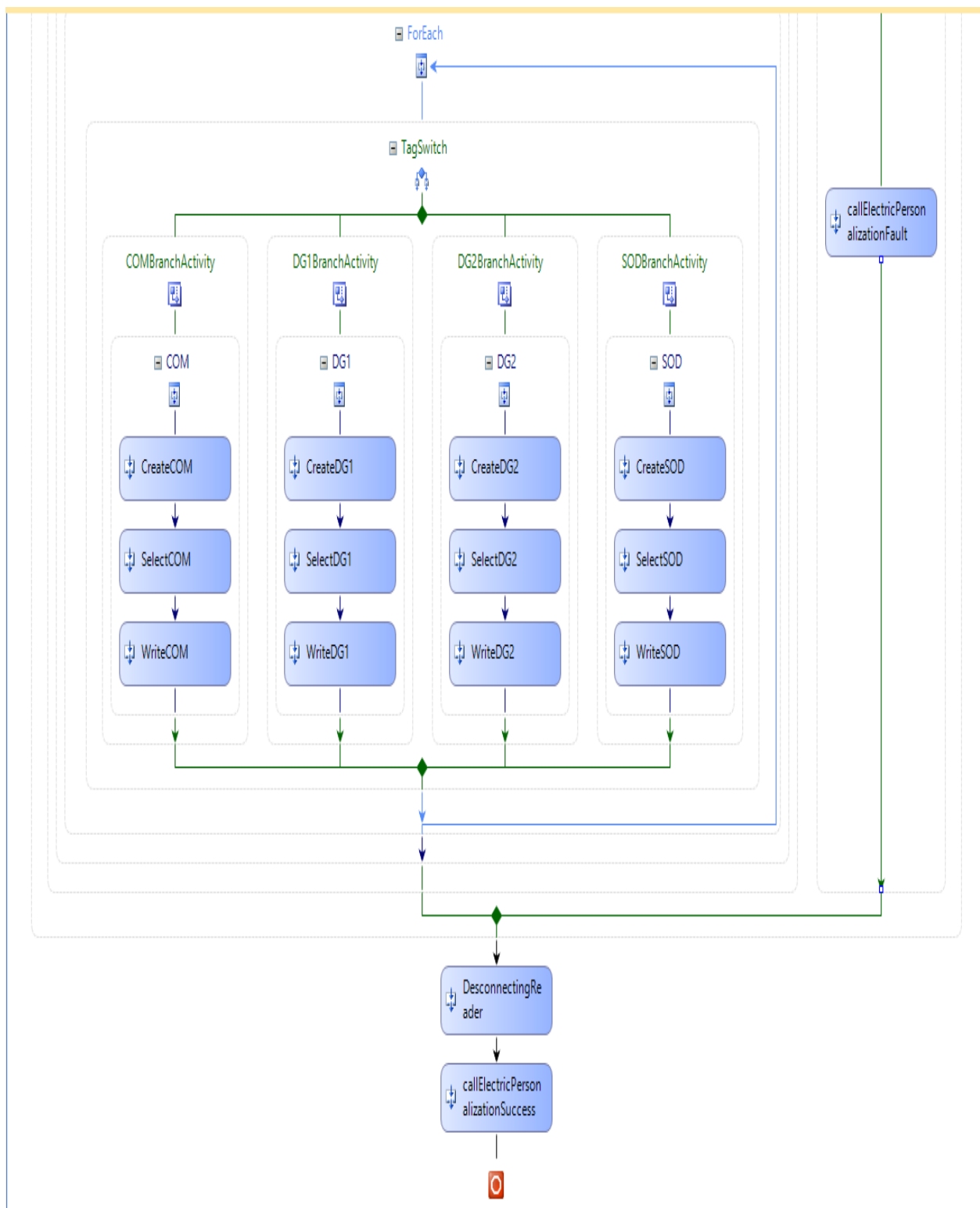


Figura 22. Vista General del Workflow parte2