

**Universidad de las Ciencias Informáticas**

**Facultad 1**



**Título:** “Solución informática para la personalización, control de calidad y entrega de documentos de identificación para la Universidad de las Ciencias Informáticas”.

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

**Autor(es):**

Laritza de la Nuez Pérez

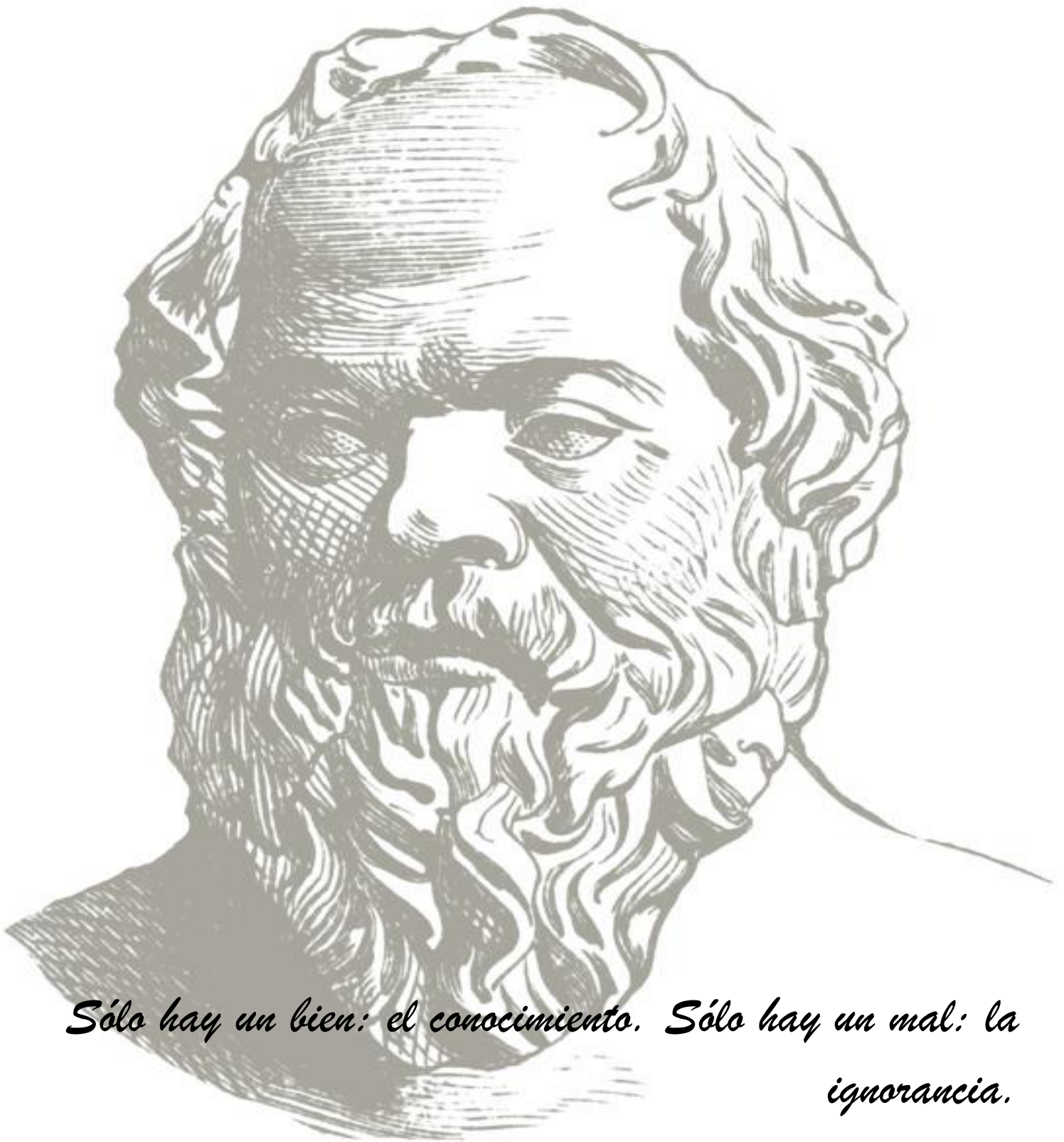
Rafael Rosales Sánchez

**Tutores:**

Ing. Ireisy Bermúdez Hoyo

Ing. Odeimy Morales Duarte

La Habana, Junio del 2013



*Sólo hay un bien: el conocimiento. Sólo hay un mal: la  
ignorancia.*

*Sócrates*

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado: Solución informática para la personalización, control de calidad y entrega de documentos de identificación para la Universidad de las Ciencias Informáticas. Y autorizamos a la Universidad de las Ciencias informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

\_\_\_\_\_  
Autor: Laritza de la Nuez Pérez

\_\_\_\_\_  
Autor: Rafael Rosales Sánchez

\_\_\_\_\_  
Tutor: Ing. Ireisy Bermúdez Hoyo

\_\_\_\_\_  
Tutor: Ing. Odeimy Morales Duarte

\_\_\_\_\_  
Tutor: Ing. Ireisy Bermúdez Hoyo

\_\_\_\_\_  
Tutor: Ing. Odeimy Morales Duarte

## DATOS DE CONTACTO

**Tutor:** Ing. Ireisy Bermúdez Hoyo.

- Ingeniero en Ciencias Informáticas, UCI 2011.
- Profesor del Departamento del Centro de Identificación y Seguridad Digital.
- Correo electrónico: [ibermudez@uci.cu](mailto:ibermudez@uci.cu)

**Tutor:** Ing. Odeimy Morales Duarte.

- Ingeniero en Ciencias Informáticas, UCI 2008.
- Categoría Docente: Profesor Instructor.
- Profesor del Departamento de Identificación del Centro de Identificación y Seguridad Digital.
- Correo electrónico: [omorales@uci.cu](mailto:omorales@uci.cu)

## *Dedicatoria y Agradecimientos.*

*A mi Señor Jesucristo por acompañarme en todo momento y permitir que hoy disfrutara de este día tan especial. Gracias por iluminar mis pasos.*

*A la memoria de mi padre, que sé, que donde quiera que esté se sentiría muy orgulloso de su hija.*

*A mi mamá Lucía y mi abuelita Olvido por su amor, por sus consejos y por su confianza depositada en mí en estos cinco años. Por inculcarme su afecto, su ternura y por su sacrificio. Gracias por ser los peldaños más fuertes para lograr esta meta.*

*A mi hermana Iris por quererme con la vida y brindarme su cariño. Mi confidente más fiel y leal. Has sido la hermana más tierna que Dios me ha dado.*

*A mi tía Gisela y a mis primas Yanet y Yaniset, por darme apoyo cuando más lo necesitaba.*

*A mis hermanos Elio y Osmany, porque aunque la distancia nos separa un poco, siempre están ahí para darme fuerzas y por ver en ellos la figura paterna que siempre he anhelado.*

*A Olguita por preocuparse por mí y acogerme como si fuera una hija más y por alentarme a que continuara adelante.*

*A mis compañeras inseparables en estos años de estudio Yanelis, Mary, Dayana, Yosleidy, Osmayda, Leidys, Jessica, Greysi, Elizabeth, Dailén y Miry por quererme, comprenderme y escucharme en los momentos alegres y tristes. Especialmente le doy las gracias a mi grupo 1504 por sus gestos tan cálidos y maravillosos que no olvidaré.*

*A mis profesores del CISED y de Identidad Cuba: Liudnet, Dannier, Ronaldo, Reynier Lester, Annie, Yandy, por su dedicación y apoyo brindado en esta ardua labor. Pero especialmente quiero agradecer a mis tutoras Ireysi y Odeimy por haber sido pacientes conmigo y por poder contar con ellas en lo que fuera necesario para la realización de esta investigación. Contribuyendo a mi formación como profesional.*

*A mi compañero de tesis por su amor, sacrificio y compromiso ante esta tarea.*

*A la Revolución cubana por brindarme la oportunidad de estudiar en una institución tan digna y maravillosa como la Universidad de las Ciencias Informáticas, cuyos recuerdos quedarán impregnados en mi memoria de estudiante. Al excelente claustro de profesores que contribuyeron a este sueño.*

*Laritzza*

*A toda mi familia, por estar siempre unida y apoyarme tanto, en especial a mis dos padres hermosos, mi abuela, mi esposa linda, mi prima Mayté, mi primo Humberto, mis primitos Alex y Amanda. Gracias por su apoyo y por confiar en mí eternamente.*

*Rafael*

## RESUMEN

En la actualidad el acelerado desarrollo de las ciencias y las tecnologías han tenido un gran impacto en los sistemas o soluciones informáticas, y con ellos los sistemas de identificación. Estos sistemas se han convertido en productos eficientes, flexibles y seguros, garantizando la autenticidad y veracidad de los documentos de identificación.

El presente trabajo de diploma forma parte del proyecto Plataforma Modular de Identificación y de Control de Acceso (PMICA). Dentro de sus objetivos, propone realizar una solución informática para la personalización, control de la calidad y entrega de un documento de identificación para la Universidad de las Ciencias Informáticas (UCI).

Esta solución está concebida para dar respuesta a los problemas existentes en la oficina de acreditación de la institución, referente a los procesos de personalización, control de calidad y entrega de los documentos de identificación. Dentro de sus principales ventajas se encuentran, la inclusión durante el proceso de personalización de medidas de seguridad a los documentos y el desarrollo de los componentes calidad y entrega con el fin de garantizar su veracidad y control.

**Palabras clave:** documentos de identificación, proceso, personalización, control de calidad y entrega.



## Índice de contenidos

Introducción .....	1
Capítulo I. Fundamentación Teórica .....	4
1.1 Introducción. ....	4
1.2 Documento de identificación. ....	4
1.3 Proceso de emisión de los documentos de identificación. ....	4
1.3.1. Conceptos asociados a los procesos de personalización, control de calidad y entrega de un documento de identificación. ....	5
1.4 Estándar internacional ISO/IEC 7810 para la confección de los documentos de identificación. ....	6
1.5 Estándares internacionales de la Organización de la Aviación Civil Internacional (OACI) en cuanto a seguridad de los documentos de viaje. ....	8
1.5.1. Elementos de seguridad incorporados durante el proceso de personalización para los documentos de viaje de lectura mecánica pertenecientes al formato ID-1 e ID-2. ....	8
1.6 Análisis de las empresas y soluciones que se dedican a la emisión de documentos de identificación. ....	9
1.7 Ambiente de desarrollo. ....	13
1.7.1. Metodología de desarrollo. ....	13
1.7.2. Herramientas CASE. ....	14
1.7.3. Lenguajes de desarrollo. ....	15
1.7.4. Lenguajes notacionales para modelados de procesos. ....	16
1.7.5. Marco de trabajo de programación. ....	17
1.7.6. Entorno integrado de desarrollo. ....	17
1.7.7. Sistema Gestor de Base de Datos. ....	18
1.7.8. <i>ORM NHibernate</i> . ....	19
1.7.9. Servidores <i>web</i> . ....	19
1.7.10. Cliente <i>TortoiseSVN-1.7.6</i> . ....	20
1.8 Análisis de las medidas de seguridad a incorporar en el documento de identificación y selección del estándar para la definición de las características físicas durante el proceso de confección. ....	21
1.8.1. Medidas de seguridad. ....	21
1.8.2. Selección del estándar para la confección física del documento. ....	21
1.9 Selección de la metodología y el marco de trabajo. ....	21
1.10 Conclusiones parciales. ....	23
Capítulo II. Análisis y diseño de la solución. ....	25
2.1. Introducción. ....	25
2.2. Modelo de negocio. ....	25
2.3. Análisis de posibles implementaciones o componentes existentes que puedan ser utilizados. ....	29
2.4. Roles del sistema. ....	31

2.5. Especificación de los requisitos funcionales. ....	31
2.6. Especificación de los requisitos no funcionales. ....	37
2.7. Arquitectura. ....	38
2.8. Propuesta de solución. ....	39
2.9. Patrones de diseño. ....	40
2.10. Especificación de los diagramas de clases del diseño. ....	41
2.11. Diagramas de secuencia. ....	42
2.12. Servicios del sistema. ....	43
2.13. Conclusiones parciales. ....	43
Capítulo III. Implementación y pruebas. ....	45
3.1. Introducción. ....	45
3.2. Estándares de codificación. ....	45
3.3. Tratamiento de errores. ....	46
3.4. Seguridad en el sistema. ....	47
3.5. Modelo de datos. ....	47
3.6. Diagrama de despliegue. ....	48
3.7. Diagrama de componentes. ....	49
3.7.1. Descripción del diagrama de componentes. ....	50
3.8. Interfaces del sistema. ....	51
3.9. Pruebas. ....	51
3.9.1. Pruebas funcionales en el diseño de casos de prueba utilizando la técnica de caja negra. ....	52
3.9.2. Resultados de las pruebas. ....	54
3.10. Conclusiones parciales. ....	55
Conclusiones generales. ....	56
Recomendaciones. ....	57
Bibliografía. ....	58
Bibliografía consultada. ....	61
Glosario de términos. ....	63
Anexos. ....	64

## Índice de figuras

Figura 1. Diagrama del proceso personalización.....	27
Figura 2. Diagrama del proceso de entrega. ....	28
Figura 3. Componentes del Sistema de Personalización de Documentos de Identificación. ....	29
Figura 4. Arquitectura del sistema.....	38
Figura 5. Propuesta de solución.....	40
Figura 6. Diagrama de secuencia del componente de personalización: Crear órdenes de solicitud de impresión. ....	43
Figura 7. Estándar de codificación: Convención Pascal para nombre de las clases. ....	45
Figura 8. Estándar de codificación: Convención Camel para el nombre de los métodos. ....	46
Figura 9. Modelo de datos.....	48
Figura 10. Diagrama de despliegue. ....	49
Figura 11. Diagrama de componentes. ....	50
Figura 12. Interfaz crear órdenes.....	51

## Índice de tablas

Tabla 1 Proceso de personalización.....	26
Tabla 2. Proceso de entrega.....	28
Tabla 3. Rol del componente de personalización.....	31
Tabla 4. Rol del componente de entrega.....	31
Tabla 5. Crear órdenes de solicitud de impresión.....	34
Tabla 6. Registrar entrega del documento.....	37
Tabla 7. Descripción del servicio LotServiceQuery.....	43
Tabla 8. Descripción de las variables del caso de prueba crear orden de solicitud de impresión.....	52
Tabla 9. Diseño del caso de prueba "Crear orden de solicitud de impresión". Escenario "Filtrar".....	53
Tabla 10. Diseño del Caso de Prueba "Crear orden de solicitud de impresión". Escenario "Crear orden".....	54
Tabla 11. Resultados de las pruebas del componente de personalización.....	54

## Introducción

La Universidad de las Ciencias Informáticas (UCI) es una institución con características especiales al resto de las universidades del país, ya que el manejo de muchos activos, información confidencial y afluencia de personal la convierten en un centro vulnerable en cuanto a seguridad. Por tales motivos cada persona perteneciente al centro debe contar con un documento de identificación llamado credencial o solapín.

Este documento de identificación debe transitar por diferentes procesos para ser emitido; los mismos son: el proceso de solicitud del documento de identificación, el proceso de captura de datos e imágenes, el proceso de supervisión de datos e imágenes y los procesos de personalización, control de calidad y entrega del mismo.

La UCI para emitir este documento cuenta con una oficina de acreditación de documentos de identificación cuyo objetivo es dar respuesta a las necesidades de gestionar los trámites de personalización de los documentos de identificación y realizar la entrega al personal correspondiente.

En la actualidad la aplicación informática encargada del proceso de emisión de las credenciales presenta deficiencias, ya que durante el proceso de personalización del documento solo le incluye una medida de seguridad digital, el código de barras, siendo fácil que cualquier persona falsifique este documento y acceda a la institución. Además esta aplicación informática no concibe un componente para el control de la calidad, que permita registrar las causas de la anulación del documento y la veracidad de los datos que han sido personalizados en él. También no realiza un registro de la entrega de forma digital, por lo que no se comprueba que el documento es entregado al personal correspondiente.

Debido a la problemática existente en la Universidad de las Ciencias Informáticas, en relación con la emisión de credenciales de identificación, en los procesos de personalización, control de la calidad y la entrega, se plantea el siguiente problema de la investigación: ¿Cómo lograr la personalización, control de calidad y entrega de un documento de identificación con elementos de seguridad en la Universidad de las Ciencias Informáticas? Partiendo del problema planteado se tiene como **objeto de estudio** los procesos de personalización, control de calidad y entrega de documentos de identificación.

Como **objetivo general** de la investigación: Desarrollar una solución informática que permita la personalización, control de calidad y entrega de documentos de identificación con elementos de seguridad incluidos para la Universidad de las Ciencias Informáticas.

Para lograr el cumplimiento del trabajo se proponen las siguientes **tareas de la investigación**:

- Realización de un estudio a nivel nacional e internacional de los sistemas dedicados a la emisión de los documentos de identificación, para identificar los procesos más frecuente por los que transcurre un documento de identificación, distinguir las tendencias tecnológicas que se utilizan para emitirlos y analizar posibles soluciones a reutilizar .
- Caracterización de los estándares internacionales que rigen la emisión de un documento de identificación, para definir la estructura física y los elementos de seguridad a incorporar en el nuevo documento.
- Elaboración de un estudio de las herramientas, tecnologías y metodologías utilizadas en el desarrollo de un producto de *software*, que permita caracterizar y distinguir las más factibles para el desarrollo de la solución.
- Descripción de los principales conceptos asociados al proceso de personalización, control de calidad y entrega de documentos de identificación que permita la comprensión del funcionamiento de los mismos.
- Modelación de los procesos personalización, control de calidad y entrega para la identificación de los requisitos funcionales y no funcionales.
- Descripción de los requisitos funcionales para comprender el flujo por los cuáles transcurre la solución a desarrollar.
- Implementación de las funcionalidades correspondientes a los componentes de personalización, control de calidad y entrega de documentos, para darle cumplimiento a la solución propuesta.
- Realización de pruebas utilizando la técnica de caja negra a los componentes de personalización, control de calidad y entrega de documentos, para validar el correcto funcionamiento del *software*.

Los **métodos científicos** a utilizar en la investigación son:

## **Métodos teóricos:**

- **Analítico – Sintético:** Con el objetivo de analizar la información y la documentación relevante para el desarrollo del *software*, enfatizando en los elementos más importantes que se relacionan con el objeto de estudio.
- **Modelación:** Utilizado para representar el modelado de procesos del negocio a informatizar. Además se modelan los diagramas de clases que definen la estructura del sistema, mostrando sus clases, métodos, atributos y las relaciones entre ellos, facilitando su comprensión por parte del desarrollador a la hora de su implementación.

## **Métodos Empíricos**

- **Entrevista:** Empleado para identificar las actividades comunes que realizan los procesos de personalización, control de calidad y entrega de un documento de identificación, a través de las entrevistas realizadas a personas con amplios conocimientos en el tema.

## **Justificación de la investigación.**

La solución propuesta tiene como implicación práctica:

- Especificación de requisitos funcionales de la propuesta de solución.
- Solución informática para la personalización, control de calidad y entrega de documentos de identificación en la Universidad de las Ciencias Informáticas.
- Documento de identificación que cumpla con los estándares propuestos por la Organización de la Aviación Civil Internacional (OACI) y la ISO/IEC 7810 en cuanto a medidas de seguridad y estructura física del documento.

## **Estructura del documento.**

La presente investigación estará compuesta por **3 capítulos**.

**Capítulo 1: Fundamentación Teórica** se plantean los principales conceptos y aspectos relacionados con el dominio del problema a resolver. Se exponen las principales tendencias tecnológicas actuales para el diseño de documentos de identificación, así como las soluciones existentes. Se mencionan los estándares internacionales que rigen el diseño de los documentos de identificación y las medidas de seguridad que deben poseer. Además se definen las herramientas, tecnologías y metodología a utilizar en el desarrollo de la solución informática.

**Capítulo 2: Análisis y diseño** se analizan el flujo actual de los procesos, se presentan los conceptos asociados a los componentes a desarrollar. Se describen los procesos relacionados con los componentes a desarrollar. Se analiza las posibles implementaciones o componentes que serán incorporados. Se identifican y describen los requisitos funcionales y no funcionales de la solución, así como la arquitectura que guiará su implementación. Se plantea la propuesta de solución, los patrones de diseño a utilizar y el modelo de diseño que incluirá las especificaciones de clases y los diagramas de secuencia.

**Capítulo 3: Implementación y pruebas** están conformadas por los estándares de codificación que se definieron para la organización en la implementación. Se describe cómo se lleva a cabo el tratamiento de errores y la seguridad en el sistema. Se modelan las principales entidades que persistirán en la base de datos, la estructura en componentes que poseerá la solución y se especifica cómo quedará desplegada la misma. Se muestran las principales interfaces del sistema y las pruebas para validar su correcto funcionamiento.

## Capítulo I. Fundamentación Teórica

### 1.1 Introducción.

En el presente capítulo se efectúa un estudio sobre el proceso de emisión de documentos de identificación, se realiza una investigación a nivel nacional e internacional sobre sistemas que son empleados para emitir estos documentos. Se mencionan los procesos que intervienen en la emisión, específicamente en los procesos de personalización, control de calidad y entrega de un documento de identificación y se expone un análisis de los estándares internacionales que rigen la creación de estos. Además se investiga sobre los diferentes elementos de seguridad, tecnologías empleadas y herramientas para el posterior desarrollo de la solución.

### 1.2 Documento de identificación.

Los documentos de identificación, también llamados documentos nacionales de identidad (DNI) o cédula de identidad, son documentos emitidos por una autoridad administrativa competente para permitir la identificación personal de los ciudadanos residentes en el país, así como aquellos que puedan encontrarse en territorio nacional de manera temporal. Constituyen documentos únicos de identificación personal e intransferible extendido a todos los ciudadanos. Estos usualmente poseen la fotografía del portador, su firma, nombre, nacionalidad, fecha de nacimiento, así como otros elementos que contribuyen a la identificación (1).

### 1.3 Proceso de emisión de los documentos de identificación.

Para determinar los procesos fundamentales para la emisión de un documento de identificación, se realizó una investigación en diferentes fuentes bibliográficas, tales como:

- Documentos de viaje de lectura mecánica para pasaportes, parte 1 volumen 1 sexta edición, año 2006.
- Documentos de viaje oficiales de lectura mecánica parte 3 volumen 1 sexta edición, año 2006.
- Propuesta de procesos para el desarrollo de un Sistema de Emisión de Documentos de Identificación (SEDI), año 2009.
- Sistema automatizado para la emisión del pasaporte andino de Venezuela, año 2006.
- Sistema automatizado de gestión para el centro de personalización e impresión de documentos del SAIME, año 2007.

Se distinguen como procesos fundamentales que intervienen en la emisión de un documento de identificación, los que se exponen a continuación:

- Proceso de solicitud de documentos de identificación.
- Proceso de captura y supervisión de datos e imágenes.



- Proceso de personalización de documentos de identificación.
- Proceso de control de calidad de los documentos de identificación.
- Proceso de entrega de documentos de identificación.

De manera general para emitir un documento de identificación se debe comenzar con la solicitud del mismo por parte de un ciudadano, al cual se le capturan los datos e imágenes siendo supervisados. Posteriormente se confecciona durante el proceso de personalización, se le controla la calidad y se entrega.

### 1.3.1. Conceptos asociados a los procesos de personalización, control de calidad y entrega de un documento de identificación.

Según la OACI (Organización de Aviación Civil Internacional) en el documento 9303 parte 1 volumen 1 “documentos de viaje de lectura mecánica” define:

**Personalización:** Proceso mediante el cual se aplica al documento datos personales de una determinada persona.

En el informe “Documentos: Manual para Desarrolladores” por los autores Diana Ombelli y Fons Knopjes se emplea el término de calidad para los documentos de seguridad. En este informe el concepto de **control de calidad** para dichos documentos, está aplicado a diferentes dimensiones como:

- conformidad: se entiende como el grado en que el documento de seguridad se ajusta a las reglas establecidas por las normas internacionales.
- rasgos: se entienden por los elementos de seguridad que protegen el documento contra el fraude.
- fiabilidad: se entiende por el grado en que la población lo acepta para el uso, para el cual ha sido concebido.
- durabilidad: se entiende por la capacidad del documento para resistir las condiciones normales de uso durante un período de tiempo acordado.

En el informe “Documentos: Manual para Desarrolladores” por los autores Diana Ombelli y Fons Knopjes se aplica el término de **entrega** a los documentos como expedición y se describe de la siguiente forma:

Una vez efectuados todos los controles de calidad, el documento puede ser expedido. La forma en que se expida el documento depende, en parte, del método de solicitud, la complejidad del procedimiento de personalización y la ubicación del organismo expedidor. Si hay un representante de dicho organismo en cada municipio, el usuario podrá recoger su documento personalmente. Ahora bien, si la entidad expedidora únicamente tiene oficinas regionales, entonces será necesario utilizar un método

diferente de expedición. Lo mismo ocurre cuando solamente hay una única oficina nacional. La expedición puede realizarse del modo siguiente:

- personalmente mientras se espera;
- personalmente con un plazo de espera de varios días;
- por correo;
- por correo con restricciones de uso.

Con el estudio realizado a los diferentes materiales consultados, los autores del presente trabajo han definido los conceptos de personalización, control de calidad y entrega como:

**Proceso de personalización:** Un proceso de personalización consiste en imprimir los datos obtenidos del ciudadano sobre el documento de identificación, así como la inclusión de medidas de seguridad en caso de ser definidas.

**Proceso de control de calidad:** Un proceso de control de calidad consiste en verificar la veracidad y calidad visual de los datos impresos en el documento. Este proceso se puede realizar de forma visual, o por medio de dispositivos.

**Proceso de entrega:** Un proceso de entrega consiste en entregarle el documento de identificación al ciudadano correspondiente.

## 1.4 Estándar internacional ISO/IEC 7810 para la confección de los documentos de identificación.

La Organización Internacional para la Normalización (ISO<sup>1</sup>) define normas internacionales para la confección de documentos y tarjetas de identificación. Con el estándar ISO/IEC 7810 se especifica formatos o tallas para tarjetas de identificación, entre los que se encuentran: ID-1, ID-2, ID-3 e ID-000 (2).

**ID-1:** Constituye el formato más conocido de las tarjetas. Presenta dimensiones de 85,6mm x 54 mm. Ha sido el más usado durante bastante tiempo y se encuentra recomendado por el estándar internacional ISO 7810 (estándar para características físicas de tarjetas con banda magnética). Una característica importante es el grosor, el mismo ha de ser igual 0.76mm con una tolerancia de 0.08mm. Este tipo de formato se emplea en las tarjetas de banco (tarjetas ATM<sup>2</sup>, tarjetas de crédito, de

---

<sup>1</sup> Organismo Internacional de Normalización: Organismo encargado de promover el desarrollo de normas internacionales de fabricación (tanto de productos como de servicios), 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 (públicas o privadas) a nivel internacional.

<sup>2</sup> Un cajero automático o redbanc es una máquina expendedora usada para extraer dinero utilizando una tarjeta de plástico con una banda magnética o chip (tarjeta de débito o tarjeta de crédito por ejemplo), sin necesidad de personal del banco.

débito). También es usado como tarjeta de identificación personal en algunos países como: Chile, Pakistán. Proporciona al usuario bastante comodidad en su manejo, de tal manera que no sea demasiado larga para llevarla en la cartera, ni demasiado pequeña que pueda perderse con facilidad (3), (4).

El formato ID-1 define otros estándares con diversas características, entre ellos:

ISO 7811: Define las técnicas tradicionales de registro de datos de ID-1 de las tarjetas de identificación, es decir, caracteres en relieve y diferentes formatos de grabación magnética (4).

ISO 7813 define características adicionales de ID-1 tarjetas bancarias de plástico, por ejemplo el espesor debe tener 0,76 mm y las esquinas redondeadas deben tener un radio de 3,18 mm (4).

ISO 7816: Define tarjetas del tipo ID-1 con un chip incorporado y superficies de contacto para la alimentación y señales de funcionamiento (4).

ISO 14443 define tarjetas de identificación con un chip integrado (tarjeta de proximidad) y una antena de bucle magnético que funciona a 13,56 MHz (RFID). Las normas más recientes de la Organización de la Aviación Civil Internacional (OACI) para la lectura de los documentos de viaje de lectura mecánica, en la norma ISO 14443 de chips RFID<sup>3</sup> especifican un formato de archivo criptográficamente firmado y un protocolo de autenticación para el almacenamiento de datos biométricos como las (fotografías del rostro, huellas dactilares y / o iris) (4) .

**ID-2:** El formato ID-2 especifica un tamaño de 105 x 74mm. Este tamaño es el formato A7. Se utiliza como documento de identidad en Alemania. Este formato ligeramente más grande proporciona suficiente espacio para una foto facial claramente reconocible, pero es todavía lo suficientemente pequeña como para ser llevada en el bolsillo (4).

**ID-3:** Especifica un tamaño de 125 x 88 mm. Este tamaño es el formato B7. Este formato se utiliza para los pasaportes y las visas (4).

**ID-000:** Especifica un tamaño de 25 mm x 15 mm. Este formato se utiliza para tarjetas SIM<sup>4</sup> (4).

---

<sup>3</sup> RFID (siglas de Radio Frequency IDentification, en español identificación por radiofrecuencia) es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas.

<sup>4</sup> Una tarjeta SIM es una tarjeta inteligente desmontable usada en teléfonos móviles y módems HSDPA o HSUPA que se conectan al puerto USB. Las tarjetas SIM almacenan de forma segura la clave de servicio del suscriptor usada para identificarse ante la red, de forma que sea posible cambiar la línea de un terminal a otro simplemente cambiando la tarjeta.

## 1.5 Estándares internacionales de la Organización de la Aviación Civil Internacional (OACI) en cuanto a seguridad de los documentos de viaje.

La OACI define como documento de viaje seguro, aquellos documentos que incluyan los elementos de seguridad definidos por la organización y que cumplan con las características físicas propuestas por el estándar ISO/IEC 7810 (5).

### 1.5.1. Elementos de seguridad incorporados durante el proceso de personalización para los documentos de viaje de lectura mecánica pertenecientes al formato ID-1 e ID-2.

Elementos básicos:

- Patrón de diseño con fondo de seguridad *guilloche* a dos tintas.
- Impresión arco iris.
- Patrón anti escáner.
- Texto en microimpresión.

Otros elementos constituyen:

- Diseño doble de seguridad.
- Elemento de diseño en relieve (3-D).
- Error intencional incorporado dentro de un texto en microimpresión.
- Numeración: En la cara de datos personales del dvLM (documento de viaje de Lectura Mecánica) aparecerá un único número de documento. El número se incorpora de preferencia en el momento de fabricación de la tarjeta, aunque se puede incorporar en una etapa posterior utilizando la misma técnica con la que se aplican los datos personales (5).
- Medidas especiales de seguridad para uso con tarjetas plásticas: Cuando un documento de viaje está construido enteramente de plástico, se emplearán elementos de seguridad ópticamente variables que proporcionen una apariencia cambiante según el ángulo de visión. Estos elementos pueden tener forma de imágenes latentes, elementos lenticulares, tintas de color cambiante o elementos de imágenes difractivas ópticamente variables (5).
- Imagen secundaria del retrato del titular (impreso o perforado) (5).
- Elementos de lectura mecánica: Un elemento de lectura mecánica es todo aquel que almacena datos y que pueda ser leído por los sistemas y tecnologías asociadas.

## 1.6 Análisis de las empresas y soluciones que se dedican a la emisión de documentos de identificación.

En la actualidad existen diversas soluciones informáticas que se dedican a la impresión y personalización de documentos de identificación con un alto nivel de seguridad, a continuación se citan algunas de ellas.

### A nivel internacional

**Bundes Druckerei:** Empresa alemana dedicada a la impresión de documentos de identificación, su sistema está basado en impresión centralizada o descentralizada, utilizan impresión láser para el grabado de los datos personales en el documento, en el caso de un documento de identificación electrónico incorporan los datos biométricos y biográficos en un chip. Su plataforma de personalización comprende componentes de *hardware* y *software* que pueden ser fácilmente adaptables a las necesidades específicas del documento de identificación. Cada máquina de personalización tiene su propio *software* de control de la máquina, esto permite que los documentos de identificación se trasladen a través del sistema de personalización y, gracias a la función de control de calidad integrado, hace que sea posible aplicar los datos correctos y completos en la estación de personalización respectiva. La plataforma puede personalizar diferentes tipos de documentos, especialmente las tarjetas de identidad electrónica, pasaportes electrónicos, permisos de residencia y de conducción electrónicos, tarjetas de licencia, entre otros. La plataforma de personalización cumple con las normas internacionales como la OACI 9303, ISO 7816, ISO 14443, etc. *Bundes* ha desarrollado más de 135 millones de carnés de identidad y más de 64 millones de pasaportes en la Unión Europea, hoy suministra documentos de identificación y tarjetas de alta seguridad (6).

Análisis de la solución: Posee el 50% del mercado de la impresión del euro para Alemania, todos los timbres fiscales y los pasaportes del país. Tienen desarrollado el pasaporte electrónico, pero la adquisición de sus soluciones para el país es muy elevada.

**François-Charles Oberthur Fiduciaire:** Empresa francesa dedicada a la impresión de documentos de valor y de alta seguridad, prestando servicios y soluciones seguras. Su dirección principal está en París, Francia con oficinas comerciales en el Reino Unido, emplea como normas de calidad: ISO 9001 versión 2000, AFAQ (*'l'Association Française pour l'Amélioration et le management de la Qualité*, en español, Asociación Francesa para la mejora y la gerencia de la Calidad). Ha impreso más de 3.5 mil millones de billetes de banco, 1.5 mil millones de documentos de seguridad, contando con 4 plantas de producción y más de 1000 empleados altamente calificados. *François-Charles Oberthur Fiduciaire* logró conquistar la posición de líder en este campo, gracias a su experiencia en el campo de los pasaportes biométricos y de los pasaportes electrónicos (7).

Análisis de la solución: Tiene como principal desventaja el alto costo del sistema de personalización e impresión.

**Mühlbauer:** Empresa alemana consultora e implementadora de la industria de tarjetas inteligentes (*Smart*), pasaportes electrónicos y RFID. Es el líder en cuanto a sistemas innovadores y soluciones de *software* para la producción y personalización; además, apoya a sus clientes en la planificación de proyectos, transferencia tecnológica incluyendo la integración del sistema y el apoyo en producción. Se han lanzado más de 100 productos de diferentes estándares y personalizados. Ha creado inteligentes soluciones de *software* para el registro de datos y la gestión de producción, las cuales han sido añadidas a la gama de tarjetas y *TECURITY*<sup>5</sup>. *Tecurity* provee una solución para la emisión de documentos de identidad llamada *TIDIS*. *Tidis* es un paquete de soluciones completas que permite a los gobiernos, las imprentas de seguridad y los integradores de sistemas adaptar sus infraestructuras de identificación (8) .

Análisis de la solución: Tiene como principal desventaja el alto costo de adquisición de sus productos los cuales solo están encaminados al desarrollo de tarjetas inteligentes y pasaportes electrónicos.

**Thomas Greg & Sons:** Empresa colombiana dedicada más de cuatro décadas a ofrecer productos y servicios de seguridad para la banca, la industria en general, el gobierno y el comercio. Se caracteriza por la excelencia en la calidad, lo que les ha permitido ser líderes en los mercados que atienden. Este liderazgo en la impresión de valores y documentos de seguridad, los ha llevado a ofrecer nuevos y novedosos productos de la más alta calidad, acompañados de modernas técnicas contra la falsificación de los documentos utilizados en el sistema financiero y comercial. *Thomas Greg & Sons* tiene la experiencia internacional, la avanzada tecnología y el capital humano necesario para que sus documentos sean elaborados con total exclusividad y los más altos patrones de calidad. Sus plantas cuentan con la certificación de calidad internacional ISO 9001 (9) .

Análisis de la solución: Su mayor desventaja es que no utilizan dispositivos ópticos de verificación tipo holograma o kinegrama, los cuales son los más fuertes desde el punto de vista de seguridad y son verificables sin medios técnicos.

**Servicio Administrativo de Identificación, Migración y Extranjería (SAIME):** Solución desarrollada en conjunto con la Universidad de las Ciencias Informáticas y la hermana República Bolivariana de Venezuela, nación para la cual ha sido creada. Interactúa con dispositivos de captura biométrica, evitando el fraude ya que posibilita la unicidad de los ciudadanos con la comparación de sus huellas dactilares. La confección de la cédula de identificación se realiza de forma digital con equipos de

---

<sup>5</sup> La solución TIDIS de Mühlbauer para la emisión de documentos de identidad es una solución integral que une todos los elementos necesarios para un proyecto de identidad completo.

impresión que incluyen el chip electrónico. Fue diseñada con el objetivo de facilitar la identidad de las personas residentes en Venezuela, ya que los documentos de identificación eran fácilmente falsificables, la entrega de los pasaportes tardaban años y con un elevado costo. Además de que no se contaba con un control sobre los ciudadanos que entraban y salían del país. Está estructurado por cuatro componentes esenciales con características y actividades específicas. Estos son: Cedulación, Pasaporte, Migración y Extranjería. Para el proceso de personalización de los pasaportes y documentos de extranjería, el impresor de pasaporte temporal realiza una verificación de la existencia del documento y el pasaporte a imprimir, en caso de que exista se imprime y se verifica la calidad de cada documento. Cuando el pasaporte y los documentos de extranjería están en la oficina, se procede posteriormente a imprimir un sobre para cada pasaporte y documento de extranjería. Más tarde se verifica la calidad de los sobres, para luego ensobrar los documentos y enviarlos al almacén para su posterior entrega al cliente (10).

Análisis de la solución: Su propósito solamente se encuentra enmarcado en la emisión de pasaportes electrónicos.

## A nivel nacional

**Sistema EMIPAS:** Es desarrollado por DATYS<sup>6</sup>. Es un *software* para la emisión de pasaportes de lectura mecánica que cumple con estándares de seguridad y calidad internacionales, basados en la OACI. Posee un ambiente amigable y flexible, con una interfaz sencilla e intuitiva. Tiene un componente de personalización. Este componente permite a los funcionarios de las oficinas emisoras que se procese la solicitud de los pasaportes. La personalización del documento se elabora a través de una sucesión de acciones. Estas acciones comprenden desde la captación de datos e imágenes, identificación, verificación, impresión del pasaporte, el control de la calidad para la emisión, hasta la entrega del pasaporte. Durante la personalización de pasaportes asigna medidas de seguridad basadas en micro textos, grabado láser, *guilliches* con imagen latente, imagen fantasma, código de barras bidimensional para datos biométricos y alfanuméricos. Posee un componente de control de calidad y supervisión para garantizar la fiabilidad y exactitud de la información contenida. Además cuenta con el componente de entrega que se encarga de la verificación dactilar en el momento de la entrega del documento. Dentro de sus clientes en el mercado exterior se encuentran México, Francia, Venezuela, entre otros (11).

Análisis de la solución: El sistema EMIPAS<sup>7</sup> solo está enfocado en la personalización, control de calidad y entrega de pasaportes por lo que no cumple con los requisitos del sistema a desarrollar.

---

<sup>6</sup> Tecnologías y Sistemas

<sup>7</sup> Sistema Emisor de Pasaportes

**Sistema de Personalización de Documentos de Identificación de la República de Cuba:** Solución informática desarrollada en la Universidad de las Ciencias Informáticas (UCI) que forma parte del proyecto Identidad Cuba. Es desarrollado usando tecnologías de punta para la personalización de documentos tanto en papel como otros materiales como PVC (cloruro de polivinilo), poliuretano, policarbonato, con la inclusión de elementos de seguridad durante el proceso de personalización. Este sistema es capaz de adaptarse a la personalización de diferentes documentos de identificación, incorporando medidas de seguridad tales como: imagen fantasma, imágenes láseres cambiantes, código de barras bidimensional, entre otras, así como el control de los insumos y medios que se requieren. Se garantiza que los documentos que se imprimen cumplen con las normas internacionales establecidas por la Organización de la Aviación Civil Internacional, para este tipo de documentos. Cuenta con un control exhaustivo de las órdenes de producción e incluye un componente de control de calidad. Con el componente de calidad permite chequear en diferentes momentos, el tránsito de las solicitudes, además de que cuenta con posibilidades de lectura mecánica de códigos OCR-B<sup>8</sup>, 1D, 2D y chip. Además se comprueban la veracidad de los datos impresos lo que permite emitir un documento seguro para los ciudadanos y evita que se cometan usurpaciones a la identidad personal (12).

Análisis de la solución: Esta solución está creada bajo una filosofía genérica siendo capaz de adaptarse a la personalización de diferentes tipos de documentos. También garantiza durante el proceso de personalización la inclusión de medidas de seguridad tales como: imagen fantasma, imágenes láseres cambiantes, código de barras bidimensional, entre otras. Además contiene un componente para control de calidad, garantizando la veracidad de los datos impresos en el documento. De esta manera este sistema se integrará a la solución a desarrollar aprovechando la compatibilidad de las tecnologías que utiliza y las ideas o propósitos con que ha sido creado.

**Sistema de Identificación UCI:** Sistema de acreditación que se encuentra desarrollado en la plataforma *Netbeans*. Esta aplicación contiene todos los datos del personal que estudia y labora en dicha institución. Es la encargada de realizar los trámites de impresión y entrega de las credenciales. Posee opciones como: consultar estado de la credencial, gestión de los estados de la credencial, buscar persona, impresión de credenciales, gestión de fotos, etc. A pesar de garantizar la impresión de los documentos hasta el momento en la universidad, este sistema cuenta con algunas deficiencias dentro de las que se encuentran: que solo se enmarca en el proceso de identificación de las personas en la universidad, además durante el proceso de personalización no realiza la inclusión de medidas de seguridad física tales como: imagen fantasma, imágenes láseres cambiantes, código de barras bidimensional, etc, la única medida de seguridad digital que presenta es el código de barras

---

<sup>8</sup> Tipo de letra específico diseñado para reconocimiento óptico de caracteres.



unidimensional. Además esta aplicación no cuenta con un componente de calidad para realizar un chequeo sobre ese documento personalizado y se registra la entrega de forma manual.

Análisis de la solución: El documento de identificación durante la personalización no les son añadidas medidas de seguridad, lo que contribuye a que sea fácilmente falsificable. La aplicación informática no chequea la calidad y no registra la entrega del documento de forma digital, por lo que hace a la universidad vulnerable por la entrada de personas ajenas al centro.

## 1.7 Ambiente de desarrollo.

### 1.7.1. Metodología de desarrollo.

Una metodología de desarrollo, es un conjunto de procedimientos, técnicas, herramientas y soporte documental que ayuda a los desarrolladores a realizar nuevo *software*. No existe una metodología de *software* universal. Se desarrollan con el objetivo de dar solución a los problemas existentes en la producción de *software*, que cada vez son más complejos. En un proyecto de desarrollo de *software* la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo (13).

**RUP (*Rational Unified Process*)** se caracteriza por ser una metodología pesada, adaptable a cualquier proyecto. Consta de cuatro fases. Se basa en casos de uso, donde se describen los resultados que se esperará del *software*. Se encuentra enfocada a la arquitectura del sistema, por lo que requiere de mucha documentación. Es muy organizativa, presenta un proceso muy general, pero se encuentra dirigida a proyectos muy grandes, por lo que es necesario adaptarlo a las características de la empresa. (14).

**FDD (*Feature Driven Development*)** es una metodología ágil para el desarrollo de sistemas, que cuenta con características que posibilitan el correcto desarrollo de los productos de *software*, dentro de las que se encuentran el no énfasis en la obtención de los requerimientos, sino en cómo se realizan las fases de diseño y construcción. Se preocupa por la calidad, por lo que incluye un monitoreo constante del proyecto. Ayuda a contrarrestar situaciones como el exceso en el presupuesto, fallas en el programa o el hecho de entregar menos de lo deseado. Propone tener etapas de cierre cada dos semanas, Se basa en un proceso iterativo con iteraciones cortas que producen un *software* funcional que el cliente y la dirección de la empresa pueden ver y monitorear (15).

La metodología FDD tiene cinco procesos. Los primeros tres se hacen al principio del proyecto y los dos restantes al final, los mismos son:

Desarrollar un modelo global: Al inicio del desarrollo se construye un modelo teniendo en cuenta la visión, el contexto y los requisitos que debe tener el sistema a construir. Este modelo se divide en áreas que se analizan detalladamente (15).

Construir una lista de los rasgos: Se elabora una lista que resuma las funcionalidades que debe tener el sistema, cuya lista es evaluada por el cliente. Cada funcionalidad de la lista se divide en funcionalidades más pequeñas, para un mejor entendimiento del sistema (15).

Planeación por funcionalidades: En este punto se procede a ordenar los conjuntos de funcionalidades conforme a su prioridad y dependencia, y se asigna a los programadores jefes (15).

Diseñar por rasgo: Se selecciona un conjunto de funcionalidades de la lista. Se procede a diseñar y construir la funcionalidad mediante un proceso iterativo, decidiendo que funcionalidad se va a realizar en cada iteración. Este proceso iterativo incluye inspección de diseño, codificación, pruebas unitarias, integración e inspección de código (15).

Construir por Rasgo: Se procede a la construcción total del proyecto (15) .

## 1.7.2. Herramientas CASE.

### ¿Qué son las herramientas CASE?

(*Computer Aided Software Engineering*, Ingeniería de *software* asistida por ordenador). Conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de *software* y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un *software*. Este ciclo de vida comprende: la investigación preliminar, el análisis, el diseño, la implementación e instalación. Constituye una innovación en la organización, un concepto avanzado en la evolución de la tecnología con un potencial profundo en la organización. Se puede ver al CASE como la unión de las herramientas automáticas de *software* y las metodologías de desarrollo de *software* formales (16).

**Rational Rose** es una herramienta de producción y *comercialización* establecidas por *Rational Software Corporation* (actualmente parte de IBM<sup>9</sup>). *Rose* es un instrumento operativo conjunto, que utiliza el Lenguaje Unificado (UML) como medio para facilitar la captura de dominio de la semántica, la arquitectura y el diseño. Este *software* tiene la capacidad de crear, ver, modificar y manipular. Algunas características que posee son:

- No es gratuito, se debe hacer un previo pago para poder adquirir el producto.
- Habilita asistentes para crear clases y provee plantillas de código que pueden aumentar significativamente la cantidad de código fuente generada. Adicionalmente, se pueden aplicar

---

<sup>9</sup> International Business Machines (IBM) es una empresa multinacional estadounidense de tecnología y consultoría con sede en Nueva York. IBM fabrica y comercializa hardware y software para computadoras y ofrece servicios de infraestructura, alojamiento de Internet y consultoría en una amplia gama de áreas relacionadas con la informática desde computadoras centrales hasta nanotecnología.

los patrones de diseño, *Rational Rose* ha provisto veinte de los patrones de diseño GOF<sup>10</sup> para *Java*.

- Admite la integración con otros entornos integrados de desarrollo (17).

**Visual Paradigm para UML** es una herramienta para el Lenguaje Unificado de Modelado (UML) profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El *software* de modelado ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Dentro de sus principales características se encuentran:

- Soporte de UML versión 2.1.
- Diagramas de procesos de negocio.
- Interoperabilidad con modelos UML 2 (meta modelos UML 2.x para plataforma Eclipse) a través de XMI<sup>11</sup> (nueva característica).
- Ingeniería inversa - código a modelo, código a diagrama.
- Distribución automática de diagramas - reorganización de las figuras y conectores de los diagramas UML (18).

### 1.7.3. Lenguajes de desarrollo.

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Constituye un modo práctico para que los usuarios puedan dar instrucciones a un equipo (19).

**C#** es un lenguaje de programación orientado a objetos. C# no permite la inclusión de funciones ni variables globales que no estén incluidos en una definición de tipos, por lo que la orientación a objetos es más pura y clara que en otros lenguajes como C++. Además, C# soporta todas las características del paradigma de la programación orientada a objetos, como son la encapsulación, la herencia y el polimorfismo. Incluye mecanismos de control de acceso a tipos de datos, lo que garantiza que no se produzcan errores difíciles de detectar como un acceso a memoria de ningún objeto. Incorpora

---

<sup>10</sup> Gang of Four es el nombre con el que se conoce comúnmente a los autores del libro *Design Patterns*, referencia en el campo del diseño orientado a objetos.

<sup>11</sup> XMI o XML Metadata Interchange (XML de Intercambio de Metadatos) es una especificación para el Intercambio de diagramas. La especificación para el intercambio de diagramas fue escrita para proveer de una forma de compartir modelos UML entre diferentes herramientas de modelado.

elementos que se ha demostrado a lo largo del tiempo que son muy útiles para el programador, como tipos decimales o booleanos, un tipo básico string ,así como una instrucción que permita recorrer colecciones con facilidad(instrucción foreach), elementos los cuales es necesario simular en lenguajes como: C++ o Java (20).

**C++** es un lenguaje imperativo orientado a objetos que se deriva del lenguaje C. Se convierte en el predecesor de C para agregarle cualidades y características que no poseía dicho lenguaje. No es un lenguaje orientado a objetos puro, como lo es el lenguaje Java y C#. Permite introducir nuevas palabras claves y operadores de clase, pero algunas de sus extensiones tienen aplicación fuera del contexto de programación con objetos (fuera del ámbito de las clases). Es un lenguaje de programación extremadamente largo y complejo, pero con mucha popularidad ya que desde sus comienzos ha tenido éxitos, por ejemplo muchos sistemas operativos y compiladores se han escrito en C++(*Windows* y *Java*) (21).

#### 1.7.4. Lenguajes notacionales para modelados de procesos.

La notación de modelado de procesos de negocios es un estándar gráfico para crear diagramas similares a los organigramas, que pueden ser entendidos con facilidad por todas las partes interesadas, como analistas, desarrolladores, programadores y gestores (22).

**SPEM (Software Process Engineering Metamodel)** es un *framework* que permite instanciar procesos. Su objetivo es satisfacer un gran rango de métodos y procesos de desarrollo de *software* o sistemas, sin agregar características específicas para dominios o disciplinas particulares. Se puede escribir de dos formas como un meta modelo y como un perfil de UML 2. Está organizado en siete paquetes (23).

**BPMN (Business Process Modeling Notation)** es una notación gráfica que describe la lógica de los pasos de un proceso de negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos, y los mensajes que fluyen entre los participantes de las diferentes actividades. Proporciona un lenguaje común para que las partes involucradas, puedan comunicar los procesos de forma clara, completa y eficiente. De esta forma BPMN define la notación y semántica de un diagrama de procesos de negocio (*Business Process Diagram, BPD*). BPD es un diagrama diseñado para ser usado por los analistas de procesos, quienes diseñan, controlan y gestionan los procesos. Dentro de un diagrama de procesos de negocios BPD, se utilizan un conjunto de elementos gráficos, que se encuentran agrupados en categorías (24).

## 1.7.5. Marco de trabajo de programación.

**ASP.NET** es un conjunto de tecnologías de desarrollo de aplicaciones *web*, comercializado por *Microsoft*. Es usado por programadores para construir sitios *web* domésticos, aplicaciones *web* y servicio XML<sup>12</sup>. Forma parte de la plataforma .NET de *Microsoft* y es la tecnología sucesora de la tecnología *Active Server Page* (ASP). Se construyó con el objetivo de resolver las limitaciones de ASP y posibilitar la creación de *software* como ASP.NET. Está construido sobre el *Common Language Runtime*, permitiéndoles a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el *Framework.NET*. Introduce el concepto del *code-behind*, por el que una misma página se compone de dos ficheros: el de la interfaz de usuario y el código, con ello se facilita la programación de aplicaciones en múltiples capas, lo que se traduce en la total separación entre lo que el usuario ve y lo que la base de datos tiene almacenado. Por lo tanto cualquier cambio drástico de especificaciones minimiza los cambios en la aplicación y maximiza la facilidad de mantenimiento (25).

## 1.7.6. Entorno integrado de desarrollo.

Un entorno de desarrollo integrado (IDE), también conocido como entorno de diseño integrado, entorno integrado de depuración o entorno de desarrollo interactivo) es una aplicación de *software* que proporciona servicios integrales a los programadores de computadoras para el desarrollo de *software* (26).

**Netbeans** es una plataforma con base modular y extensible usada como una estructura de integración, para crear aplicaciones de escritorio grande. Ofrece servicios comunes a las aplicaciones de escritorio, lo que permite que el desarrollador se enfoque en la lógica específica de la aplicación que vaya a desarrollar. Permite administrar las interfaces de usuario, las configuraciones del usuario y las ventanas. Es un *framework* basado en asistentes (diálogos paso a paso). Provee soporte para las aplicaciones orientadas a servicios (SOA<sup>13</sup>), incluyendo herramientas de esquemas XML, un editor WSDL<sup>14</sup> y un editor BPEL<sup>15</sup> para *web services* (27).

---

<sup>12</sup> XML, siglas en inglés de eXtensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes.

<sup>13</sup> SOAP, La 'Arquitectura Orientada a Servicios de cliente' (en inglés Service Oriented Architecture), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.

<sup>14</sup> WSDL (en ocasiones leído como wisdel) son las siglas de Web Services Description Language, un formato XML que se utiliza para describir servicios web.

<sup>15</sup> (Web Services) Business Process Execution Language, WS-BPEL (en castellano, Lenguaje de Ejecución de Procesos de Negocio con Servicios Web), es un lenguaje estandarizado para la composición de servicios web.

**Microsoft Visual Studio 2010 Ultimate** es el exhaustivo paquete de herramientas de administración del ciclo de vida de las aplicaciones para equipos. Con este paquete se puede garantizar la calidad de los resultados, desde el diseño hasta la implementación. *Microsoft Visual Studio 2010 Ultimate* incluye potentes herramientas que simplifican todo el proceso de desarrollo de aplicaciones, de principio a fin. Los equipos pueden observar una mayor productividad y ahorro de costes al utilizar características de colaboración avanzadas, así como herramientas de pruebas y depuración integradas que le ayudarán a crear siempre un código de gran calidad. Algunas características de modelado y visualización que extiende son:

- Generar gráficos de dependencias para código de C o C++ y proyectos *web* de ASP.NET.
- Crear diagramas de capas a partir del código de C o C++ y validar las dependencias del código con los diagramas de capas.
- Generar código para los diagramas de clases UML.
- Crear diagramas de clases UML a partir del código.
- Importar elementos de secuencia UML, clase y diagrama de casos de uso de los archivos XML 2.1, que puede ser exportados de otras herramientas.
- Proporcionar la extensibilidad que permite escribir código personalizado para modificar los diagramas de capas y validar el código con ellos (28), (29).

## 1.7.7. Sistema Gestor de Base de Datos.

Un sistema de Gestión de Base Datos (SGBD) es el conjunto de programas que permiten definir, manipular y utilizar la información que contienen las bases de datos. Así como realizar todas las tareas de administración necesaria para mantenerlas operativas, mantener su integridad, confidencialidad y seguridad. Una BD nunca se accede o manipula directamente, sino a través del SGBD. Se puede considerar el SGBD como la interfaz entre el usuario y la base de datos. (30)

**Oracle** es un Sistema Gestor de Base de Datos (SGBD) con características objeto-relacionales que pertenecen al modelo evolutivo de Sistema Gestor de Base de Datos (SGBD). Posee un entorno cliente –servidor. Tiene la capacidad de gestionar grandes bases de datos. Además puede soportar la alta concurrencia de usuarios. Mantiene un alto rendimiento en las transacciones. Posee una disponibilidad controlada de los datos de las aplicaciones. Posee portabilidad y compatibilidad (31).

**PostgreSQL** es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (*Berkeley Software Distribution*) y con su código fuente disponible libremente. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos, para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. *PostgreSQL*

funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Posee características tales como:

- Se caracteriza por ser un sistema estable, de alto rendimiento, gran flexibilidad ya que funciona en la mayoría de los sistemas *Unix*, además tiene características que permiten entender fácilmente el sistema.
- Puede ser integrado al ambiente *Windows* permitiendo de esta manera a los desarrolladores, generar nuevas aplicaciones o mantener las ya existentes.
- Permite desarrollar o migrar aplicaciones para que utilicen a *PostgreSQL* como servidor de base de datos (32).

### 1.7.8. ORM *NHibernate*.

Es un asignador relacional de objetos ORM<sup>16</sup>. *NHibernate* es un marco que permite el trabajo con la base de datos relacional de una manera orientada a objetos. Se pueden persistir los objetos de una base de datos y se pueden cargar dichos objetos. *NHibernate* genera las sentencias SQL para efectuar las operaciones de insertar, actualizar, eliminar y cargar datos. Está diseñado para llevar interacciones con las soluciones RDBMS<sup>17</sup>. Por lo que se encarga justamente de relacionar clases con tablas. De forma muy simple, una tabla se mapea contra una clase, y cada columna contra un atributo de dicha clase. De esta forma se encargará de ocultar la complejidad del acceso a datos, exponiendo solamente objetos. Constituye un puerto .NET de *Hibernate Java* ya que es la conversión de *Hibernate* de lenguaje *Java* a *C#* para su integración en la plataforma .NET. El uso de *NHibernate* permite que la aplicación sea agnóstica en cuanto al motor de base de datos a utilizar en producción, ya que soporta los más habituales en el mercado: *MySQL*, *PostgreSQL*, *Oracle*, *MS SQL Server*. Solamente para trabajar en una base de datos distinta es necesario cambiar una línea en el fichero de configuración (33).

### 1.7.9. Servidores *web*.

Los servidores *web* son aquellos cuya tarea es alojar sitios y/o aplicaciones, las cuales son accedidas por los clientes utilizando un navegador que se comunica con el servidor utilizando el protocolo

---

<sup>16</sup> El **mapeo objeto-relacional** (más conocido por su nombre en inglés, Object-Relational mapping, o sus siglas O/RM, ORM, y O/R mapping) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional utilizando un motor de persistencia.

<sup>17</sup> Un **sistema de gestión de bases de datos relacionales** es aquel que sigue el modelo relacional. Los sistemas de base de datos relacionales son aquellos que almacenan y administran de manera lógica los datos en forma de tablas.



HTTP.<sup>18</sup> Consta de un intérprete HTTP, el cual se mantiene a la espera de peticiones de clientes y le responde con el contenido según sea solicitado. El cliente una vez recibido el código, lo interpreta y lo exhibe en pantalla (34).

**Servidor Apache** es un servidor *web* de *software* libre desarrollado por la *Apache Software Foundation* cuyo objetivo es suministrar páginas *web* a los clientes *web* o navegadores que las solicitan. Corre en diferentes sistemas operativos, que lo distinguen como *software* universal. Es un servidor altamente configurable de diseño modular. Es sencillo de ampliar las capacidades de este servidor, ya que existen muchos módulos para Apache que son adaptables a este. Posee alta configurabilidad en la creación y gestión de los logs (35), (36).

**Internet Information Server** es un componente de *Windows* que convierte un sistema *Windows* en un servidor de *Internet*. Engloba un conjunto de herramientas destinadas al control de servicios de Internet como el *web*, FTP, correo y servidores de noticias. Incluye el soporte necesario para la creación de páginas dinámicas en el servidor mediante el lenguaje ASP. Este servicio de *software* admite la creación, la configuración y la administración de los sitios *web*, así como otras funciones de Internet. Los requisitos de *hardware* para que funcione en buenas condiciones, dependen de las exigencias específicas del servidor, del número potencial de usuarios conectados, de eventuales interconexiones con base de datos, entre otros. En la actualidad es el segundo más popular sistema de servidor *web* (37).

#### 1.7.10. Cliente *TortoiseSVN-1.7.6*.

El control de versiones, es el arte de manejar cambios en la información. *Tortoise SVN* es un cliente gratuito de código abierto para el sistema de control de versiones *Subversion*. *Tortoise SVN* maneja ficheros y directorios a lo largo del tiempo. Estos ficheros son almacenados en un repositorio central. Este repositorio recuerda todos los cambios que se hayan hecho a sus ficheros y directorios. Esto garantiza que se pueda realizar una recuperación de las versiones antiguas de los ficheros, y examinar cuándo y cómo se realizaron cambios y quién fue la persona que realizó los cambios. Estos sistemas están diseñados para manejar árboles de código fuente. Poseen características que son distintivas para el desarrollo del *software* como: el entendimiento nativo de los lenguajes de programación y proporcionan herramientas para compilar *software* (38).

---

<sup>18</sup> Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse.



## **1.8 Análisis de las medidas de seguridad a incorporar en el documento de identificación y selección del estándar para la definición de las características físicas durante el proceso de confección.**

### **1.8.1. Medidas de seguridad.**

Teniendo en cuenta las medidas de seguridad propuestas por la OACI, se definieron los siguientes elementos de seguridad, los cuales serán incluidos en la personalización de los documentos de identificación de la Universidad de las Ciencias Informáticas, garantizando el fortalecimiento en cuanto a la seguridad de los mismos:

La imagen fantasma, es una reproducción de la foto del portador del documento, la misma evita las falsificaciones e imitaciones.

Micro textos, conjunto de letras que poseen tamaños muy pequeños. Estos son poco visibles a simple vista, por lo que es necesario utilizar un instrumento para comprobar que existen en el documento.

Elementos de lectura mecánica, en este caso los códigos bidimensionales, los mismos permiten almacenar más información que un código de barras unidimensional. Además garantiza que en el documento exista un área fija, para leer mecánicamente los datos almacenados en dicho código a través de métodos OCR<sup>19</sup>.

Error intencional incorporado dentro de un texto en microimpresión, no son más que errores ortográficos impuestos a cualquier palabra en el documento, que permita reconocer un documento falso, en caso de que la palabra que lo posea se encuentre escrita de forma correcta.

### **1.8.2. Selección del estándar para la confección física del documento.**

Para la confección física del documento de identificación la UCI como cliente seleccionó el estándar ISO/IEC 7810 con la talla ID-1. Por lo que la solución a desarrollar debe ser capaz de emitir un documento que cumpla con la composición física seleccionada.

## **1.9 Selección de la metodología y el marco de trabajo.**

Para definir la metodología y el marco de trabajo en la solución informática que se desea desarrollar, se valoran las principales características, ventajas, desventajas y el análisis del avance paulatino a nivel mundial, en cuanto al empleo de las mismas.

Al conocer las características que presenta una metodología pesada como RUP, la cual genera muchos artefactos y roles, además, se aplica para grandes grupos de desarrollo y el cliente no forma

---

<sup>19</sup> Reconocimiento óptico de caracteres.

parte durante todo el proceso de desarrollo, se decide realizar investigaciones sobre el funcionamiento de las metodologías ágiles.

Las metodologías ágiles constituyen las más recomendables, las mismas generan pocos artefactos y roles, además el cliente forma parte en todo el proceso de desarrollo. Es aplicable a grupos pequeños de desarrollo, aproximadamente para menos de 10 integrantes.

La metodología de desarrollo de *software* seleccionada para la solución que se desea implementar es FDD. La misma está definida para proyectos cortos y con un equipo de desarrollo pequeño, y la solución a implementar cumple con estas características, además esta metodología se basa en un proceso iterativo, permitiéndole al cliente, recibir después de cada iteración un pedazo funcional del programa y posibilitándole estar informado de la situación del mismo, de manera que si el desarrollo del producto, se aleja de sus propósitos, pueda intervenir. La metodología FDD está basada en la planeación e iteración por funcionalidades, que permite identificar los requisitos con mayor prioridad, y diseñarlos a través de este proceso iterativo. Esto facilita el diseño de la solución, la descripción textual y la revisión de los mismos antes de comenzar a implementar. Otra característica es que propone la existencia de una jerarquía en el equipo de desarrollo, donde el jefe de proyecto y el arquitecto guían y supervisan a los programadores.

Entre las herramientas CASES que fueron estudiadas, en este caso la herramienta *Rational Rose* y *Visual Paradigm*. La herramienta que se empleará para modelar con el Lenguaje Unificado de Modelado (UML) es el *Visual Paradigm*, ya que en la etapa de requisitos apoya cada uno de los artefactos que genera la metodología, a través de diagramas. Es disponible en diferentes plataformas de sistemas operativos y permite modelar los diagramas a partir del código, es decir una vez implementada la aplicación, se genera un diagrama, para corroborar si se corresponde con el diseñado anteriormente por los analistas. Además ya se posee experiencia con esta herramienta en el proyecto productivo.

Se empleará la notación BPMN ya que es un estándar internacional para el modelado de procesos. Es una notación de modelado independiente con respecto a cualquier metodología de procesos. Además disminuye la brecha entre los procesos de negocio y la implementación de ellos. También modela los procesos de forma unificada y generalizada permitiendo el entendimiento de los mismos, por todas las personas involucradas en el proyecto. Es fácil de aplicar y posibilita la automatización de las actividades.

La tecnología escogida es ASP.NET, ya que es una plataforma de programación *web* notable y unificada que proporciona los servicios suficientes, para que un programador cree aplicaciones *web*. Permite realizar aplicaciones más seguras, escalables y estables. Posee un entorno basado en .NET, por lo que permite crear aplicaciones en cualquier lenguaje que sea compatible con la plataforma. Está

diseñado para funcionar con herramientas de programación como *Microsoft Visual Studio*, siendo este el entorno escogido para programar. El *Visual Studio* simplifica la creación, depuración e implementación de las aplicaciones. Además posee herramientas de pruebas, crea diagrama de clases UML a partir del código y entre otras facilidades. El lenguaje de desarrollo que se utilizará es C# ya que es un lenguaje sencillo, orientado a objetos que permite manipular errores. Posee todas las características del paradigma orientado a objetos como: la encapsulación, herencia, polimorfismo. Además de que ahorra tiempo en la programación ya que posee una librería de clases bien diseñada.

Se decide utilizar como sistema gestor de base de datos a *Postgre SQL* ya que es un gestor estable y con buen rendimiento. Además es extensible porque el código fuente está disponible de forma gratuita. Posee gran capacidad de almacenamiento. Puede operar sobre distintas plataformas. Tiene un buen sistema de seguridad para gestionar contraseñas, grupos de usuarios, usuarios y permite la alta concurrencia de estos últimos.

Para la persistencia de los objetos de la base de datos se empleará el *ORM NHibernate*, ya que soporta el lenguaje orientado a objetos, la herencia, el polimorfismo, la composición y la estructura de colecciones de .NET, incluyendo colecciones genéricas. Soporta contextos de persistencia de larga duración y es de código abierto, porque se encuentra bajo la licencia de LGPL (Licencia Pública de GNU<sup>20</sup>).

Además se empleará el *Internet Information Server* como servidor *web*, el cual proporciona herramientas y funciones necesarias para la administración de un servidor *web* seguro. Además de que garantiza un método, para transferir datos entre el cliente y el servidor de forma segura e impide que usuarios con direcciones IP conocidas, obtengan acceso no autorizado al servidor.

Por último se ha empleado el cliente *Tortoise Subversion*, el mismo permite manejar cada uno de los archivos y modificaciones que serán realizadas a la aplicación.

## 1.10 Conclusiones parciales.

El desarrollo de un amplio estudio de los sistemas dedicados a la emisión de documentos de identificación ha proporcionado identificar los procesos por los que transcurre un documento de identificación para ser emitidos, así como las tecnologías en cuanto a elementos de seguridad que son incorporados en el mismo. Además permitió detectar al SPDI como sistema a integrar.

Se realizó un estudio sobre las medidas de seguridad que les son incorporadas a los documentos de identificación, así como los estándares internacionales que definen su creación, posibilitando una comprensión de estas características y seleccionando para la personalización del documento, las

---

<sup>20</sup> proyecto GNU iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre: el sistema GNU.

medidas de seguridad: imagen fantasma, código de barras, micro textos y errores ortográficos. En el caso de los estándares internacionales para la confección física de la tarjeta se seleccionó el estándar ID-1 normado por la ISO 7810 para regir el tamaño.

Se han identificado tecnologías, herramientas y estándares que a través de sus beneficios, permiten realizar el modelado y la implementación de la futura solución. Se han expuesto las razones por las cuáles se realizará una solución informática para la personalización, control de calidad y entrega del documento de identificación. Lo que garantiza la emisión del documento de una forma más segura, dando respuesta a las necesidades actuales de la Universidad de las Ciencias Informáticas.

## Capítulo II. Análisis y diseño de la solución.

### 2.1. Introducción.

En el presente capítulo se describe detalladamente el flujo actual de los procesos que se realizan en la oficina de acreditación de la UCI. Se modela el negocio que se propone, donde se identifican cada uno de los responsables que intervienen en los procesos, se detallan los mismos y se plantea una propuesta del sistema a partir de los requerimientos funcionales y no funcionales. Además se muestra con el análisis y diseño, la solución, expuesta mediante los diferentes diagramas de clases. También se muestra la propuesta de arquitectura para los procesos de personalización y entrega.

### 2.2. Modelo de negocio.

#### Flujo actual de los procesos.

Actualmente la oficina de acreditación de la Universidad de las Ciencias Informáticas es la encargada de llevar a cabo la impresión del documento de identificación, en este caso el solapín o credencial. Los datos y las fotos de los estudiantes, profesores o trabajadores son cargados de las bases de datos Personal UCI para los trabajadores y profesores; y de la base de datos Gestión Universitaria para los estudiantes. El sistema de identificación UCI lee los datos de cualquiera de las bases de datos existentes, basándose en el número del solapín o credencial y el número de carnet de identidad. Una vez cargado estos datos, se procede a personalizar el documento, si durante este proceso ocurre algún error o dificultad, se imprime nuevamente la credencial, y luego se le realiza la entrega de forma manual al estudiante o trabajador correspondiente.

#### Concepción de los procesos de personalización y entrega.

**Proceso de personalización:** Proceso donde se imprimen los datos establecidos según el documento realizándose el proceso de calidad de la impresión de los mismos.

**Proceso de entrega:** Proceso encargado de realizar la entrega al individuo de su documento de identificación personalizado. Durante este proceso se verifica la identidad de la persona a quien se entrega el documento.

#### Descripción del proceso personalización.

Nombre:	Personalización.
Objetivos:	Personalizar el documento.
Evento(s) que lo generan:	Necesidad de personalizar un documento.
Precondiciones:	Documentos listos para personalizar.
Poscondiciones:	Documento personalizado.

## Capítulo II. Análisis y diseño de la solución

Reglas de negocio:	Las órdenes de impresión pertenecen a un mismo tipo de documento. Una orden de impresión debe estar compuesta por al menos una solicitud de impresión. El envío de órdenes y lotes de documentos del sistema al Sistema de Personalización de Documentos de Identificación (SPDI) se realizará mediante servicios <i>web</i> .
Responsables:	Responsable de personalización. Sistema SPDI.
Clientes internos:	-
Clientes externos:	-
Entradas:	Documento virgen.
Salidas:	Documento personalizado y con calidad.
Actividades:	Solicita crear órdenes de impresión. Crea órdenes de impresión. Envía órdenes de impresión. Recibe órdenes de impresión. Preparar datos. Imprimir órdenes. Comprobar calidad. Realiza solicitud de lotes. Recibe solicitud de lotes. Crea lotes. Envía lotes. Recibe lotes. Realizar supervisión.

Tabla 1 Proceso de personalización. Fuente: Elaboración propia.

Para realizar un estudio más detallado del flujo básico y paralelo, consultar Anexo 1.

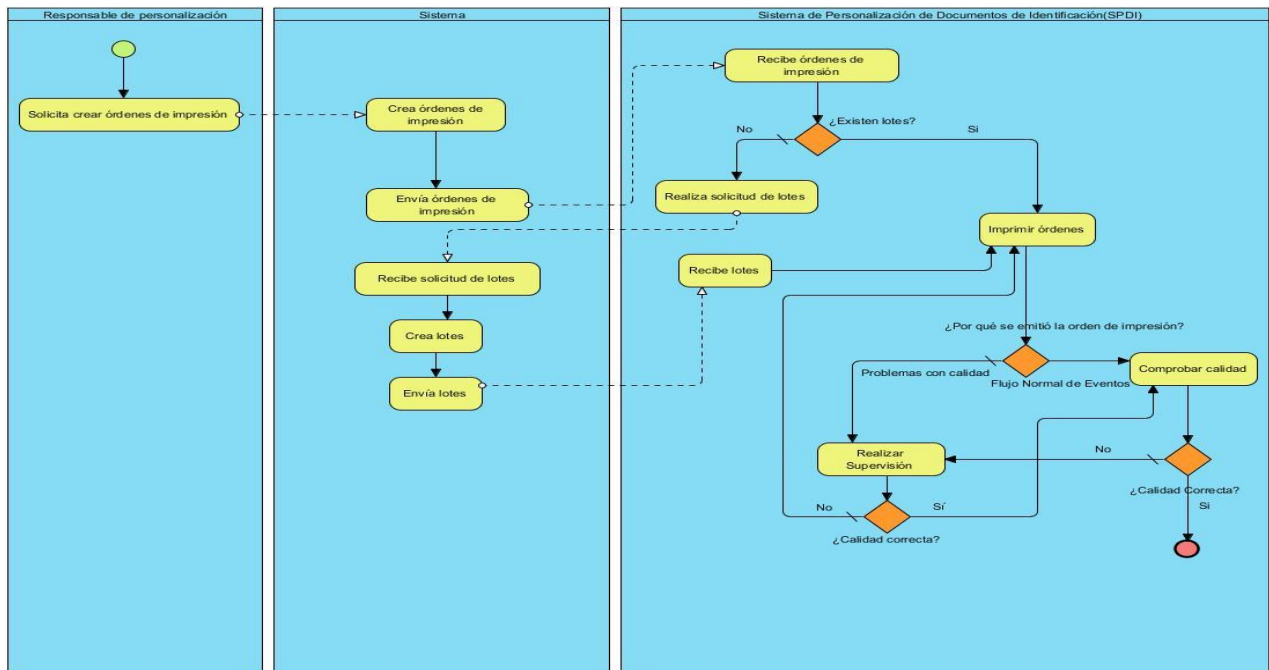


Figura 1. Diagrama del proceso personalización. Fuente: Elaboración propia.

Para ver el diagrama del proceso de personalización más detallado, consultar Anexo 2.

## Descripción del proceso de entrega.

Nombre:	Entrega.
Objetivos:	Realizar la entrega del documento de identificación.
Evento(s) que lo generan:	Solicitud de entrega del documento.
Precondiciones:	-
Poscondiciones:	Queda entregado el documento de identificación.
Reglas de negocio:	Solo se entregará el documento al titular, o en otro caso a la persona responsable, previamente registrada en el sistema según sea el tipo de trámite.
Responsables:	Responsable de entrega.
Clientes internos:	-
Clientes externos:	-
Entradas:	Documento de identificación personalizado.
Salidas:	Documento de identificación entregado.

# Capítulo II. Análisis y diseño de la solución

Actividades:	<p>Solicita documento.          Recibe solicitud.          Busca trámite.          Recibe notificación.          Recibe resultado.          Muestra datos del trámite.          Verifica identidad.          Actualiza el estado del documento a "Entregado".          Actualiza el estado del trámite a "Efectivo".          Notifica cierre de trámite.          Entrega del documento.          Recibe documento.          Captura datos.          Realiza lectura de los datos.          Compara datos.</p>
--------------	---

Tabla 2. Proceso de entrega. Fuente: Elaboración propia.

Para ver el flujo básico y paralelo del proceso de entrega más detallado, consultar Anexo 3.

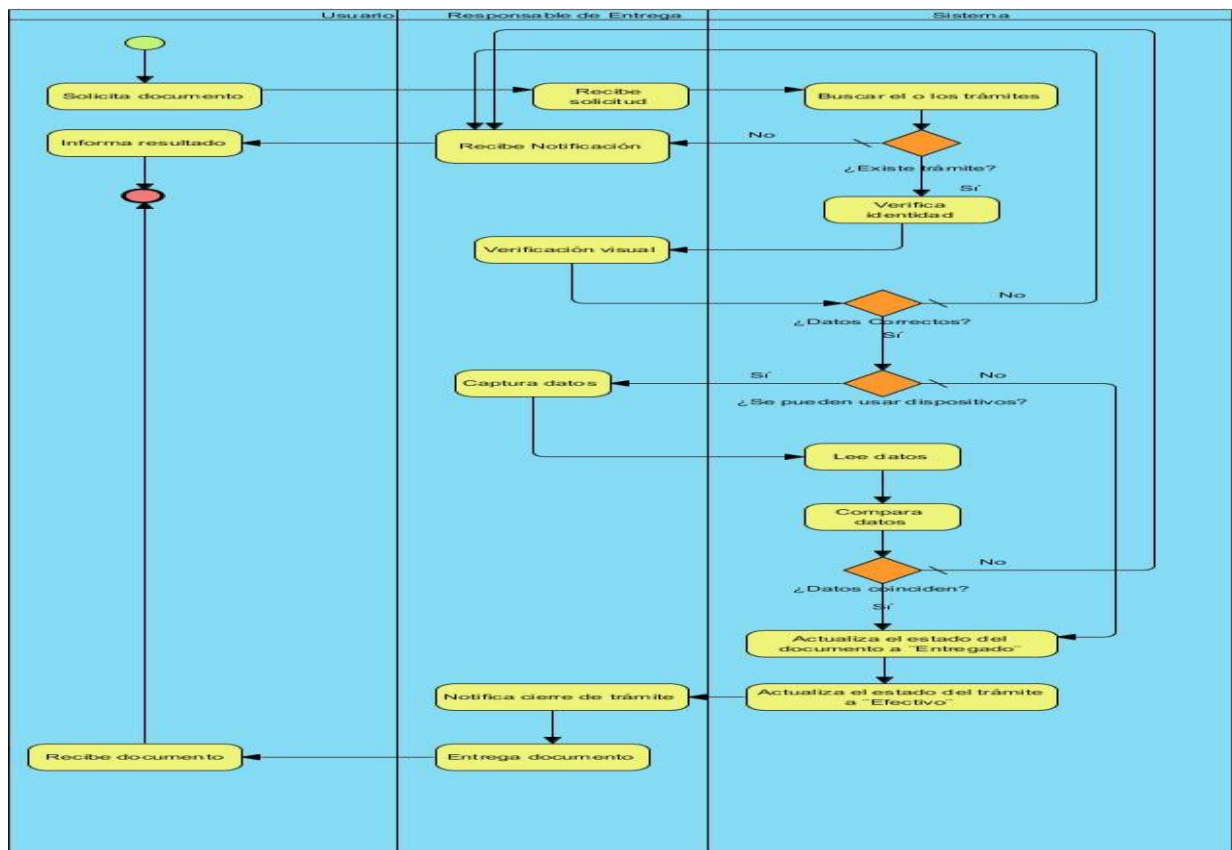
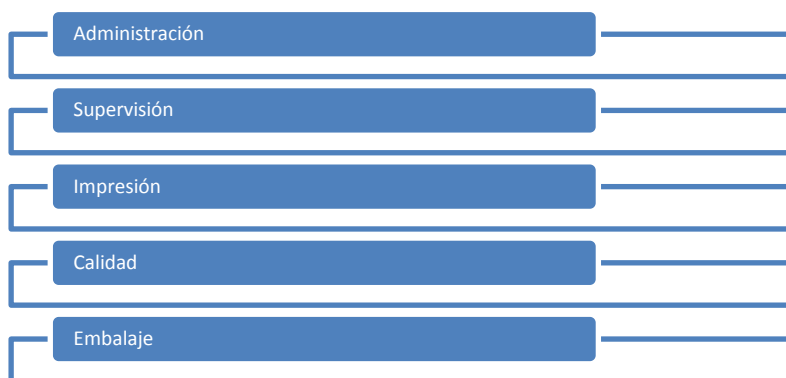


Figura 2. Diagrama del proceso de entrega. Fuente: Elaboración propia.



### 2.3. Análisis de posibles implementaciones o componentes existentes que puedan ser utilizados.

El sistema detectado para utilizar en la creación de la solución para la personalización, control de calidad y entrega de documentos de identificación para la UCI es el Sistema de Personalización de Documentos de Identificación (SPDI), el mismo brinda varias facilidades ya que posibilita el desarrollo del componente de personalización perteneciente a la Plataforma Modular de Identificación y Control de Acceso (PMICA) y contiene desarrollado un componente para el control de la calidad de los documentos una vez personalizados. El SPDI está compuesto por varios componentes, los mismos se muestran en la siguiente figura:



**Figura 3. Componentes del Sistema de Personalización de Documentos de Identificación.**Fuente:Elaboración propia.

El componente Administración está compuesto por las diferentes funcionalidades:

- Gestión de los puestos de trabajo.
- Edición y configuración de tipos de documentos de identificación.
- Gestión de plantillas de documentos de identificación.
- Creación y definición de reglas de asignación.

El componente Supervisión presenta:

- Listado de solicitudes de documentos de identificación en espera del inicio de la jornada laboral.
- Realización de solicitudes de lotes.
- Asignación de lotes a los puestos de trabajo.
- Reasignación de órdenes a los puestos de trabajo.
- Seguimiento del control de flujo del EMIPAS.
- Seguimiento del control de flujo de las órdenes efectuadas.

- Emisión de reportes sobre el control de los documentos de identificación personalizados.
- Reimpresión y supervisión de órdenes que contienen solicitudes anuladas.
- Seguimiento de los fallos de servicio.

El componente Impresión presenta:

- Impresión de las órdenes listas para realizar este proceso.
- Laminado de las órdenes listas para realizar este proceso.

El componente de Calidad presenta:

- Comprobación de la calidad de las órdenes emitidas.

El componente Embalaje presenta:

- Envío a las oficinas de entrega las órdenes listas para realizar el proceso de embalaje.
- Envío de lotes personalizados a los almacenes.

A partir del estudio sobre los diferentes componentes que proporciona el SPDI se llegó a la conclusión que serán reutilizados los siguientes:

Del componente **Administración** se utilizarán las funcionalidades de Gestión de los puestos de trabajo, que permite ver todos los locales creados para los cuales serán enviadas las órdenes de impresión, además de que permite la modificación o eliminación de dichos puestos. Otra es la Edición y configuración de los tipos de documentos de identificación, que permitirá la edición de las plantillas basadas en los estándares ISO/IEC7810.

Del componente de **Supervisión** se emplearán las funcionalidades de realizar Solicitud de lotes, funcionalidad encargada de solicitar el material o las formas valiosas para la confección de los documentos. También se empleará la actividad Asignar lotes a los diferentes puestos de impresión, que consiste en asignar cada lote a los diferentes puntos de impresión. Se utilizará la actividad Iniciar jornada laboral, la cual permite hacer un reporte de la cantidad de solicitudes a imprimir en el día y las pendientes en caso de no poder imprimirlas todas. La utilización de la actividad Reasignar lotes a los diferentes puestos de impresión, permite que en caso de presentarse una emergencia en un puesto de impresión, como por ejemplo una impresora rota, posibilita enviar la cantidad de órdenes con solicitudes sin terminarse de imprimir hacia otro puesto de impresión. La utilización del Control del flujo de información, permite tener un control de cómo ha transcurrido el proceso de cada orden hacia los diferentes puestos de impresión. Además con la reutilización de la actividad de Comprobar calidad de reimpresión y Supervisión de las órdenes con solicitudes anuladas, permite que las solicitudes tendrán la oportunidad de ser supervisadas su calidad y luego podrán ser reimprimadas nuevamente.

Del componente de **Impresión** se utilizará la impresión de las órdenes listas para imprimir, la misma permite la inclusión de medidas de seguridad durante la impresión del documento, esto responde a las medidas de seguridad propuestas por la OACI para la emisión de un documento seguro.

Del componente de **Calidad** se utilizará la comprobación de la calidad de cada una de las órdenes que se emitan, la misma se realiza de forma visual y por dispositivos y garantizará que el documento se ajuste a las reglas establecidas por las normas internacionales, además de comprobar la veracidad de los datos impresos.

### 2.4. Roles del sistema.

Los roles son las funciones que desempeñan las personas que tienen el privilegio de interactuar en el sistema en la parte que le es asignada.

#### Componente de personalización.

Rol	Objetivo
Responsable de personalización	Encargado de crear órdenes de impresión, enviar órdenes de impresión al sistema de personalización.

Tabla 3. Rol del componente de personalización. Fuente: Elaboración propia.

#### Componente de entrega.

Rol	Objetivo
Responsable de entrega	Cumple con la función de realizar la entrega de los documentos personalizados y con la calidad requerida. Notifica los documentos entregados. Registra a la persona a quien entrega los documentos.

Tabla 4. Rol del componente de entrega. Fuente: Elaboración propia.

### 2.5. Especificación de los requisitos funcionales.

Luego de realizar el modelado del negocio que brinda la comprensión entre el cliente y el equipo de desarrollo, se especificarán las características y condiciones que debe cumplir las funcionalidades del sistema a desarrollar. Según la IEEE (*Standard Glossary of Software Engineering Terminology*) define un requerimiento como la condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Es además la condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema, para satisfacer un contrato, estándar u otro documento impuesto formalmente (39).

#### Catálogo de requisitos funcionales.

Para la obtención de los requisitos funcionales se tuvieron en cuenta varias entrevistas a las personas involucradas en la gestión de los trámites de los procesos de personalización y entrega. Con la

capacitación dada acerca de cómo se realizaban estos procesos en la entidad, luego de conformar el modelado de procesos del negocio, se obtuvo una visión de las principales actividades que ayudarían a identificar lo que debe hacer el sistema. Los siguientes requisitos abarcan los componentes de personalización y entrega respectivamente. Para ver los requisitos generales, consultar Anexo 4.

### Componente de personalización.

**RFP1.** Crear orden de solicitud de impresión.

**RFP2.** Enviar orden de impresión al sistema de personalización.

**RFP3.** Consultar estado de la orden de impresión.

**RFP4.** Obtener resultado de la asignación del documento.

### Componente de entrega.

**RFE1.** Registrar el tipo de entrega.

**RFE2.** Mostrar personas autorizadas a recoger el documento.

**RFE3.** Registrar entrega del documento.

**RFE4.** Imprimir comprobante de entrega.

### Descripción de los requisitos funcionales.

A continuación se muestran algunas de las descripciones de los requisitos funcionales de los componentes de personalización y entrega. Seguidamente se muestra la descripción del requisito funcional “Crear orden de solicitud de impresión” que pertenece al componente de personalización.

Precondiciones	Deben existir solicitudes de impresión. El usuario debe estar autenticado con el rol responsable de personalización.
Funcionalidades asociadas	RFP1, RFG9, RFG10.
Conceptos tratados	Solicitud de impresión, Orden de impresión.
Descripción básica	<p>1. El sistema muestra la interfaz Listar solicitudes de impresión, con un listado que contiene las solicitudes de impresión, con los siguientes datos y opciones:</p> <p>Datos:</p> <ul style="list-style-type: none"> <li>• Identificador (Corresponde al identificador de la solicitud de impresión)</li> <li>• Fecha y hora (Fecha en la que se creó. Formato dd/mm/aaaa, 00:00)</li> <li>• Estado (Corresponde al estado de la solicitud)</li> <li>• Tipo de documento</li> <li>• Nombre y Apellidos</li> </ul> <p>Opciones:</p> <ul style="list-style-type: none"> <li>• Filtrar (Por criterios)</li> </ul>

- Crear orden
- Cancelar

1.1. Si el usuario autenticado introduce un criterio de selección y presiona la opción Filtrar, el sistema muestra las solicitudes que correspondan con el criterio introducido en el listado de solicitudes:

Parámetros de filtro:

- Carnet de identidad
- Tipo de documento
- Nombre y Apellidos

1.2. Si el usuario autenticado selecciona las solicitudes con que se creará la orden y presiona la opción Crear orden, el sistema confecciona una orden de impresión y luego es enviada al Sistema de Personalización de Documentos de Identificación (SPDI) para que sea personalizada con los siguientes parámetros:

Parámetros de la orden de impresión:

- Listado de solicitudes
- Prioridad
- Tipo de documento
- Oficina

1.2.1. El sistema verifica que todos los datos sean correctos. En ese caso elimina las solicitudes del listado y muestra un mensaje indicando que la orden se ha creado de forma satisfactoria. (En caso contrario consultar: Descripción alterna 1).

1.3. Si el usuario desea utilizar el paginado, basta con seleccionar el número de la página o las opciones Anterior y Siguiente.

### Prototipos

**Interfaz: Listar solicitudes de impresión**

Filtrar

Criterios: Identificador

Identificador: 000001

Filtrar

**Listado de solicitudes:**

<input type="checkbox"/>	Identificador	Fecha	Estado	Tipo de documento
<input checked="" type="checkbox"/>	000001	12/01/2013	Creada	Credencial
<input type="checkbox"/>	000002	12/01/2013	Creada	Credencial

Crear orden Anterior 1 2 3 4 5 Siguiente

Descripción alterna

Descripción alterna 1

1. En caso en que los datos sean incorrectos, el sistema muestra un mensaje indicando el error.

## Capítulo II. Análisis y diseño de la solución

Prototipos	-
Validaciones	- Si no existen solicitudes de impresión el listado se mostrará vacío. - Al crear una orden de impresión se debe verificar que contenga al menos una solicitud de impresión y que las solicitudes que la conforman pertenezcan al mismo tipo de documento.
Postcondiciones	Queda conformada una orden de impresión.
Requisito no funcional	-
Servicios	-
Componente	-

**Tabla 5. Crear órdenes de solicitud de impresión. Fuente: Elaboración propia.**

Seguidamente se muestra la descripción del requisito funcional “Registrar entrega del documento” que pertenece al componente de entrega.

Precondiciones	Debe existir un documento personalizado con calidad satisfactoria, que contenga como estado “Enviada a oficina de entrega”. El usuario debe estar autenticado con el rol responsable de entrega.
Funcionalidades asociadas	RFG1, RFG2, RFG3, RFG4, RFG5, RFG6, RFG7, RFE1, RFE2, RFE3, RFE4.
Conceptos tratados	Documento de identificación, Entrega de documento, Comprobante de entrega.
Descripción básica	<p>1. Si el usuario autenticado selecciona la opción Entrega, el sistema muestra una interfaz que contiene un listado de los documentos listos para entregar y la opción de filtro:</p> <p>Datos del listado:</p> <ul style="list-style-type: none"> <li>• Carnet de identidad del ciudadano al que pertenece el documento.</li> <li>• Nombres y apellidos del ciudadano al que pertenece el documento.</li> <li>• Tipo de documento</li> <li>• Estado del documento.</li> </ul> <p>Opciones:</p> <ul style="list-style-type: none"> <li>• Filtrar</li> <li>• Registrar entrega</li> <li>• Paginado</li> </ul> <p>1.1. Si el usuario autenticado introduce un criterio de selección y presiona la opción Filtrar, el sistema muestra las solicitudes que correspondan con el criterio introducido en el Listado de documentos listos para entregar:</p> <p>Parámetros para el filtrado:</p> <ul style="list-style-type: none"> <li>• Carnet de identidad del ciudadano al que pertenece el documento.</li> <li>• Tipo de documento</li> <li>• Nombre y Apellidos</li> </ul> <p>1.2. Si el usuario autenticado selecciona un documento del listado</p>

y presiona la opción Registrar entrega, el sistema muestra una interfaz que contiene los siguientes datos y opciones:

Datos:

Tipos de entrega :

- Titular
- Responsable

Datos del documento:

- Primer nombre
- Segundo nombre
- Primer apellido
- Segundo apellido
- Carnet de identidad
- Foto

Opciones:

- Filtrar por (Carnet de identidad del ciudadano responsable de recoger el documento, en caso de realizar la entrega a un Responsable) Consultar Descripción alterna 1.
- Registrar Entrega
- Cancelar

1.2.1. El usuario presiona la opción Registrar Entrega, el sistema actualiza el estado del documento a “Entregado” y la fase del trámite a “Efectiva”, luego se muestra una interfaz para imprimir el comprobante correspondiente a dicha entrega y con la opción Siguiete regresa a la interfaz anterior con la próxima entrega. Consultar Descripción alterna 3.

1.2.2. .En caso de seleccionar la opción Cancelar consultar Descripción alterna 2).

## Prototipos

Interfaz: Listado de documentos listos para entregar

Filtrar

Criterios: Identificador

Identificador: 695251

Filtrar

Listado de documentos:

Identificador	Nombres y apellidos	Tipo de documento
<input checked="" type="checkbox"/> 695251	Juan Perez Delgado	Credencial
<input type="checkbox"/> 95656	Yoley Estevan Guzman	Credencial

Registrar entrega Anterior 1 2 3 4 5 Siguiete

# Capítulo II. Análisis y diseño de la solución

Descripción alterna

Descripción alterna 1

Descripción alterna

1. El usuario autenticado introduce un criterio de búsqueda y presiona la opción Buscar, si el sistema encuentra a la persona, muestra los datos encontrados y regresa al flujo básico al paso 1.2.1 (En caso de que no se encuentre la persona el sistema muestra un mensaje indicando el error y no puede registrar la entrega al Responsable).

Prototipos

Descripción alterna 2

2. Si el usuario selecciona la opción Cancelar, el sistema regresa a la interfaz anterior.

Prototipos

-

Prototipos

-

Descripción alterna 3



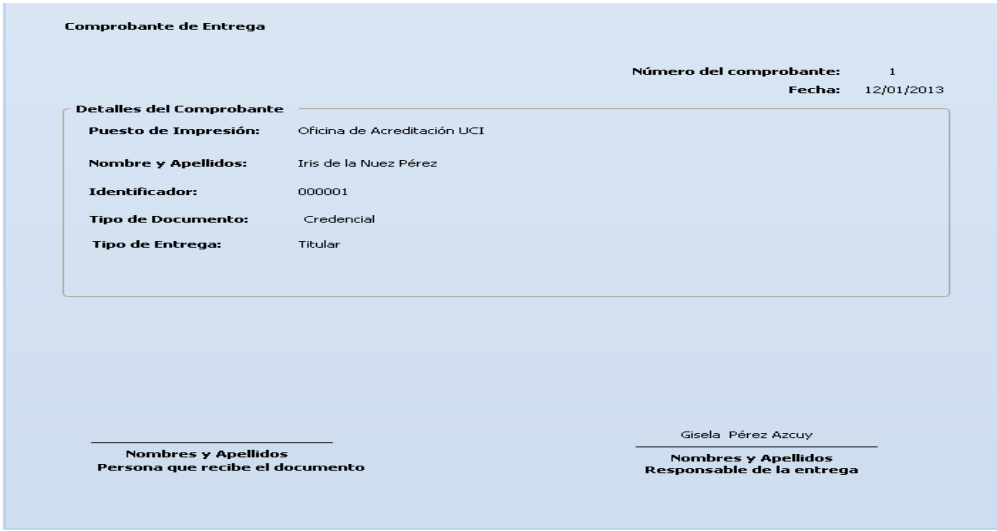
	<p>3. El sistema genera un comprobante de entrega en formato PDF, que contiene los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Número del comprobante</li> <li>• Fecha (Formato dd/mm/aaaa)</li> <li>• Puesto de impresión</li> <li>• Nombres y Apellidos (Corresponden a la persona titular del documento)</li> <li>• Carnet de identidad del ciudadano</li> <li>• Tipo de documento</li> <li>• Tipo de entrega (Titular, Responsable)</li> </ul>
<p>Prototipos</p> 	
Validaciones	<ul style="list-style-type: none"> <li>- Solo se puede seleccionar un tipo de entrega.</li> <li>- El campo Fecha se corresponde con la fecha del día actual y es generada automáticamente por el sistema.</li> </ul>
Postcondiciones	Queda entregado el documento.
Requisito no funcional	-
Servicios	-
Componente	-

Tabla 6. Registrar entrega del documento. Fuente: Elaboración propia.

### 2.6. Especificación de los requisitos no funcionales.

Los requisitos no funcionales son las propiedades o cualidades que el producto debe de tener. Son características que definen al producto como atractivo, confiable y rápido. Permite que los clientes y los usuarios valoren las características no funcionales del producto. Existen diferentes categorías para su clasificación, a continuación se muestran la obtención de los que fueron definidos durante el estudio realizado.

**Requisitos de soporte:** El soporte de las aplicaciones y componentes utilizados en el desarrollo del sistema será responsabilidad de los proveedores de los mismos.

**Requisitos de software:** Sistema operativo *Windows XP SP3* o superior con antivirus actualizado, navegador *web Mozilla Firefox v17.0* o superior, *Microsoft .Net framework v4.0*. Además del *software* necesario para cada uno de los dispositivos utilizados por el sistema.

**Requisitos de hardware:** PC *Pentium 4* a 2 GHz o superior, mínimo 2 GB de RAM, 250 GB o superior de disco duro.

Para ver más detalladamente la descripción de cada requisito no funcional, consultar Anexo 5.

## 2.7. Arquitectura.

El sistema en su vista más abstracta es una solución cliente - servidor que define la relación entre dos aplicaciones en las cuales una de ellas (cliente) envía peticiones a la otra (servidor) y este último le envía las respuestas.

El estilo arquitectónico utilizado es en capas, por lo que el sistema se encuentra lógicamente dividido en cinco capas definiendo claramente las responsabilidades de cada una, permitiendo reducir el acoplamiento y aumentar la reutilización de las mismas. Esta vista permite la realización de cambios en las capas sin realizar grandes cambios en las demás. La comunicación entre las capas se realizará a nivel de interfaces permitiendo trabajar de manera transparente a las instancias reales.

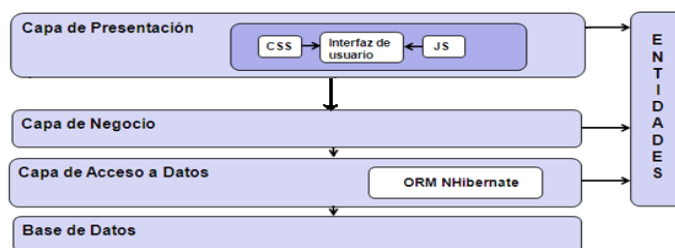


Figura 4. Arquitectura del sistema. Fuente: Tomado del documento Arquitectura del subsistema de identificación.

**Capa de Presentación:** Es la capa donde el sistema interactúa con el usuario, haciendo uso de varias tecnologías para la validación de los datos de entrada así como el uso de componentes. En esta capa se encuentran todas las interfaces que serán mostradas a los usuarios donde para su correcto funcionamiento se hace uso de código *Java Script*, integrando los componentes de *JQuery*, así como archivos *CSS* que contienen los estilos de la aplicación.

**Capa de Negocio:** Encapsula la lógica del sistema, compuesta por clases que manejan las entidades y que valiéndose de la capa de acceso a datos da respuesta a las funcionalidades requeridas, para mostrar correctamente las interfaces al usuario a través de la capa de presentación

**Capa de Acceso a Datos:** Es el componente que da soporte a las funcionalidades de la capa de negocio que se encuentran relacionadas con la fuente de datos. La principal función de esta capa es realizar una implementación de las funcionalidades definidas en las interfaces de la capa de negocio y al mismo tiempo trabajar directamente con la fuente de datos.

**Capa de Entidades:** contiene las clases entidades que se gestionan en la aplicación, las cuales persisten en la base de datos y se muestran en la presentación.

**Base de Datos:** Está constituida por todo el conjunto de tablas y procedimientos que permiten el almacenamiento de la información recolectada y procesada por los procesos. Se encuentra dividida en varias vistas, que separan la gestión de los procesos de los datos de negocio y la información de soporte.

### 2.8. Propuesta de solución.

La solución informática a desarrollar, estará compuesta por los componentes de personalización y entrega, pertenecientes al PMICA y los componentes de administración, supervisión, impresión y calidad pertenecientes al SPDI. Esta integración se realizará a través de servicios *web*. Los cuales serán consumidos por los diferentes componentes de ambos sistemas. A continuación se detalla la forma de integración entre los sistemas.

El componente de personalización está compuesto por las funcionalidades: **crear orden de impresión**, la misma es la encargada de consultar los trámites que están listos para personalizar, una vez consultados se seleccionan y se conforma una orden de impresión la cual es enviada al SPDI a través de un servicio *web*; **control de flujo**, esta funcionalidad consulta y muestra los estados por los que la orden transita en el SPDI, además actualiza en la base de datos del PMICA las órdenes en correspondencia de los estados en que se encuentran en el SPDI; **atender solicitud de lotes**, funcionalidad encargada de atender las solicitudes de lotes de documentos preimpresos realizadas por el SPDI, este sistema para poder comenzar el proceso de impresión necesita de lotes de documentos preimpresos virtuales, por lo que crea la solicitud de los mismos y la envía al componente de personalización y la funcionalidad atender solicitud de lotes le da respuesta enviándolos a través de servicios *web*.

El componente entrega contiene las funcionalidades: **realizar entrega**, la misma consulta el estado de las órdenes que se encuentran en el SPDI, de existir una orden con el estado calidad efectiva, se

procede a entregar, lo que consiste en seleccionar todos los documentos pertenecientes a dicha orden y realizar la entrega a su titular o responsable; **historial de entrega**, la misma muestra un historial de todas las entregas realizadas, permitiendo gestionarlas.

A continuación se muestra en la figura 5, la imagen general de la solución.

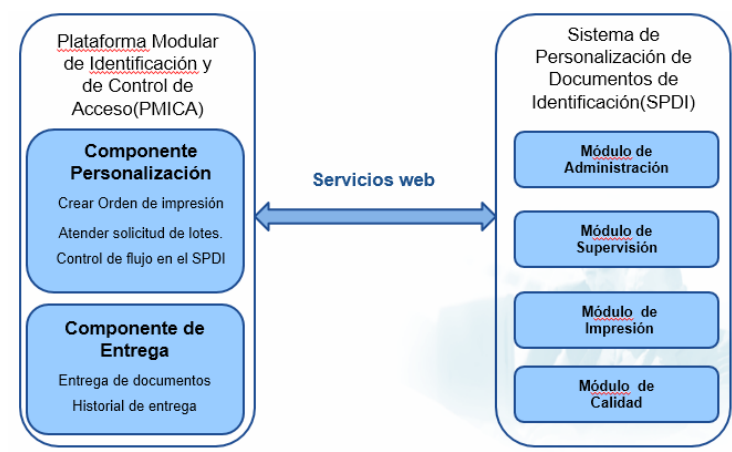


Figura 5. Propuesta de solución. Fuente: Elaboración propia.

Para establecer la integración entre ambos sistemas es necesario un conjunto de modificaciones y configuraciones, que permitan la impresión de documentos de identificación así como el control de calidad de los mismos. Dentro de las modificaciones a realizar para establecer la comunicación entre el componente de personalización del PMICA y el SPDI se encuentran:

- Consumo por parte de los componentes personalización y entrega pertenecientes al PMICA, de los servicios de envío de lotes, órdenes y estados de las mismas publicados en el SPDI.
- Agregar un nuevo tipo de documento: solapín en el SPDI.
- Modificar el *workflow* general del SPDI.
- Modificación de las vistas correspondientes a control de calidad.

Para ver estos cambios más detalladamente, consultar Anexo 6.

### 2.9. Patrones de diseño.

Según Christopher Alexander: "Un patrón describe un problema que ocurre una y otra vez en nuestro entorno, así como la solución a ese problema, de tal modo que se pueda aplicar esta solución, un millón de veces, sin hacer lo mismo dos veces" (40). En general, un patrón posee cuatro elementos esenciales: el nombre del patrón, que permite describir un problema de diseño, con sus soluciones y consecuencias. El problema, el cual describe cuándo aplicar el patrón. La solución que describe los

elementos que constituyen el diseño, relaciones, responsabilidades y colaboraciones y por último las consecuencias que constituyen los resultados, ventajas e inconvenientes de aplicar el patrón.

Luego de realizar un estudio de los diferentes patrones de diseño, se definieron los siguientes, para utilizarlos en el desarrollo de la solución:

**Patrón Singleton:** Garantiza que una clase tenga solo una instancia y proporciona un punto de acceso global a ella. Es necesario cuando hay clases que tienen que gestionar de manera centralizada un recurso. En la solución se encuentra ejemplificada en la clase `Negocio_Personalización` la cual gestiona todo lo referente a las órdenes de impresión .

**Patrones GRASP (Patrones Generales de Software para asignar Responsabilidades):** Patrones Generales de Software para la asignación de responsabilidades. Son una serie de buenas prácticas de aplicación recomendable en el diseño de *software*.

Patrón Creador: Ayuda a identificar quién debe ser el responsable de la creación(o instanciación) de nuevos objetos o clases. Encuentra un Creador que se debe conectar con el objeto producido en cualquier evento. Cuando se escoge al Creador, se da soporte al Bajo Acoplamiento. En la aplicación este patrón es utilizado en la clase `negocio_Entrega` que es la encargada de crear en este caso una orden de impresión para insertarla en la base de datos.

Patrón Controlador: Un controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. En el sistema se hace uso del patrón controlador en la clase `wfrmEntrega.aspx.cs`, la misma maneja todos los eventos relacionados con las funcionalidades de entrega.

### 2.10. Especificación de los diagramas de clases del diseño.

Para la modelación de los diagramas de clases del diseño se han empleado estereotipos *web*. Los estereotipos representan el principal mecanismo de extensión de UML.

Algunos de los estereotipos utilizados son las *server page* o páginas servidoras, que son páginas *web* que contienen código que se ejecuta en el servidor. Las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página. Otro estereotipo es la *client page* o página cliente, la cual es una página *web*, que contiene formato HTML<sup>21</sup>. Constituye una mezcla de datos, presentación y lógica. Estas páginas son interpretadas por el navegador. Cada página cliente es construida por una página servidora. También se encuentran los formularios, los

---

<sup>21</sup> HTML no es más que una aplicación del SGML (Standard Generalized Markup Language), un sistema para definir tipos de documentos estructurados y lenguajes de marcas para representar esos mismos documentos.

cuales conforman una colección de elementos de entrada que son parte de una página cliente. Además se encuentran las clases conectoras las cuales acceden directamente a la base de datos. Estas clases interactúan con las entidades de acceso a datos, las cuales cuentan con el conocimiento de la lógica de acceso a datos, lo que logra una independencia total del gestor de base de datos a utilizar. Otra de las utilizadas son las clases gestoras, las cuales utilizan las clases conectoras que toman los datos que devuelven y conforman las clases o los objetos del negocio. Por otra parte se encuentran las clases entidades las cuales modelan la información. Estas poseen una larga vida, a menudo son persistentes, y constituyen fenómenos, conceptos y sucesos que ocurren en el mundo real (41). Para realizar un estudio más detallado sobre los diagramas de clases del diseño de personalización y entrega, consultar Anexo 7.

## 2.11. Diagramas de secuencia.

Los diagramas de secuencia muestran gráficamente las interacciones del actor y de las operaciones a que dan origen. Estos diagramas se elaboran durante la etapa de diseño, mostrando los escenarios a partir de las funcionalidades, los eventos que generan los actores externos y los eventos internos del sistema (42). A continuación se muestra el diagrama de secuencia de la función crear órdenes perteneciente al componente de personalización, en la figura 6. Para ver el diagrama sobre la función Registrar entrega, consultar Anexo 8.

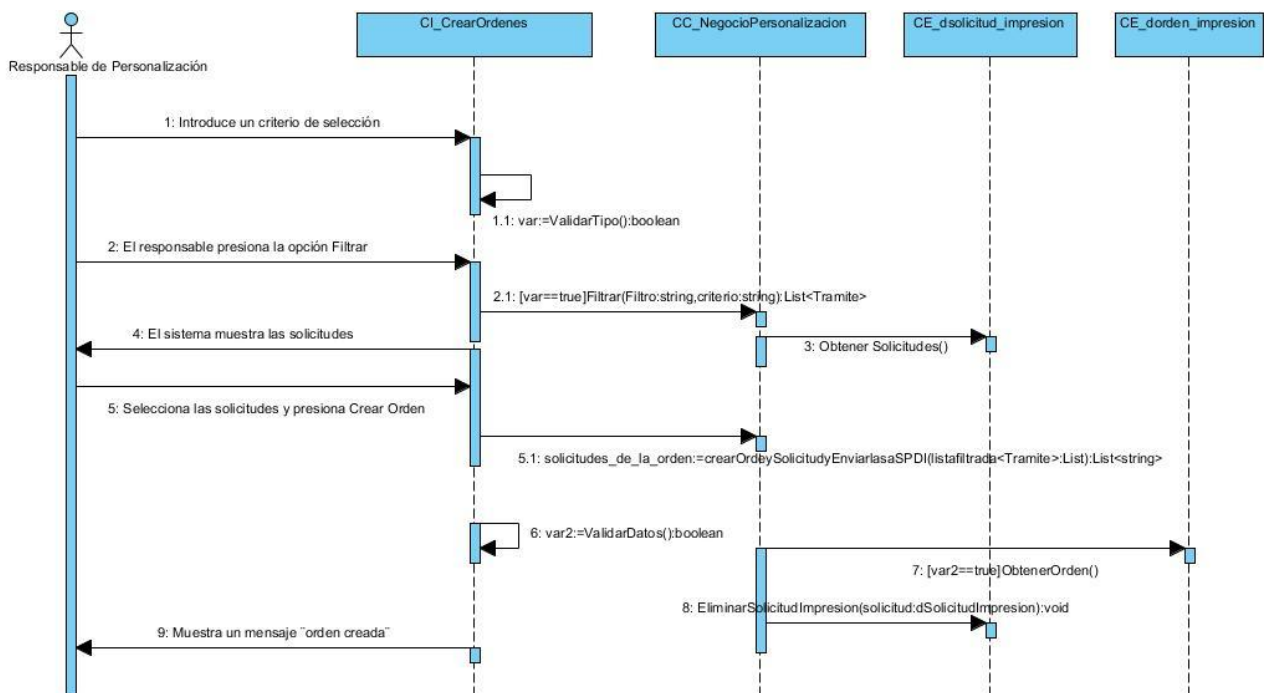


Figura 6. Diagrama de secuencia del componente de personalización: Crear órdenes de solicitud de impresión. Fuente: Elaboración propia.

### 2.12. Servicios del sistema.

Un servicio web es un sitio en la *web* que ofrece la posibilidad de realizar una o múltiples tareas a través de él, cuando se utiliza un servicio *web* lo que se hace es dar las instrucciones a realizar y recibir el resultado, por ejemplo: los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la *web*. Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario (43). A continuación se muestra la descripción del servicio web que se encuentra publicado en el SPDI que será consumido en la Plataforma Modular de Identificación y Control de Acceso (PMICA) para dar respuesta a la funcionalidad : Atender solicitud de lotes. Para realizar un estudio más detallado, consultar Anexo 9.

#### Descripción del servicio.

Nombre	LotServiceQuery	
Descripción	Servicio publicado por el SPDI, encargado de solicitar lotes y recibirlos una vez creados.	
Métodos	Parámetros	Definición
List<LotRequestInfo> getLotRequests	int page, int cant, int columnSelected, int sortMode, string filter, List<ColModel>colModel	Obtiene las solicitudes de lotes.
LotRequest getLotRequest	string identifier	Dado un identificador se obtiene una solicitud de lotes específica.
int countLotRequests	int columnSelected, int sortMode, string filter, List<ColModel>colModel	Cuenta la cantidad de solicitudes de lotes.
void receiveLots	LotRequest lotRequest, List<Lot> lots	Recibe los lotes de una solicitud en específico.
void addLotRequest	int documentType, List<int> lotConfiguration	Inserta en la base de datos una solicitud de lotes.

Tabla 7. Descripción del servicio LotServiceQuery. Fuente: Elaboración propia.

### 2.13. Conclusiones parciales.

El análisis del flujo de los procesos a desarrollar permitió una mayor visión de las actividades a automatizar. Se identificó los componentes a reutilizar pertenecientes al Sistema de Personalización de Documentos de Identificación facilitando el desarrollo de la propuesta de solución. Se definieron los requisitos funcionales los cuales permiten precisar las condiciones que deben ser alcanzadas para

## Capítulo II. Análisis y diseño de la solución

---

lograr la implementación de la solución propuesta. Se definió la arquitectura n-capas para el desarrollo de la solución, que posee dentro de sus principales objetivos asignar correctamente las funcionalidades a cada capa, lo que garantiza la reutilización de las capas inferiores, posibilitando que no tengan dependencias con las superiores. Se diseñaron los diagramas de clases del diseño que permiten establecer una organización y estructura en la implementación, de acuerdo a los requisitos planteados.



## Capítulo III. Implementación y pruebas.

### 3.1. Introducción.

El siguiente capítulo muestra a través de representaciones gráficas, el desarrollo de la implementación del *software*, enfocándose en el diagrama de componentes y diagrama de despliegue de la solución. Se describe detalladamente cómo se lleva el tratamiento de errores, la seguridad así como los estándares de codificación y estilos a utilizar. Se describen los métodos de pruebas para comprobar la validación de cada uno de los requisitos y para verificar el funcionamiento y la calidad del sistema.

### 3.2. Estándares de codificación.

Emplear técnicas de codificación y aplicar buenas prácticas de programación, es necesario para proporcionar un buen rendimiento y generar el código apropiado, lo que permite que el *software* que se vaya a realizar, tenga la calidad adecuada. Un estándar de codificación debe comprender todos los aspectos de la generación de código. Un código fuente debe reflejar un estilo, que refleje como si un único programador hubiera escrito todo el código de una sola vez (44). A continuación se muestran las técnicas empleadas para el desarrollo del *software*. Para realizar un estudio más detallado, consultar Anexo 10.

Convención Pascal: El primer carácter de cada palabra es en mayúscula y el resto en minúscula.

Convención Camel: El primer carácter de cada palabra es en mayúscula (excepto la primera palabra) y el resto en minúscula.

Convención Pascal para nombre de las clases.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace PMICA.Identificacion.Entities
{
    public class Entregadocumento
    {
        public Guid IdEntregaDocumento { set; get; }
        public DateTime FechaYHora{set;get;}
        public Documentoidentificacion DocumentoIdentificacion { set; get; }
        public TipoEntrega TipoEntrega{set;get;}
    }
}
```

Figura 7. Estándar de codificación: Convención Pascal para nombre de las clases. Fuente: Elaboración propia.

Convención Camel para nombre de los métodos.

```
public object filtrarparaMostrar(int jtStartIndex = 0, int jtPageSize = 0, string jtSorting = null)
{
    try
    {
        List<TramiteParaMostrar> listramite = FiltrarparallamarMostrar(jtStartIndex, jtPageSize, jtSorting);
        int UserCount = CountTramites();
        //Return result to jTable
        return new { Result = "OK", Records = listramite, TotalRecordCount = UserCount };
    }
    catch (Exception ex)
    {
        return new { Result = "ERROR", Message = ex.Message };
    }
}
```

Figura 8. Estándar de codificación: Convención Camel para el nombre de los métodos. Fuente: .Elaboración propia.

### 3.3. Tratamiento de errores.

El tratamiento de excepciones permite a la persona que está programando controlar los errores o problemas que ocurren durante la ejecución de un programa. Además de que permite al programador realizar aplicaciones que sean capaces de tolerar fallas, resistentes a los errores y que se pueda ejecutar el programa sin verse afectado por estos inconvenientes o problemas. Resumiendo el tratamiento de errores se basa en capturar excepciones, y una excepción no es más que la respuesta a situaciones anormales o erróneas que se producen en el tiempo en que la aplicación está ejecutándose (45).

Para el tratamiento de errores en el SPDI, este sistema tiene definida una interfaz que almacena una traza de cada una de las excepciones lanzadas en el sistema, además de tener una clase para las excepciones que están definidas para los procesos del negocio, en este caso los *workflows* a través del *Bison FrameWork*, que las hace persistir en un fichero junto con los logs.

Para el tratamiento de errores en los componentes de personalización y entrega, pertenecientes al PMICA, se desarrolló el uso de excepciones, el mismo se realiza a través de las sintaxis de `try {...}` `catch (Exception ex) {...}`, en los diferentes métodos implementados. Además se realizó la validación de los campos de los formularios pertenecientes a las interfaces relacionadas con la aplicación para detectar los campos vacíos mostrando un mensaje de error.

### 3.4. Seguridad en el sistema.

Deben existir controles apropiados para proteger la seguridad interna del proceso de expedición, a efectos de prevenir el uso por empleados y otras personas que puedan obtener acceso a la totalidad o parte del sistema de expedición oficial (46). Esto garantiza de una forma u otra la seguridad del documento de identificación.

La aplicación como está desarrollada en ASP.NET, provee mecanismos de seguridad. Estos mecanismos son la autenticación y la autorización. La autenticación no es más que verificar que el usuario es precisamente quien debe ser y la autorización consiste en limitar los derechos de acceso, concediendo o denegando permisos específicos a la identidad autenticada. En el sistema se han definido roles con su usuario y contraseña. Además a cada rol se le asigna un permiso desde el archivo *web.config* de la aplicación para la navegación, lo que garantiza el acceso adecuado a las diferentes funcionalidades del sistema.

La seguridad se observa de forma transversal en el estilo arquitectónico por capas. Es decir funciona como un todo ya que si un rol no puede acceder a la capa presentación, entonces no puede acceder a ningún recurso de las restantes capas de la aplicación.

En el servidor de base de datos también se establece la seguridad a través de archivos de configuración como son el *pg\_hba.conf* el cual permite que las conexiones de los clientes al servidor de datos se puedan restringir por dirección IP y el archivo *postgresql.conf* permite conectarse remotamente a la base de datos desde otro equipo.

### 3.5. Modelo de datos.

Un modelo de datos no es más que la representación de un fenómeno de la realidad objetiva a través de los objetos, sus propiedades y las relaciones que se establecen entre ellos (47). El siguiente modelo de datos muestra cada una de las entidades persistentes, los atributos y cada una de las relaciones existentes entre ellas. Las entidades representadas a continuación pertenecen a los componentes de personalización y entrega. Para ver la descripción de las entidades y los nomencladores del modelo de datos, consultar Anexo 11.



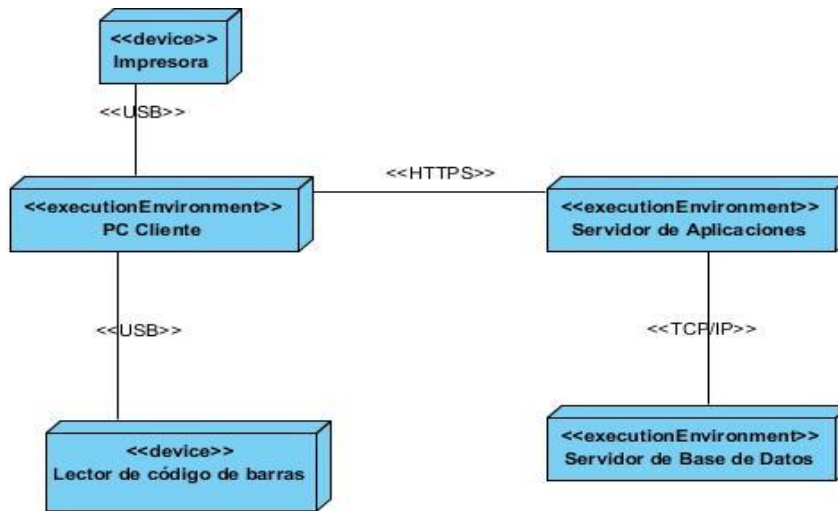


Figura 10. Diagrama de despliegue. Fuente: Elaboración propia.

### 3.7. Diagrama de componentes.

Un diagrama de componentes permite visualizar de manera general la estructura del sistema a desarrollar y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces (49). A continuación se muestra el diagrama de componentes de la solución:

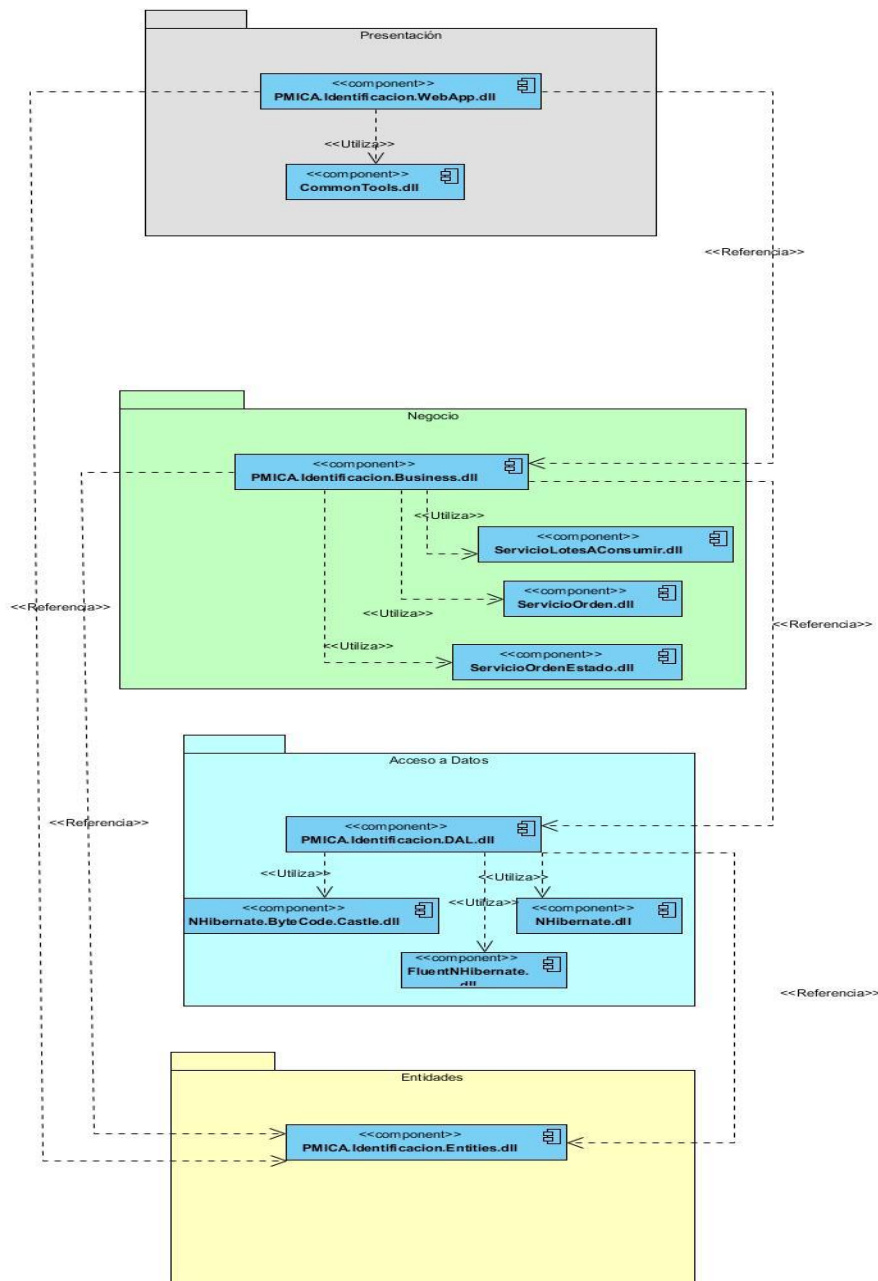


Figura 11. Diagrama de componentes. Fuente: Elaboración propia.

### 3.7.1. Descripción del diagrama de componentes.

En el diagrama de componentes se muestran por paquetes cómo se encuentra conformada la aplicación. El sistema está compuesto por dos componentes fundamentales el *PMICA.Identificacion.WebApp.dll* encargado de la presentación y el *PMICA.Identificacion.Business.dll* para controlar la lógica del negocio. Estos componentes hacen referencia a otros tales como: *PMICA.Identificacion.DAL.dll* que es el encargado del acceso a la base de datos y el

PMICA.*Identificacion.Entities.dll* que contiene las clases entidades que se gestionan en la aplicación, las cuales persisten en la base de datos.

### 3.8. Interfaces del sistema.

La aceptación final de un producto de *software* por parte del usuario depende en gran manera de la percepción que éste tenga del sistema, por lo que la apariencia del mismo se puede considerar como uno de los requisitos de gran importancia en el desarrollo del *software* y esto se logra mediante las interfaces, las cuales deben ser usables, robustas, flexibles e intuitivas. A continuación se muestra la interfaz de usuario en la figura 12 correspondiente al componente personalización. Para visualizar las restantes interfaces, consultar Anexo 12.

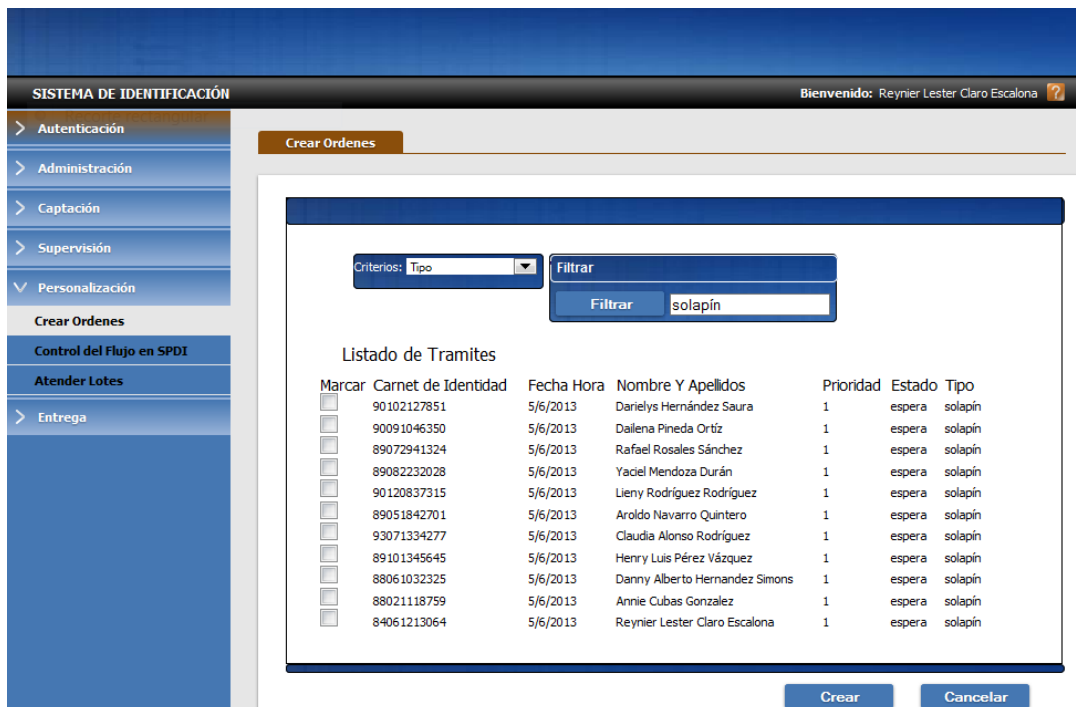


Figura 12. Interfaz crear órdenes. Fuente: Elaboración propia.

### 3.9. Pruebas.

Las pruebas a los sistemas se integran dentro de las diferentes fases del ciclo de vida del *software*. Las mismas se definen como una actividad en la cual un sistema o uno de sus componentes se ejecutan en circunstancias previamente especificadas, registrándose los resultados obtenidos. Seguidamente se realiza un proceso de evaluación en el que los resultados obtenidos se comparan con los resultados esperados para localizar fallos en el *software*. Estos fallos conducen a un proceso

de depuración en el que es necesario identificar la falta asociada con cada fallo y corregirla, pudiendo dar lugar a una nueva prueba (50).

Existen diferentes técnicas de pruebas del *software*, las cuales están constituidas por las pruebas de caja blanca o estructural y las pruebas de caja negra o funcional, además se encuentran las estrategias de pruebas como las pruebas unitarias, las pruebas de integración, las pruebas del sistema, entre otras. A continuación se exponen las pruebas realizadas a la propuesta de solución.

### 3.9.1. Pruebas funcionales en el diseño de casos de prueba utilizando la técnica de caja negra.

Pruebas funcionales sin acceso al código fuente de las aplicaciones, se trabaja con entradas y salidas. Se refiere a las pruebas que se llevan a cabo sobre la interfaz del *software*. Para la ejecución de estas pruebas se emplean los casos de prueba, los que constituyen un conjunto de entradas con datos de pruebas, unas condiciones de ejecución y unos resultados esperados, con el propósito de identificar las condiciones que se llevarán a cabo en las pruebas. Por lo que los casos de prueba pretenden demostrar que las funciones del *software* son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene (51), (52).

A continuación se muestra el diseño de caso de prueba de la funcionalidad "Crear orden de solicitud de impresión" del componente personalización utilizando la técnica de caja negra, Para consultar el resto de los casos de pruebas, así como su resultado y descripción, consultar Anexo 13.

No	Nombre de Campo	Clasificación	Valor Nulo	Descripción
1	Criterio	Lista desplegable	Sí	Este campo contiene varios criterios como: Tipo, Nombres y apellidos, Carnet de Identidad.
2	Campo de filtro	Campo de texto	Sí	Este campo admite solamente letras si la selección en la lista de criterios fue: Tipo, Nombre y Apellidos.
3	Marcar	Campo de selección	Sí	Este campo puede o no estar seleccionado.

Tabla 8. Descripción de las variables del caso de prueba crear orden de solicitud de impresión. Fuente: Elaboración propia.



## Capítulo III. Implementación y pruebas.

Escenario	Descripción	Variable1	Variable2	Variable3	Respuesta del sistema.	Flujo central
EC 1.1 Filtrar	El usuario selecciona un criterio y luego introduce los datos a filtrar y luego oprime el botón filtrar.	V (Tipo)	V(solapín)	NA	El sistema muestra todos los trámites que correspondan a ese filtro.	1. Crear órdenes 2. Seleccionar criterios. 3. Introducir texto. 4. Filtrar.
		V (Tipo)	I(Campo vacíos)	NA	El sistema no muestra ningún trámite.	1. Crear órdenes 2. Seleccionar criterios. 3. Introducir texto. 4. Filtrar.
		I(Valor por defecto: Selecciona)	V(Cualquier texto)	NA	El sistema no muestra ningún trámite porque no se ha seleccionado ningún criterio.	1. Crear órdenes 2. Seleccionar criterios. 3. Introducir texto. 4. Filtrar.
		I(Valor por defecto: Selecciona)	I(Campo vacíos)	NA	El sistema no muestra ningún trámite porque no se ha seleccionado ningún criterio.	1. Crear órdenes 2. Seleccionar criterios. 3. Introducir texto. 4. Filtrar.

Tabla 9. Diseño del caso de prueba "Crear orden de solicitud de impresión". Escenario "Filtrar". Fuente:

Elaboración propia.

Escenario	Descripción	Variable1	Variable2	Variable3	Respuesta del sistema.	Flujo central
-----------	-------------	-----------	-----------	-----------	------------------------	---------------

EC 1.1 Crear orden	El usuario selecciona el o los trámites y luego oprime el botón Crear.	NA	NA	V(Cuando se selecciona uno o varios elementos)	El sistema muestra un mensaje de Orden Creada.	1. Crear órdenes 2. Seleccionar Criterios. 3. Introducir texto. 4. Filtrar. 5. Marcar. 6. Crear.
		NA	NA	I (Cuando no se selecciona ningún elemento)	El sistema muestra un mensaje de Error al crear la orden.	1. Crear órdenes. 2. Seleccionar criterios. 3. Introducir texto. 4. Filtrar. 5. Marcar. 6. Crear

Tabla 10. Diseño del Caso de Prueba "Crear orden de solicitud de impresión". Escenario "Crear orden". Fuente: Elaboración propia

### 3.9.2. Resultados de las pruebas.

Las siguientes tablas muestran los resultados que arrojaron las pruebas realizadas a los componentes de personalización y de entrega. Estas tablas representan las no conformidades, las cuales indican los problemas detectados y las iteraciones representan la cantidad de veces que fue probada la aplicación, para comprobar los resultados satisfactorios en cada una de las funcionalidades implementadas.

#### Componente de personalización.

Iteraciones	Funcionalidades	No conformidades
1	RFG10, RFG11, RFP1, RFP2	7
2	RFG10, RFG11, RFP1, RFP2	3
3	RFG10, RFG11, RFP1, RFP2	6
4	RFG10, RFG11, RFP1, RFP2	0
5	RFP3, RFP4	3
6	RFP3, RFP4	1
7	RFP3, RFP4	0

Tabla 11. Resultados de las pruebas del componente de personalización. Fuente: Elaboración propia.

### 3.9.2.1. Descripción de los resultados de las pruebas para el componente de personalización.

Los resultados arrojados para el componente de personalización, estuvieron definidos por siete iteraciones de pruebas a los diferentes requisitos, donde en la primera iteración para los cuatro requisitos probados, se detectaron siete no conformidades, para la segunda tres, para la tercera se detectaron seis y para la cuarta cero. De igual manera, para los tres requisitos restantes en una quinta iteración se detectaron 3 no conformidades que fueron erradicadas luego de dos iteraciones más.

### 3.10. Conclusiones parciales.

Con la utilización de los diferentes estándares de codificación y la aplicación de las buenas prácticas de programación se logró la implementación de un código con buen rendimiento y apropiado, lo que permite que el *software* desarrollado, tenga la calidad adecuada. Además permitió que el código reflejase un solo estilo, permitiendo el entendimiento del mismo si se decidiese reutilizar. También con el desarrollo del tratamiento de errores, en este caso las excepciones y el empleo de medidas de seguridad durante la implementación, se logró realizar una aplicación capaz de tolerar fallas, pudiéndose ejecutar la solución sin verse afectada por estos inconvenientes y robusta en cuanto a seguridad. Además con la definición de los componentes que intervienen en la solución, así como la relación de los nodos físicos que los componen, se logró el desarrollo de una primera versión de la solución. También se le realizaron las pruebas a la aplicación basándose en los diseños de casos de prueba definidos, donde se muestra el progreso por iteraciones de la solución, logrando la calidad, de manera que el *software* cumpla con las especificaciones del cliente.

### Conclusiones generales

Con el desarrollo del presente trabajo de diploma se ha logrado dar cumplimiento a los objetivos trazados, proporcionando una mejor propuesta de solución a los actuales procesos encargados de la emisión de los documentos de identificación de la UCI. A continuación se citan los principales logros de la investigación:

La solución desarrollada constituye una aplicación informática del Centro de Identificación y Seguridad Digital (CISED) que aporta un conjunto de conocimientos técnicos, productos de la información y la experiencia adquirida, que contribuye al éxito comercial del CISED en cuanto a productos de emisión de documentos de identificación.

El desarrollo de los componentes personalización, control de calidad y entrega utilizando al SPDI permite la creación de un documento de identificación que cumpla con los estándares internacionales en cuanto a seguridad y composición física.

La solución informática desarrollada, permite la personalización, el control de la calidad y la entrega de un documento de identificación con elementos de seguridad para la UCI.

## Recomendaciones

Se recomienda a los interesados en mejorar o continuar esta investigación:

- Agregar la implementación de más medidas de seguridad tales como: código QR.
- Desarrollar un servicio de *Windows* que actualice los estados de las órdenes en la base de datos del PMICA cada cierto tiempo.

## Bibliografía

1. *Los documentos de viaje en la lucha contra el terrorismo*. **Ramírez Mastrapa, Yudenia**. La Habana : UCIENCIA, 2010.
2. **CODERCO**. CODERCO. [En línea] [Citado el: 8 de Enero de 2013.] [http://www.coderco.com/pag\\_estandares.htm](http://www.coderco.com/pag_estandares.htm).
3. **Gutierrez, Jaime y Tena, Juan**. *Protocolos Criptográficos y Seguridad en Redes*. s.l. : Servicios de Publicaciones de la Universidad de Cantabria, 2003. ISBN 84-8102-345-0.
4. **Access Control China**. Access Control China. [En línea] 2004-2011. [Citado el: 9 de Enero de 2013.] [http://www.accesscontrolchina.com/Technical\\_ISO\\_IEC\\_14443\\_7810\\_15693.html](http://www.accesscontrolchina.com/Technical_ISO_IEC_14443_7810_15693.html).
5. **Organización de Aviación Civil Internacional**. *Documentos de viaje de lectura mecánica.Documentos de viaje oficiales de lectura mecánica.Volumen1.dvLM con datos de lectura mecánica almacenados en formato de reconocimiento óptico de caracteres*. . Quebec,Canadá : s.n., 2008. ISBN 978-92-9231-339-5.
6. **Bundes Druckerei**. Bundes Druckerei. *Bundes Druckerei*. [En línea] [Citado el: 11 de Enero de 2013.] [http://www.bundesdruckerei.de/..](http://www.bundesdruckerei.de/)
7. **Oberthur Technologies**. Oberthur Technologies . *Oberthur Technologies* . [En línea] [Citado el: 13 de Enero de 2013.] [http://www.oberthur.com/..](http://www.oberthur.com/)
8. **Muhlbauer**. *Muhlbauer High Tech International*. [Informe] 2008.
9. **Thomas Greg & Sons**. Thomas Greg & Sons. . *Thomas Greg & Sons*. [En línea] [Citado el: 13 de Enero de 2013.] <http://www.thomasgreg.com/index.php/es>.
10. *Modernización del proceso de tramitación de carnet de extranjeros de la Dirección de Inmigración y Extranjería de la República de Cuba* . . **Ballate Oquendo, José Antonio, Salinas Peña, Onel y Pérez Bandomo, Dariel Ricardo**. 8, La Habana : Serie Científica de la Universidad de las Ciencias Informáticas, 2012, Vol. 5. ISSN: | RNPS.
11. **DATYS**. DATYS . *DATYS.Tecnologías y Sistemas*. . [En línea] [Citado el: 13 de Enero de 2013.] [www.datys.cu](http://www.datys.cu)..
12. **CISED**. CISED. Centro de Identificación y Seguridad Digital. *CISED*. [En línea] [Citado el: 14 de Enero de 2013.] [portal.proiden.uci.cu](http://portal.proiden.uci.cu)..
13. **SEDICI**. SEDICI.Repositorio Institucional de la Universidad de la Plata. *SEDICI*. [En línea] 2003-2013. [Citado el: 14 de Enero de 2013.] [http://sedici.unlp.edu.ar/bitstream/handle/10915/4055/3\\_-\\_Ingenier%C3%ADa\\_de\\_software.pdf?sequence=7](http://sedici.unlp.edu.ar/bitstream/handle/10915/4055/3_-_Ingenier%C3%ADa_de_software.pdf?sequence=7).
14. **Molpeceres, Alberto**;. *Procesos de desarrollo:RUP,XP y FDD*. Alemania : Licencia de documentación de Java Hispano v1.0, 2003.
15. **Buenas Tareas**. Buenas Tareas. *Buenas Tareas*. [En línea] 2013. [Citado el: 15 de Enero de 2013.] <http://www.buenastareas.com/ensayos/Metodologia-Fdd/551499.html>.
16. **Instituto Nacional de Estadística e Informática**. *Herramientas CASE*. s.l. : Oficina Técnica de Difusión Estadística y Tecnología informática del Instituto Nacional de Estadística e Informática(INEI), Noviembre 1999. 875-99-OI-OTDETI-INEI.
17. **monografias.com**. monografias.com. *monografias.com*. [En línea] [Citado el: 18 de Enero de 2013.] <http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>.
18. **Free Download Manager**. Free Download Manager. *Free Download Manager*. [En línea] [Citado el: 19 de Enero de 2013.] [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%28M%C3%8D%29\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/).

19. **Kioskea.net**. Kioskea.net. *Kioskea.net*. [En línea] [Citado el: 21 de Enero de 2013.] <http://es.kioskea.net/contents/304-lenguajes-de-programacion>.
20. **Scribd**. Scribd. *Scribd*. [En línea] [Citado el: 22 de Enero de 2013.] <http://es.scribd.com/doc/7411856/Caracteristicas-de-C..>
21. **Curso C++**. Curso C++. *Curso C++*. [En línea] [Citado el: 19 de Enero de 2013.] [www.zator.com/Cpp/E1\\_2.htm](http://www.zator.com/Cpp/E1_2.htm).
22. **Altova**. Altova. *Altova*. [En línea] 2005-2013. [Citado el: 20 de Enero de 2013.] <http://www.altova.com/es/umodel/business-process-modeling.html>.
23. **Pilar Vasquez, María, Giandini, Roxana y Bazán, Patricia**. *Análisis comparativo de Lenguajes Notacionales para Modelado de Procesos*. s.l. : Universidad Nacional de la Plata .
24. **BPMN**. *BPMN Business Process Modeling Notation*. Bogotá, Colombia : s.n.
25. **Sanders, William B**. *A Beginners Guide, ASP.NET 3.5*.
26. **CarlosBlanco.pro**. Entornos de Desarrollo Integrado. *Entornos de Desarrollo Integrado*. [En línea] [Citado el: 22 de Enero de 2013.] <http://carlosblanco.pro/2012/04/entornos-desarrollo-integrado-introduccion/>.
27. **Netbeans Accesible.Blog de desarrollo de contenido** . Netbeans Accesible.Blog de desarrollo de contenido . *Netbeans Accesible.Blog de desarrollo de contenido* . [En línea] [Citado el: 23 de Enero de 2013.] <http://netbeansaccesible.blogspot.com/2012/07/caracteristicas-la-plataforma-netbeans.html>.
28. **Intercambios virtuales**. Intercambios virtuales . *Intercambios virtuales*. [En línea] <http://www.intercambiosvirtuales.org/software/microsoft-visual-studio-2010-ultimate-espanol>.
29. **msdn**. msdn. *msdn.Visual Studio*. [En línea] [Citado el: 18 de Enero de 2013.] <http://msdn.microsoft.com/es-es/library/vstudio/ff636699%28v=vs.100%29.aspx>.
30. **Slideshare**. Sistema-Gestin-de-Bases-de-Datos-2657624. [En línea] 2008. [Citado el: 19 de Enero de 2013.] <http://www.slideshare.net/jeiisonlarry/sistema-gestin-de-bases-de-datos-2657624>.
31. **Hernando Velasco, Roberto;** www2.rhernando.net. *www2.rhernando.net*. [En línea] [Citado el: 20 de Enero de 2013.] <http://www2.rhernando.net/modules/tutorials/doc/bd/oracle.html>.
32. **PostgreSQL-es**. PostgreSQL-es. *PostgreSQL-es.Porta en español sobre PostgreSQL*. [En línea] [Citado el: 20 de Enero de 2013.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
33. **Chinchilla Rivera, José Miguel**. *NHibernate*. [Medio Electrónico] s.l. : Universidad de El Salvador, 2012.
34. **monografias.com**. monografias.com. *monografias.com*. [En línea] [Citado el: 22 de Enero de 2013.] <http://www.monografias.com/trabajos75/servidores-web/servidores-web.shtml>.
35. **Observatorio Tecnológico**. Observatorio Tecnológico. *Observatorio Tecnológico*. [En línea] [Citado el: 23 de Enero de 2013.] <http://recursostic.educacion.es/observatorio/web/es/software/servidores/580-elvira-mifsud>.
36. **Ciberaula**. Ciberaula. *Ciberaula*. [En línea] 2010. [Citado el: 23 de Enero de 2013.] [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro](http://linux.ciberaula.com/articulo/linux_apache_intro).
37. **Scribd**. Scribd. *Scribd*. [En línea] [Citado el: 19 de Enero de 2013.] <http://es.scribd.com/doc/22300726/Especificaciones-Tecnicas-IIS-vs-Apache>.
38. —. Scribd. *Scribd*. [En línea] [Citado el: 20 de Enero de 2013.] <http://es.scribd.com/doc/46334453/Tortoise-SVN>.

39. **Ayala, Beatriz, Ramírez, Claudia Marcela and Ocampo, Lina María.** *La ingeniería de requerimientos aplicada al desarrollo de sistema de la información.*
40. **Gamma, Erich, y otros.** *Patrones de Diseño. Elementos de software orientado a objetos reutilizable.* . Madrid : Pearson Educación, 2003. ISBN:84-7829-059-1.
41. **Entorno Virtual de Aprendizaje.** . Entorno Virtual de Aprendizaje. . *Entorno Virtual de Aprendizaje.* . [En línea] 2010-2011. [Citado el: 24 de Enero de 2013.] [http://eva.uci.cu/file.php/160/Curso\\_2010-2011/Semana\\_3/Conferencia\\_3/Materiales\\_Basicos/Diseno\\_Web.pdf..](http://eva.uci.cu/file.php/160/Curso_2010-2011/Semana_3/Conferencia_3/Materiales_Basicos/Diseno_Web.pdf..)
42. **Aramúndiz, Silvia.** Entorno Virtual de Aprendizaje. *Entorno Virtual de Aprendizaje.* [En línea] 2010-2011. [Citado el: 19 de Enero de 2013.] [http://eva.uci.cu/file.php/159/Curso\\_2010-2011/Clases/Semana\\_11/Laboratorio\\_3/Materiales\\_Complementarios/DIAGRAMA\\_DE..](http://eva.uci.cu/file.php/159/Curso_2010-2011/Clases/Semana_11/Laboratorio_3/Materiales_Complementarios/DIAGRAMA_DE..)
43. **W3C.** W3C. W3C. [En línea] 2013. [Citado el: 10 de Febrero de 2013.] <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb..>
44. **msdn.** MSDN . MSDN . [En línea] [Citado el: 8 de Marzo de 2013.] <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx..>
45. **Algoritmo Digital . .** Algoritmo Digital. . *Algoritmo Digital . .* [En línea] [Citado el: 10 de Marzo de 2013.] [http://jms32.eresmas.net/tacticos/programacion/VBNet/VBN03/alg\\_aspnet/abr04\\_TratamientoErroresASPNET\\_1.htm..](http://jms32.eresmas.net/tacticos/programacion/VBNet/VBN03/alg_aspnet/abr04_TratamientoErroresASPNET_1.htm..)
46. **Organización de la Aviación Civil Internacional.** *Documentos de viaje de lectura mecánica .Parte 3 Volumen1.* Canadá : s.n., 2008. ISBN 978-92-9231-339-5.
47. **Mato, Rosa María.** *Diseño de Base de Datos.* Octubre,1999.
48. **Marca Huallpara, Hugo Michael and Quisbert Limachi, Nancy Susana.** *Análisis y Diseño de Sistemas II.Trabajo de investigación y Exposición.Diagrama de Despliegue.*
49. **MSDN.** MSDN. MSDN. [En línea] [Citado el: 15 de Marzo de 2013.] <http://msdn.microsoft.com/es-es/library/dd409390.aspx..>
50. **Juristo, Natalia, Moreno, Ana M y Vegas, Sira.** *Técnicas de Evaluación de Software.* Versión 12.0. Octubre,2006.
51. **it-Mentor.ITM.** *Capacitación guía para el desarrollo de software.Pruebas de software.* .
52. **Hernández Aguilar, Violena y González Jorrín, Michael.** Entorno Virtual de Aprendizaje. *Entorno Virtual de Aprendizaje.* [En línea] [Citado el: 5 de Abril de 2013.] [http://eva.uci.cu/file.php/158/Documentos/Recursos\\_didacticos/Presentaciones\\_digitales\\_UD\\_2/Proceso\\_de\\_pruebas\\_de\\_caja\\_negra\\_basado\\_en\\_la\\_descripcion\\_de\\_casos\\_de\\_uso.pdf..](http://eva.uci.cu/file.php/158/Documentos/Recursos_didacticos/Presentaciones_digitales_UD_2/Proceso_de_pruebas_de_caja_negra_basado_en_la_descripcion_de_casos_de_uso.pdf..)



## Bibliografía consultada.

1. **Access Control China.** Access Control China. [En línea] 2004-2011. [Citado el: 9 de Enero de 2013.] [http://www.accesscontrolchina.com/Technical\\_ISO\\_IEC\\_14443\\_7810\\_15693.html](http://www.accesscontrolchina.com/Technical_ISO_IEC_14443_7810_15693.html).
2. **Organización de Aviación Civil Internacional.** *Documentos de viaje de lectura mecánica.Documentos de viaje oficiales de lectura mecánica.Volumen1.dvLM con datos de lectura mecánica almacenados en formato de reconocimiento óptico de caracteres.* . Quebec,Canadá : s.n., 2008. ISBN 978-92-9231-339-5.
3. **SEDICI.** SEDICI.Repositorio Institucional de la Universidad de la Plata. *SEDICI.* [En línea] 2003-2013. [Citado el: 14 de Enero de 2013.] [http://sedici.unlp.edu.ar/bitstream/handle/10915/4055/3\\_-\\_Ingenier%C3%ADa\\_de\\_software.pdf?sequence=7](http://sedici.unlp.edu.ar/bitstream/handle/10915/4055/3_-_Ingenier%C3%ADa_de_software.pdf?sequence=7).
4. **Molpeceres, Alberto;** *Procesos de desarrollo:RUP,XP y FDD.* Alemania : Licencia de documentación de Java Hispano v1.0, 2003.
5. **Buenas Tareas.** Buenas Tareas. *Buenas Tareas.* [En línea] 2013. [Citado el: 15 de Enero de 2013.] <http://www.buenastareas.com/ensayos/Metodologia-Fdd/551499.html>.
6. **Instituto Nacional de Estadística e Informática.** *Herramientas CASE.* s.l. : Oficina Técnica de Difusión Estadística y Tecnología informática del Instituto Nacional de Estadística e Informática(INEI), Noviembre 1999. 875-99-OI-OTDETI-INEI.
7. **monografias.com.** monografias.com. *monografias.com.* [En línea] [Citado el: 18 de Enero de 2013.] <http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>.
8. **Free Download Manager.** Free Download Manager. *Free Download Manager.* [En línea] [Citado el: 19 de Enero de 2013.] [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%28M%C3%8D%29\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/).
9. **Altova.** Altova. *Altova.* [En línea] 2005-2013. [Citado el: 20 de Enero de 2013.] <http://www.altova.com/es/umodel/business-process-modeling.html>.
10. **Pilar Vasquez, María, Giardini, Roxana y Bazán, Patricia.** *Análisis comparativo de Lenguajes Notacionales para Modelado de Procesos.* s.l. : Universidad Nacional de la Plata .
11. **BPMN.** *BPMN Business Process Modeling Notation.* Bogotá,Colombia : s.n.
12. **Kioskea.net.** Kioskea.net. *Kioskea.net.* [En línea] [Citado el: 21 de Enero de 2013.] <http://es.kioskea.net/contents/304-lenguajes-de-programacion>.
13. **Scribd.** Scribd. *Scribd.* [En línea] [Citado el: 22 de Enero de 2013.] <http://es.scribd.com/doc/7411856/Caracteristicas-de-C..>
14. **Organización de la Aviación Civil Internacional.** *Documentos de viaje de lectura mecánica .Parte 3 Volumen1.* Canadá : s.n., 2008. ISBN 978-92-9231-339-5.
15. **Mato, Rosa María.** *Diseño de Base de Datos.* Octubre,1999.
16. **Marca Huallpara, Hugo Michael and Quisbert Limachi, Nancy Susana.** *Análisis y Diseño de Sistemas II.Trabajo de investigación y Exposición.Diagrama de Despliegue.*
17. **MSDN.** MSDN. *MSDN.* [En línea] [Citado el: 15 de Marzo de 2013.] <http://msdn.microsoft.com/es-es/library/dd409390.aspx..>
18. **Juristo, Natalia, Moreno, Ana M y Vegas, Sira.** *Técnicas de Evaluación de Software.* Versión 12.0. Octubre,2006.

19. **it-Mentor.ITM.** *Capacitación guía para el desarrollo de software.Pruebas de software.* .
20. **Hernández Aguilar, Violena y González Jorrín, Michael.** Entorno Virtual de Aprendizaje. *Entorno Virtual de Aprendizaje.* [En línea] [Citado el: 5 de Abril de 2013.]  
[http://eva.uci.cu/file.php/158/Documentos/Recursos\\_didacticos/Presentaciones\\_digitales\\_UD\\_2/Proceso\\_de\\_pruebas\\_de\\_caja\\_negra\\_basado\\_en\\_la\\_descripcion\\_de\\_casos\\_de\\_uso.pdf](http://eva.uci.cu/file.php/158/Documentos/Recursos_didacticos/Presentaciones_digitales_UD_2/Proceso_de_pruebas_de_caja_negra_basado_en_la_descripcion_de_casos_de_uso.pdf).
21. **Aviles Membribes, Yamilka.** *Sistema Automatizado para la Emisión de Pasaporte Andino de Venezuela.* . La Habana : s.n., 2006.
22. **Castro Milán, Ronaldo.** “*Propuesta de procesos para el desarrollo de un Sistema de Emisión de Documentos de Identificación*”. . La Habana : s.n., 2009.
23. **de la Torre Luis, Maikel y Proenza Labadie, Diovis.** *Aplicación de Gestión de Interfaces de Usuario para el Sistema de Emisión de Documentos de Identificación del Centro de Identificación y Seguridad Digital.* . La Habana : s.n., Junio,2009.
24. **Crespo Cabezas, Jenny y Terreno Galano, Roberkis.** *Sistema Automatizado de Gestión para el Centro de Personalización e Impresión de Documentos del SAIME.* La Habana : s.n., 2007.
25. **Ombelli, Diana y Knopjes, Fons.** *Documentos:Manual para Desarrolladores.* Suiza,Ginebra : Organización Internacional para las Migraciones, abril,2008. ISBN 978-92-9068-609-5.
26. **Organización de la Aviación Civil Internacional.** *Doc 9303 Parte 1 Volumen 1.Pasaportes con datos de lectura mecánica almacenados en formato de reconocimiento óptico de caracteres.* . 2006.
27. **Galens Ameneiro, Linnet y Sierra Obregón, Dannier.** *Sistema de Personalización de Documentos de Identificación de la República de Cuba.* . La Habana : s.n., 2010.
28. **Valdés Fernández, Yanisleivi y Valdés Cabrera, Yandy.** *Manual de Usuario del Sistema de Personalización de Documentos de Identificación(SPDI).* . La Habana : s.n., Diciembre2011.
29. **White, Stephen A.** *BPMN.Guía de referencia y modelado.* s.l. : Future Strategies, 2009. ISBN13:978-0-9819870-3-3.

## Glosario de términos.

**Calidad:** Propiedad o conjunto de propiedades inherentes a una cosa que permiten caracterizarla y valorarla como igual, mejor o peor que las restantes de su especie.

**Comprobante:** Recibo o documento que confirma una gestión. Documento que certifica una acción.

**Control de calidad:** Proceso que permite chequear que los documentos personalizados cumplan con la calidad requerida.

**Documento de Identificación:** Documento único de identificación personal e intransferible, que contiene datos personales de carácter público emitido por una autoridad administrativa competente para permitir la identificación personal de los ciudadanos. Usualmente conocidos como Cédula de Ciudadanía (CC), Carné de Identidad (CI), Cédula de Identidad (CI) o Documento Nacional de Identidad (DNI).

**Entrega:** Proceso mediante el cual una persona recibe el documento de identificación.

**Identificación:** Reconocimiento de la identidad de alguien. Documento de identidad.

**Imagen:** Reproducción de la figura de una cosa, ya sea una firma, una huella, una fotografía, que es captada por un aparato óptico, o de fotografía, o de otro tipo y se almacenan en un sistema.

**Oficina:** Lugar físico donde se lleva a cabo el proceso de emisión de documentos de identificación (solicitud personal, captura de datos, captura de imágenes, revisión de datos e imágenes, impresión de documentos de identificación, control de la calidad y entrega de documentos). En estas áreas los procesos son llevados a cabo por funcionarios con diferentes roles.

**Personalización:** Proceso para obtener documentos personales utilizables a partir de documentos vírgenes. Esta etapa es esencial para todo documento en cuestiones como seguridad, calidad y la coherencia para garantizar y mantener en forma todos los documentos.

**Proceso:** Conjunto de actuaciones, decisiones, actividades y tareas que se encadenan de forma secuencial y ordenada para conseguir un resultado que satisfaga los requerimientos del cliente al que va dirigido.

**Rol:** Categoría que se le asigna a uno o varios usuarios para permitir el acceso a determinadas funcionalidades del sistema. Este es asignado o revocado a través de la administración.

**Solicitud:** Documento formal en el que se pide una cosa.

**Trámite:** Pasos que se realizan de manera sucesiva para solucionar un asunto. Cada uno de los estados o diligencias necesarios para resolver un asunto.

## **Anexos.**

### **Anexo 1. Descripción del flujo básico del proceso personalización.**

**Solicita crear orden de impresión:** El responsable de personalización solicita crear orden de impresión a través de la opción correspondiente.

Responsable: Responsable de personalización.

Entradas: Trámites listos para comenzar la personalización.

Salidas: Iniciado proceso de personalización.

**Crea órdenes de impresión:** El sistema crea órdenes de impresión formada por un conjunto de solicitudes de impresión.

Responsable: Sistema.

Entradas: Solicitudes de impresión.

Salidas: Órdenes de impresión.

**Envía órdenes de impresión:** El sistema envía las órdenes de impresión al SPDI.

Responsable: Sistema.

Entradas: Órdenes de impresión con el conjunto de solicitudes que la conforman.

Salidas: Órdenes de impresión enviada.

**Recibe órdenes de impresión:** El SPDI imprime los datos e imágenes en el documento.

Responsable: SPDI.

Entradas: Órdenes de impresión.

Salidas: Órdenes de impresión recibida.

**Imprimir órdenes:** El SPDI imprime cada una de las solicitudes que componen la(s) orden(es) de impresión.

Responsable: SPDI.

Entradas: Orden de impresión, Documento virgen.

Salidas: Documento personalizado.

**Comprobar calidad:** El SPDI comprueba la calidad de los documentos personalizados pertenecientes a una orden de impresión.

Responsable: SPDI.

Entradas: Documento personalizado.

Salidas: Documento con calidad.

### **Descripción de los flujos paralelos.**

**Realiza solicitud de lotes:** El SPDI realiza solicitud de lotes de documentos vírgenes.

Responsable: SPDI.

Entradas: -.

Salidas: Solicitud de lotes de documentos vírgenes.

**Recibe solicitud de lotes:** El sistema recibe solicitud de lotes de documentos vírgenes por parte del SPDI.

Responsable: Sistema.

Entradas: Solicitud de lotes de documentos vírgenes.

Salidas: Solicitud de lotes de documentos vírgenes recibida.

**Crea lotes:** El sistema crea lotes de documentos vírgenes.

Responsable: Sistema.

Entradas: Solicitud de lotes de documentos vírgenes.

Salidas: Lotes de documentos vírgenes creado.

**Envía lotes:** El sistema envía lotes de documentos vírgenes al SPDI.

Responsable: Sistema.

Entradas: Lotes de documentos vírgenes creado.

Salidas: Lotes de documentos vírgenes enviados.

**Recibe lotes:** El SPDI recibe lotes de documentos vírgenes.

Responsable: SPDI.

Entradas: Lotes de documentos vírgenes enviado.

Salidas: Lotes de documentos vírgenes recibido.

**Realiza supervisión:** El SPDI realiza supervisión cuando la orden es reimpressa por problemas de calidad.

Responsable: SPDI.

Entradas: Orden de impresión reimpressa.

Salidas: Supervisión realizada.

**Comprobar calidad:** El SPDI realiza una comprobación de calidad una vez que la orden es reimpresa por problemas de calidad y el paso de supervisión indica que tiene calidad.

Responsable: SPDI.

Entradas: Orden de impresión reimpresa y supervisada con calidad.

Salidas: Calidad comprobada.

**Realiza supervisión:** El SPDI realiza supervisión cuando la orden tiene problemas de calidad detectados en la actividad comprobar calidad.

Responsable: SPDI.

Entradas: Orden de impresión sin calidad.

Salidas: Supervisión realizada.

**Comprobar calidad:** El SPDI comprueba la calidad una vez realizada la supervisión y el documento tenga la calidad requerida.

Responsable: SPDI.

Entradas: Orden de impresión con calidad.

Salidas: Comprobación de calidad realizada.

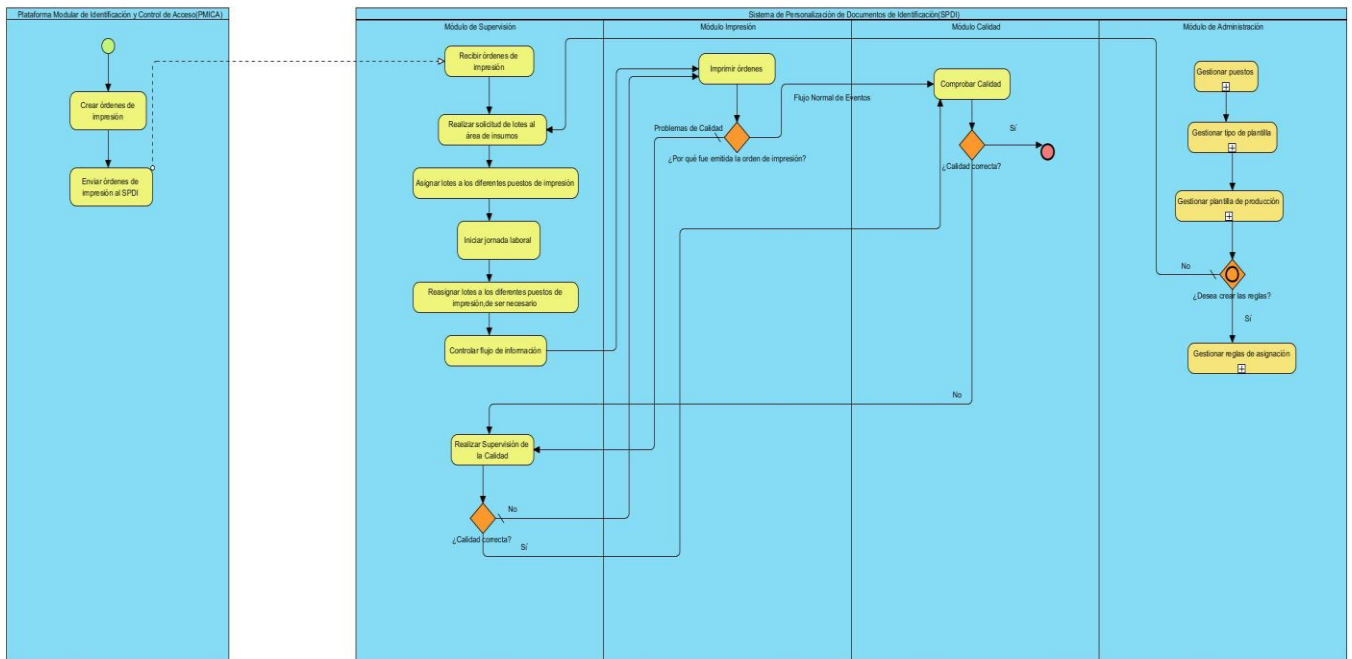
**Imprimir órdenes:** El SPDI reimprime cada una de las solicitudes que componen la(s) orden(es) de impresión.

Responsable: SPDI.

Entradas: Orden de impresión, Documento virgen.

Salidas: Documento reimpreso.

**Anexo 2. Diagrama de personalización.**



**Figura 13. Diagrama de personalización. Fuente: Elaboración propia con ayuda del manual de usuario del SPDI.**

### Anexo 3. Descripción del flujo básico del proceso de entrega.

**Solicita documento:** El usuario se presenta en la oficina y solicita la entrega de un documento de identificación.

Responsable: Usuario.

Entradas: -

Salidas: Solicitud de entrega de un documento de identificación.

**Recibe solicitud:** El responsable de entrega recibe la solicitud de entrega de documento.

Responsable: Responsable de entrega.

Entradas: Solicitud de entrega de un documento de identificación.

Salidas: Solicitud de entrega recibida.

**Busca trámite:** El sistema busca el trámite de la persona que solicita la entrega del documento.

Responsable: Sistema.

Entradas: -

Salidas: Si se encuentra o no el trámite listo para entregar.

**Muestra datos del trámite:** El sistema muestra los datos relativos al trámite.

Responsable: Sistema.

Entradas: Trámite seleccionado.

Salidas: Datos del trámite.

**Verifica identidad:** El responsable de entrega verifica la identidad de la persona.

Responsable: Responsable de entrega.

Entradas: Datos del trámite.

Salidas: Identidad verificada.

**Captura datos:** El responsable de entrega coloca el documento en el dispositivo para capturar los datos.

Responsable: Sistema.

Entradas: Documento de identificación.

Salidas: Documento colocado en el dispositivo.

**Realiza lectura de los datos:** El sistema realiza la lectura del documento y muestra los datos capturados.

Responsable: Sistema.

Entradas: Documento colocado en el dispositivo.

Salidas: Datos leídos.

**Compara datos:** El sistema realiza una comparación entre los datos impresos en el documento y los datos capturados.

Responsable: Sistema.

Entradas: Datos leídos.

Salidas: Datos comparados.

**Actualizar el estado del documento a “Entregado”:** Si no existen dispositivos para verificar la identidad, el sistema actualiza el estado del documento a “Entregado”.

Responsable: Sistema.

Entradas: -.

Salidas: Estado del documento actualizado a “Entregado”.

**Actualizar el estado del trámite a “Efectivo”:** Si el documento tiene el estado “Entregado”, el sistema actualiza el estado de la fase del trámite a “Efectivo”.



Responsable: Sistema.

Entradas: Estado del documento "Entregado".

Salidas: Estado de la fase del trámite actualizado a "Efectivo".

**Notifica cierre de trámite:** El responsable de entrega notifica al usuario que el trámite se ha cerrado.

Responsable: Responsable de entrega.

Entradas: Estado del documento "Entregado", estado de la fase del trámite "Efectivo".

Salidas: Cierre de trámite.

**Entrega del documento:** El responsable de entrega le da al usuario el documento de identificación.

Responsable: Responsable de entrega.

Entradas: Cierre de trámite.

Salidas: Documento de identificación.

**Recibe documento:** El usuario recibe por parte del responsable de entrega el documento de identificación.

Responsable: Responsable de entrega.

Entradas: Documento de identificación.

Salidas: Fin del proceso.

### **Descripción de los flujos paralelos.**

**Recibe notificación:** El responsable de entrega recibe la notificación de que el trámite no se encuentra listo para entregar.

Responsable: Responsable de entrega.

Entradas: Listado de trámites listos para entregar.

Salidas: Notificación.

**Recibe resultado:** El usuario recibe la notificación de que el trámite no se encuentra listo para entregar.

Responsable: Usuario.

Entradas: Notificación.

Salidas: Termina el proceso.

**Actualizar el estado del documento a "Entregado":** Si no existen dispositivos para verificar la identidad, el sistema actualiza el estado del documento a "Entregado".

Responsable: Sistema.

Entradas: -.

Salidas: Estado del documento actualizado a "Entregado".

**Actualizar el estado del trámite a "Efectivo":** Si el documento tiene el estado "Entregado", el sistema actualiza el estado de la fase del trámite a "Efectivo".

Responsable: Sistema.

Entradas: Estado del documento "Entregado".

Salidas: Estado de la fase del trámite actualizado a "Efectivo".

**Notifica cierre de trámite:** El responsable de entrega notifica al usuario que el trámite se ha cerrado.

Responsable: Responsable de entrega.

Entradas: Estado del documento "Entregado", estado de la fase del trámite "Efectivo".

Salidas: Cierre de trámite.

**Entrega del documento:** El responsable de entrega le da al usuario el documento de identificación.

Responsable: Responsable de entrega.

Entradas: Cierre de trámite.

Salidas: Documento de identificación.

**Recibe documento:** El usuario recibe por parte del responsable de entrega el documento de identificación.

Responsable: Responsable de entrega.

Entradas: Documento de identificación.

Salidas: Fin del proceso.

**Recibe notificación:** El responsable de entrega recibe la notificación de que la identidad de la persona no coincide con lo registrado en el sistema.

Responsable: Responsable de entrega.

Entradas: Comparación de datos.

Salidas: Datos incorrectos.

**Recibe resultado:** El usuario recibe la notificación de que no se entregará el documento ya que los datos comparados son incorrectos.

Responsable: Usuario.

Entradas: Notificación.

Salidas: Termina el proceso.

#### **Anexo 4.Requisitos generales de los componentes de personalización y entrega.**

##### **Requisitos generales.**

**RFG1.** Buscar trámite.

- Mostrar criterios para buscar el trámite.
- Mostrar los datos del trámite.

**RFG2.** Listar trámites.

**RFG3.** Buscar información de la persona.

- Mostrar criterios para buscar información de la persona.

**RFG4.**Listar personas.

**RFG5.** Mostrar información del ciudadano.

**RFG6.** Actualizar el estado del documento.

**RFG7.** Actualizar estado de las fases del trámite.

**RFG8.** Mostrar datos del documento de identificación.

**RFG9.** Buscar solicitudes.

- Mostrar criterios para buscar las solicitudes de impresión.
- Mostrar los datos de las solicitudes de impresión.

**RFG10.**Listar solicitudes.

**RFG11.** Buscar órdenes.

- Mostrar criterios para buscar las órdenes.
- Mostrar los datos de las órdenes.

**RFG12.**Listar órdenes.

#### **Anexo 5.Requisitos no funcionales.**

- **Tipo de usuario final:** Las características del usuario final se describen en la siguiente tabla:

No.	Sexo	Edad	Nivel de escolaridad	Ocupación	Experiencia profesional	Experiencia con la aplicación informática.	Tipo de discapacidad	Otras
1	Ambos	De 16 a 65 años	Técnico medio, Preuniversitario o Universitario	Cualquiera	Más de 1 año	Más de 1 año	Ninguna	

Tabla 12. Tipo de usuario final. Fuente: Tomado del catálogo de requisitos no funcionales: proyecto <Plataforma Modular de Identificación y Control de Acceso>.

- **Tipo de aplicación informática:** Aplicación *web*.
- **Finalidad:** Gestionar los procesos de identificación y control de acceso de una entidad.
- **Ambiente:**

**Estaciones clientes:**

- **Hardware:** PC Pentium 4 a 1 GHz o superior, mínimo 512 MB de RAM, 40 GB o superior de disco duro.
- **Software:** Sistema operativo *Windows XP SP3* o superior con antivirus actualizado, navegador *web* Mozilla Firefox v17.0 o superior, *Microsoft .Net framework v4.0*. Además del *software* necesario para cada uno de los dispositivos utilizados por el sistema.

**Otras:** Local climatizado de ser posible.

**Estaciones de servidores:**

- **Hardware:** PC Pentium 4 a 2 GHz o superior, mínimo 2 GB de RAM, 250 GB o superior de disco duro.
- **Software:** Sistema operativo *Windows Server 2008 R2* con SP2, antivirus actualizado, *Microsoft .Net framework v4.0*. IIS 7.5 y *PostgreSQL 9.1*.

**Otras:** Climatizado obligatoriamente.

- **Requisitos de usabilidad.**

RNF1. El sistema podrá ser utilizado por cualquier usuario con las siguientes características:

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

RNF2. El sistema será distribuido en idioma español.

RNF3. Los términos utilizados se establecerán acorde al negocio correspondiente para facilitar la comprensión de la herramienta de trabajo.

RNF4. El sistema poseerá estructura y diseño homogéneos en todas sus pantallas, que facilite la navegación.

- Menús laterales y desplegados que permitan el acceso rápido a la información.
- **Requisitos de fiabilidad.**

RNF5. El sistema debe estar disponible las 24 horas del día los 7 días de la semana realizándose copias de seguridad semanalmente tanto de la aplicación como de la base de datos.

RNF6. El tiempo medio entre fallos no debe ser superior a 1 hora.

RNF7. El sistema estará 1 hora fuera de operación luego de haber fallado 3 veces.

- **Requisitos de eficiencia.**

RNF8. El sistema debe ser capaz de soportar una cantidad escalable de peticiones y dar respuestas efectivas y rápidas.

RNF9. El sistema debe hacer un uso eficiente de los recursos de *hardware* donde se encuentre desplegado.

- **Requisitos de soporte.**

RNF11. El sistema dispondrá de un año de soporte técnico donde se corregirán las fallas que atenten contra el buen funcionamiento del sistema.

- **Restricciones de diseño.**

RNF12. Plataforma de desarrollo .NET 4.0 utilizando *Visual Studio Ultimate 2010*.

RNF13. Biblioteca *jQuery*.

RNF16. Para el acceso a datos se utilizará el *ORM NHibernate 3.1.0*.

- **Requisito de interfaz.**

RNF19. El sistema dispondrá de un diseño ameno e intuitivo acorde las características de la institución.

- **Estándares aplicables.**

RNF21. ISO/IEC 7810 Estándar que define las características físicas de las tarjetas de identificación.

## **Anexo 6. Modificaciones realizadas al SPDI.**

- Dentro de WS Order Import en Templates, se creará un XSLT de tipo solapín con todos los atributos para que convierta la entidad solapín en una genérica.

- Dentro de WS Order Import en Template, se creará un XML de tipo solapín con todos los tipos de datos que contiene la nueva credencial.
- Dentro de WS Order Import en Document se creará un nuevo tipo de orden, Solapín\_Order.cs que hereda de Order.
- Dentro de WS Order Import en Document se creará un solapín.Order\_Request que hereda de Order\_Request.
- Dentro de WS Order Import en Tools.cs se creará un case para validar la nueva OrderDocumentType.solapin.
- Dentro de WS Order Import en ImportResponse.cs se creará dentro del enum OrderDocumentType. Un solapín=6 para que sea llamado en Tools.cs.
- Se creará el visual de Crear órdenes de personalización, para que el sistema SPDI pueda comenzar a funcionar.
- Se creará una interfaz donde se inserten los lotes para que el sistema SPDI los reciba y pueda comenzar su funcionamiento.

**Anexo 7. Diagrama de clases del diseño de los componentes de personalización y entrega.**

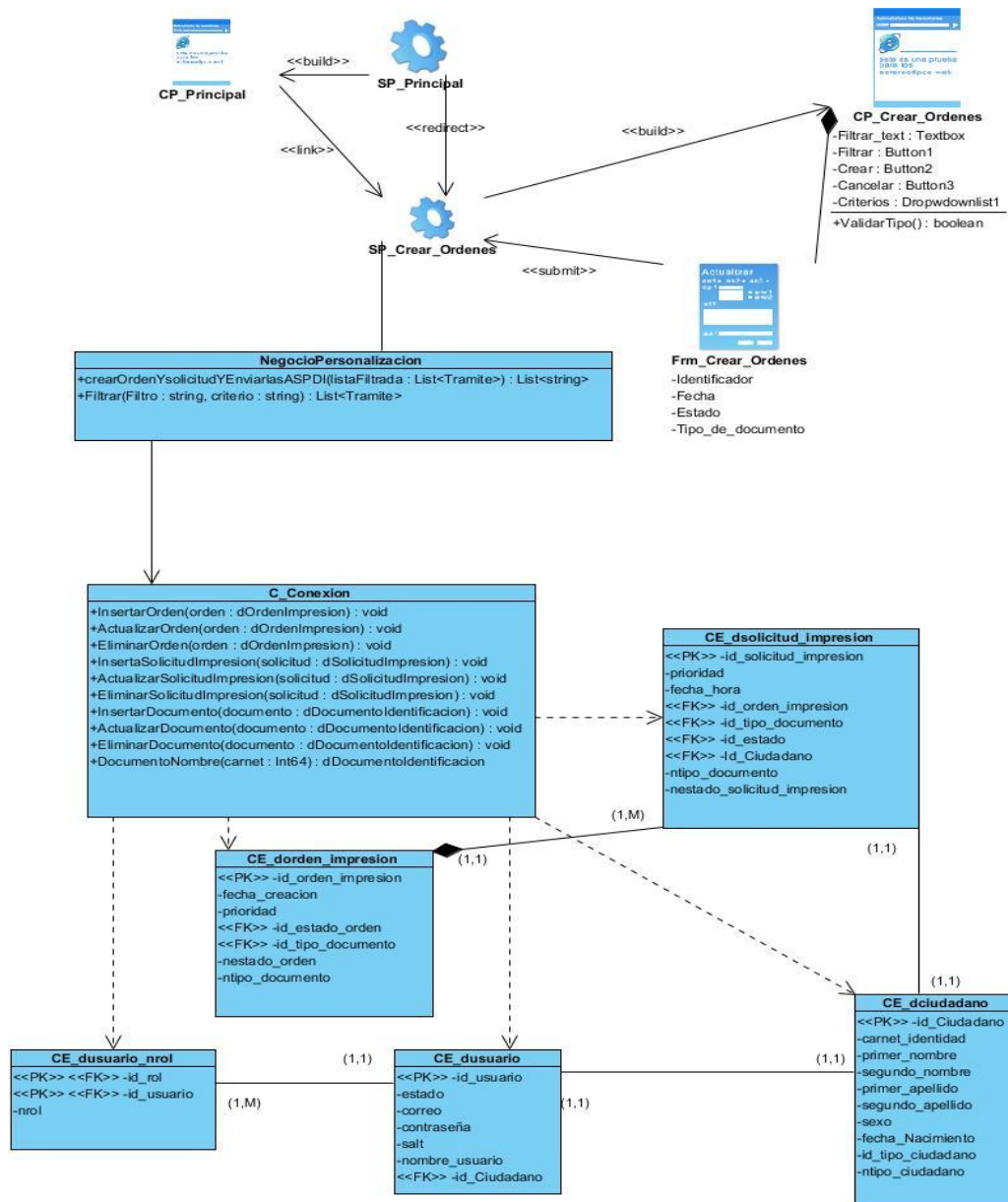


Figura 14. Diagrama de clases del diseño del componente personalización: Crear órdenes de solicitud de impresión. Fuente: Elaboración propia.

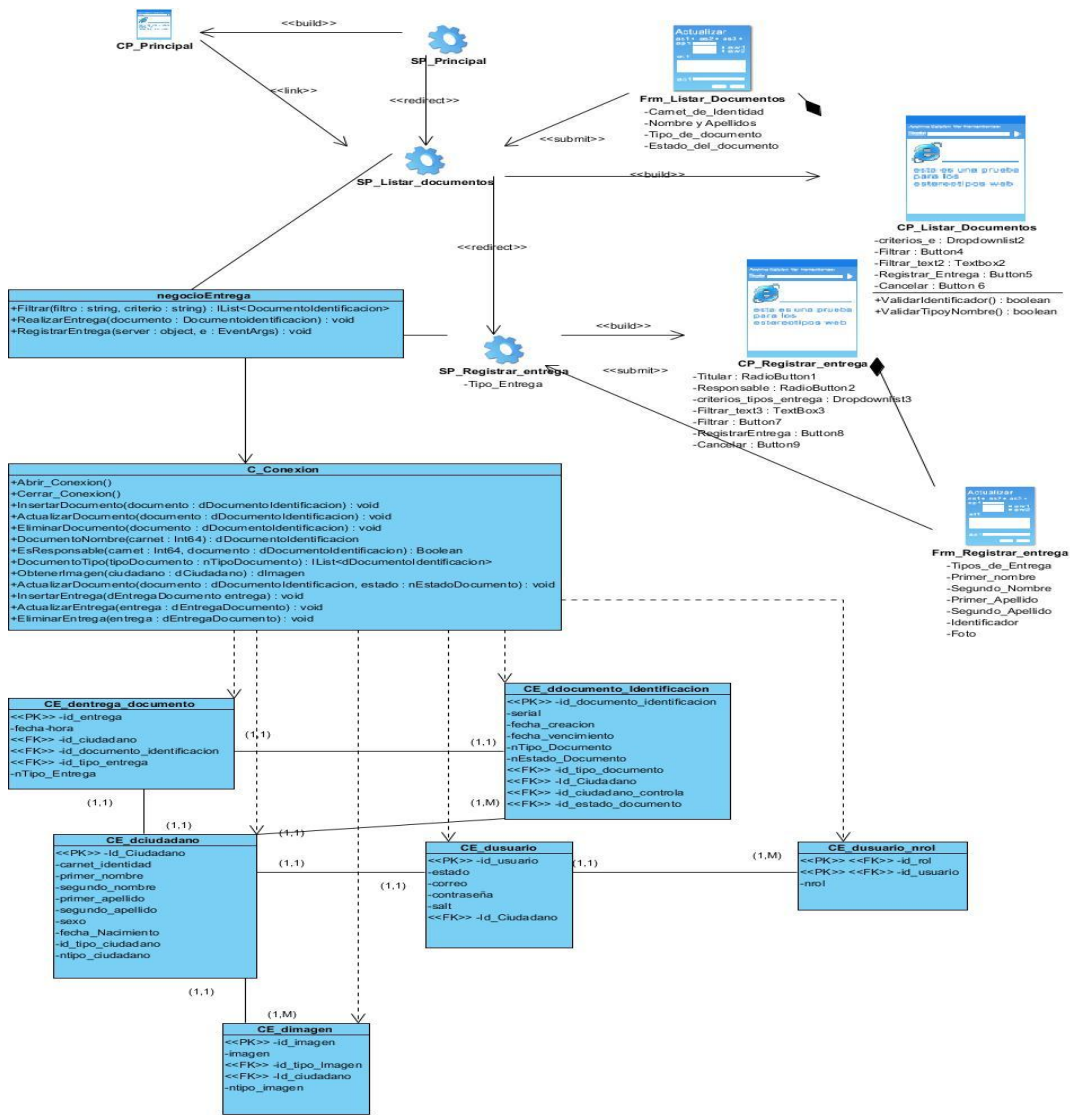


Figura 15. Diagrama de clases del diseño del componente entrega: Registrar entrega. Fuente: Elaboración propia.

### Descripción de las clases controladoras.

Nombre	Negocio_Personalización	
Descripción	Es la clase encargada de gestionar las operaciones con los trámites para la creación y envío de las órdenes.	
Métodos	Parámetros	Definición
List<Trámite> Filtrar	string Filtro, string criterio	Efectúa la búsqueda de un trámite según un criterio especificado.



List<string> crearOrdenYsolicitudYEnviarlasASPDI	List<Trámite> listaFiltrada	Crea y envía la orden al Sistema de Personalización de Documentos de Identificación.
--	--------------------------------	--

Tabla 13. Descripción de los métodos principales de la lógica del negocio del componente de personalización.  
Fuente: Elaboración propia.

### Descripción de las clases conectoras.

<b>Nombre</b>	C_Conexión	
<b>Descripción</b>	Gestiona cada una de las operaciones para ejecutar las consultas en la Base de Datos.	
<b>Métodos</b>	<b>Parámetros</b>	<b>Definición</b>
void InsertarOrden	dOrdenImpresión orden	Permite insertar una orden en la base de datos.
void ActualizarOrden	dOrdenImpresión orden	Dada una orden actualiza dicha orden en la base de datos.
void EliminarOrden	dOrdenImpresión orden	Dada una orden la elimina en la base de datos.
void InsertaSolicitudImpresión	dSolicitudImpresión solicitud	Dada una solicitud de impresión se inserta en la base de datos.
void ActualizarSolicitudImpresión	dSolicitudImpresión solicitud	Dada una solicitud se actualiza en la base de datos.
void EliminarSolicitudImpresión	dSolicitudImpresión solicitud	Dada una solicitud de impresión se elimina de la base de datos.
void InsertarDocumento	dDocumentoidentificación documento	Permite insertar un documento en la base de datos.
void ActualizarDocumento	dDocumentoidentificación documento	Dado un documento actualiza dicho documento en la base de datos.
void EliminarDocumento	dDocumentoidentificación documento	Dado un documento elimina dicho documento en la base de datos.
dDocumentoidentificación DocumentoNombre	Int64 carnet	Dado un carnet de identidad devuelve el documento de identificación.

Tabla 14. Descripción de los métodos principales de la clase conexión del componente de personalización.  
Fuente: Elaboración propia.

### Descripción de la clase controladora y conectora del componente de entrega.

<b>Nombre</b>	Negocio_Entrega	
<b>Descripción</b>	Es clase encargada de gestionar las operaciones para efectuar la entrega del documento de identificación al ciudadano.	
<b>Métodos</b>	<b>Parámetros</b>	<b>Definición</b>
IList<Documentoidentificación> Filtrar	string Filtro, string criterio	Efectúa la búsqueda de un documento listo para entregar según

		un criterio especificado.
Void RealizarEntrega	Documentoidentificacion documento	Efectúa la operación de la entrega del documento de identificación.

**Tabla 15. Descripción de los métodos principales de la lógica del negocio del componente de entrega. Fuente: Elaboración propia.**

<b>Nombre</b>	C_Conexión	
<b>Descripción</b>	Gestiona cada una de las operaciones para ejecutar las consultas en la base de datos.	
<b>Métodos</b>	<b>Parámetros</b>	<b>Definición</b>
void InsertarDocumento	dDocumentoidentificacion documento	Permite insertar un documento en la base de datos.
void ActualizarDocumento	dDocumentoidentificacion documento	Dado un documento actualiza dicho documento en la base de datos.
void EliminarDocumento	dDocumentoidentificacion documento	Dado un documento elimina dicho documento en la base de datos.
dDocumentoidentificacion DocumentoNombre	Int64 carnet	Dado un carnet de identidad devuelve un documento de identificación.
Boolean es Responsable	Int64 carnet, dDocumentoidentificacion documento	Verifica si la persona que se dirige a recoger el documento es responsable.
IList<dDocumentoidentificacion> DocumentoTipo	nTipoDocumento tipoDocumento	Dado un tipo de documento devuelve una lista de ese tipo de documento que fue pasado por parámetro.
dImagen ObtenerImagen	dCiudadano ciudadano	Dado un ciudadano obtiene la foto del ciudadano.
void Actualizar Documento	dDocumentoidentificacion documento, nEstadoDocumento estado	Dado un documento y un estado actualiza ese

		documento.
void InsertarEntrega	dEntregaDocumento entrega	Dada una entrega la inserta en la base de datos.
void ActualizarEntrega	dEntregaDocumento entrega	Dada una entrega la actualiza en la base de datos.
void EliminarEntrega	dEntregaDocumento entrega	Dada una entrega la elimina en la base de datos.

**Tabla 16. Descripción de los métodos principales de la clase conexión del componente de entrega. Fuente: Elaboración propia.**

### Descripción de las entidades de los componentes de personalización y entrega.

<b>Nombre</b>	dorden_impresión	
<b>Descripción de la entidad</b>	Es uno de los pasos a seguir a la hora de realizar un trámite, este paso permite la personalización y creación del documento de identificación.	
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>
id_orden_impresión	Llave primaria	uuid
estado_orden	Llave foránea	serial
fecha_creación	-	timestamp
prioridad	-	integer
id_tipo_documento	Llave foránea	serial

**Tabla 17. Descripción de la entidad dorden\_impresión. Fuente: Tomado del Diccionario de datos del proyecto Plataforma Modular de Identificación y Control de Acceso.**

<b>Nombre</b>	dsolicitud_impresión	
<b>Descripción de la entidad</b>	Es la necesidad de esa persona de la creación de un documento de identificación planteada de forma formal escrita o digital.	
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>
id_solicitud_impresión	Llave primaria	uuid
prioridad	-	integer
fecha_hora	-	timestamp
id_ciudadano	Llave foránea	uuid
id_orden_impresión	Llave foránea	uuid

id_tipo_documento	Llave foránea	serial
id_estado_solicitud_impresión	Llave foránea	serial

Tabla 18.Descripción de la entidad dsolicitud\_impresión. Fuente: Tomado del Diccionario de datos del proyecto Plataforma Modular de Identificación y Control de Acceso.

<b>Nombre</b>	dentrega_documento	
<b>Descripción de la entidad</b>	Acción en la que el ciudadano obtiene su documento de identificación.	
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>
id_entrega	Llave primaria	uuid
fecha_hora	-	timestamp
id_documento_identificación	Llave foránea	uuid
id_ciudadano	Llave foránea	uuid
id_tipo_entrega	Llave foránea	serial

Tabla 19.Descripción de la entidad dentrega\_documento. Fuente: Tomado del Diccionario de datos del proyecto Plataforma Modular de Identificación y Control de Acceso.

<b>Nombre</b>	nestado_solicitud_impresión	
<b>Descripción de la entidad</b>	Descripción de los estados que puede tener una solicitud de impresión.	
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>
id_estado_solicitud_impresión	Llave primaria	serial
clasificación	-	varchar

Tabla 20.Descripción de la entidad nestado\_solicitud\_impresión. Fuente: Tomado del Diccionario de datos del proyecto Plataforma Modular de Identificación y Control de Acceso.

<b>Nombre</b>	ntipo_entrega	
<b>Descripción de la entidad</b>	Da una descripción a cada entrega del documento que se realiza.	
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>
id_tipo_entrega	Llave primaria	serial
clasificación	-	varchar

Tabla 21.Descripción de la entidad ntipo\_entrega. Fuente: Tomado del Diccionario de datos del proyecto Plataforma Modular de Identificación y Control de Acceso.

<b>Nombre</b>	ddocumento_identificación	
---------------	---------------------------	--

<b>Descripción de la entidad</b>	Es lo que va a permitir identificar al ciudadano.	
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>
id_documento_identificación	-	uuid
serial	-	varchar
fecha_creación	-	timestamp
fecha_vencimiento	-	timestamp
id_ciudadano	Llave foránea	uuid
id_estado_documento	Llave foránea	serial
id_tipo_documento	Llave foránea	serial

**Tabla 22. Descripción de la entidad ddocumento\_identificación. Fuente: Tomado del Diccionario de datos del proyecto Plataforma Modular de Identificación y Control de Acceso.**

<b>Nombre</b>	dimagen	
<b>Descripción de la entidad</b>	Es donde se capturan las imágenes de las personas a las cuales se le está haciendo el trámite.	
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>
id_imagen	Llave primaria	uuid
imagen	-	byte
id_tipo_imagen	Llave foránea	serial
id_ciudadano	Llave foránea	uuid

**Tabla 23. Descripción de la entidad dimagen. Fuente: Tomado del Diccionario de datos del proyecto Plataforma Modular de Identificación y Control de Acceso.**

<b>Nombre</b>	nestado_documento	
<b>Descripción de la entidad</b>	Descripción de los estados que puede tener un documento de identificación.	
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>
id_estado_documento	Llave primaria	serial
clasificación	-	varchar

**Tabla 24. Descripción de la entidad nestado\_documento. Fuente: Tomado del Diccionario de datos del proyecto Plataforma Modular de Identificación y Control de Acceso.**

<b>Nombre</b>	nestado_orden	
---------------	---------------	--

<b>Descripción de la entidad</b>	Descripción de los estados que puede tener una orden.	
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>
id_estado_orden	Llave primaria	serial
clasificación	-	varchar

**Tabla 25. Descripción de la entidad nestado\_orden. Fuente: Tomado del Diccionario de datos del proyecto Plataforma Modular de Identificación y Control de Acceso.**

<b>Nombre</b>	ntipo_documento	
<b>Descripción de la entidad</b>	Descripción de los estados que puede tener una orden.	
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>
id_tipo_documento	Llave primaria	serial
clasificación		varchar

**Tabla 26. Descripción de la entidad ntipo\_documento. Fuente: Tomado del Diccionario de datos del proyecto Plataforma Modular de Identificación y Control de Acceso.**

<b>Nombre</b>	dciudadano	
<b>Descripción de la entidad</b>	Es cualquiera persona que de alguna forma está vinculado a la institución que necesite un documento de identificación para acceder a la misma.	
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>
id_cuidadano	Llave primaria	uuid
sexo	-	char
fecha_nacimiento	-	timestamp
primer_nombre	-	varchar
segundo_nombre	-	varchar
primer_apellido	-	varchar
segundo_apellido	-	varchar
id_tipo_ciudadano	Llave foránea	serial

**Tabla 27. Descripción de la entidad dciudadano. Fuente: Tomado del Diccionario de datos del proyecto Plataforma Modular de Identificación y Control de Acceso.**

<b>Nombre</b>	ntipo_imagen	
---------------	--------------	--

<b>Descripción de la entidad</b>	Da una descripción a cada imagen que se captura de las personas a las cuales se les está haciendo el trámite.	
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>
id_tipo_imagen	Llave primaria	serial
imagen	-	varchar

**Tabla 28. Descripción de la entidad ntipo\_imagen. Fuente: Tomado del Diccionario de datos del proyecto Plataforma Modular de Identificación y Control de Acceso.**

<b>Nombre</b>	ntipo_ciudadano	
<b>Descripción de la entidad</b>	Da una clasificación de cada ciudadano que está registrado en el sistema.	
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>
id_tipo_ciudadano	Llave primaria	serial
clasificación	-	varchar

**Tabla 29. Descripción de la entidad ntipo\_ciudadano. Fuente: Tomado del Diccionario de datos del proyecto Plataforma Modular de Identificación y Control de Acceso.**

## **Anexo 8. Diagrama de secuencia del componente de entrega.**

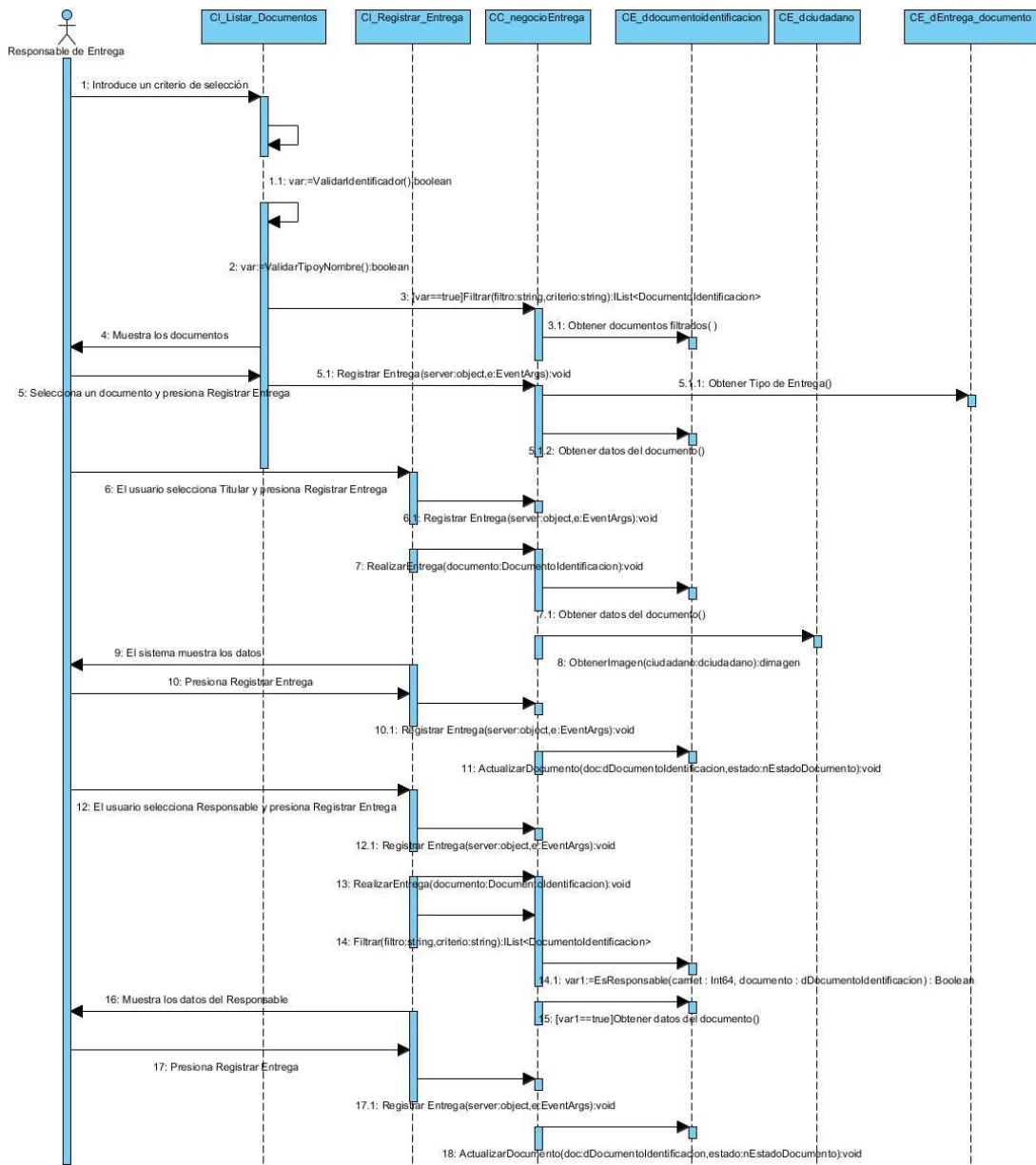


Figura 16. Diagrama de secuencia del componente entrega: Registrar entrega. Fuente: Elaboración propia.

**Anexo 9. Descripción de los servicios.**

<b>Nombre</b>	OrderImportService		
<b>Descripción</b>	Servicio a consumir para crear las órdenes.		
<b>Métodos</b>	<b>Parámetros</b>	<b>Definición</b>	
ImportResponse ImportOrder	Order order	Recibe una entidad orden según el tipo de orden de	



		una solicitud que contiene todas las solicitudes y devuelve el estado de la importación.
--	--	--

Tabla 30. Descripción del servicio OrderImportService. Fuente: Elaboración propia.

<b>Nombre</b>	ServicioLotes.	
<b>Descripción</b>	Servicio que responde a cada una de las acciones del servicio LotServiceQuery.	
<b>Métodos</b>	<b>Parámetros</b>	<b>Definición</b>
List<LotRequestInfo> getLotRequests	int page, int cant, int columnSelected, int sortMode, string filter, List<ColModel> colModel	Recibe una entidad orden según el tipo de orden de una solicitud que contiene todas las solicitudes y devuelve el estado de la importación.
int countLotRequests	int columnSelected, int sortMode, string filter, List<ColModel> colModel	Cuenta la cantidad de solicitudes de lotes.
void receiveLots	string identifier	Dado un identificador, recibe los lotes de una solicitud

Tabla 31. Descripción del servicio ServicioLotes. Fuente: Elaboración propia.

## Anexo 10. Estándares de codificación utilizados.

### Convenciones para los Nombres.

Indicadores	Ejemplos
Nombre de las clases	Usar la convención Pascal para el nombre de las clases  <i>public class HelloWorld</i>  {... }
Interfaces	Usar el prefijo "I" con la convención Camel para las interfaces (Ejemplo: <i>IEntity</i> )
Nombre de los métodos	Usar la convención Pascal para el nombre de los métodos  <i>void SayHello(string name){</i>  ...}
Nombre de variables y parámetros	Usar la convención Camel para el nombre de variables y parámetros

	<pre>int totalCount = 0; void SayHello(string name) {     string fullMessage = "Hello " + name;     ... }</pre> <p>Usar el significado de las palabras para el nombre de las variables, no utilizar abreviaturas.</p> <p>Bien:</p> <pre>string address int salary</pre> <p>Mal:</p> <pre>string nam string addr int sal</pre> <p>No usar caracteres simples para el nombre de las variables i, n, s etc. Emplear nombres como index, temp</p> <p>Una excepción de esta regla son las variables de los ciclos</p> <pre>for ( int i = 0; i &lt; count; i++ ) {     ... }</pre> <p>No usar underscores ( _ ) para los nombres de las variables locales.</p> <p>Todas las variables miembros de las clases deben ser prefijadas con <i>underscore</i> ( _ ) para ser diferenciadas de otras variables.</p> <p>No usar nombres de variables que coincidan con palabras reservadas.</p> <p>Las variables booleanas y las propiedades se les pueden poner como prefijos "is".</p> <p>Ejemplo: <i>private bool _isFinished</i></p>
--	--

Nombre de los Namespace	<p>Deben seguir el siguiente estándar:</p> <p><i>&lt;company name&gt;.&lt;product name&gt;.&lt;top level module&gt;.&lt;bottom level module&gt;</i></p> <p>Para el caso del proyecto: Plataforma Modular de Identificación y Control de Acceso.</p> <p><i>&lt; product name &gt;.&lt;top level module&gt;.&lt;bottom level module&gt;</i></p> <p>Ejemplo: <i>PMICA.Workflow.Runtime</i></p>
-------------------------	---

**Tabla 32. Convenciones para los nombres. Fuente: Tomado del informe : Estándares de codificación para C#. Proyecto Plataforma Modular de Identificación y Control de Acceso**

### Buenas prácticas de programación.

- Evitar escribir métodos de más de 25 líneas.
- El nombre del método debe decir que hace. No se deben usar abreviaturas.
- Cada método debe cumplir solamente una función. No se deben combinar más de una función por método, tratar además que las funcionalidades sean los más atómicas posibles.
- Si un valor errado es encontrado en los ficheros de configuración, la aplicación debe mostrar al usuario los valores correctos.
- No programar más de una clase por cada fichero.
- Evitar tener ficheros muy largos. Se deben dividir en dos o más clases.
- Evitar el paso de demasiados parámetros a los métodos, tratar de que no haya métodos de más de 3 parámetros.

### Anexo 11: Descripción de las entidades y nomencladores del modelo de datos.

<b>Nombre de la entidad</b>		dsolicitud_impresión					
<b>Descripción de la entidad</b>		Entidad que almacena los datos de una solicitud.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_solicitud_impresión	Llave primaria de la solicitud.	uuid	no			no	si
prioridad		integer	si			si	no
fecha_hora	Fecha y Hora de	timestam p	no			no	si

	creación de la solicitud.						
id_ciudadano	Llave foránea del ciudadano al que le pertenece la solicitud.	uuid	no			no	si
id_orden_impresión	Llave foránea de la orden a la que pertenece la solicitud.	uuid	no			no	si
id_tipo_documento	Llave foránea del tipo de documento que es la solicitud.	integer	no			si	no
id_estado_solicitud_impresión	Llave foránea del estado de la solicitud.	integer	no			si	no

**Tabla 33. Descripción de la entidad dsolicitud\_impresión. Fuente: Tomado del Diccionario de datos del Proyecto Plataforma Modular de Identificación y Control de Acceso.**

<b>Nombre de la entidad</b>		ddocumento_identificación					
<b>Descripción de la entidad</b>		Es la entidad que guarda los datos de los documentos personalizados.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_documento_identificación	Llave primaria	uuid	no			no	si

serial		varchar	no	20		no	si
fecha_creación		timestamp	no			si	no
fecha_vencimiento		timestamp	no			si	no
id_ciudadano	Llave foránea	uuid	no			no	si
id_estado_documento	Llave foránea	integer	no			si	no
id_tipo_documento	Llave foránea	integer	no			si	no
<b>Nombre de la entidad</b>		dorden_impresión					
<b>Descripción de la entidad</b>		Entidad que guarda la orden creada					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_orden_impresión	Llave primaria	uuid	no			no	si
estado_orden	Llave foránea	integer	no			si	no
fecha_creación		timestamp	no			si	no
prioridad		integer	si			si	no
id_tipo_documento	Llave foránea	integer	si			si	no

**Tabla 34.** Descripción de las entidades dorden\_impresión y ddocumento\_identificación. Fuente: Tomado del Diccionario de datos del Proyecto Plataforma Modular de Identificación y Control de Acceso.

<b>Nombre de la entidad</b>		ntipo_documento					
<b>Descripción de la entidad</b>		Describe el tipo de documento por el cual se va a permitir la identificación de la persona.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_tipo_documento	Llave primaria	serial	no			no	si
clasificación		varchar	no	20		no	si

**Tabla 35.** Descripción del nomenclador ntipo\_documento. Fuente: Tomado del Diccionario de datos del Proyecto Plataforma Modular de Identificación y Control de Acceso.

<b>Nombre de la entidad</b>	ntipo_entrega
-----------------------------	---------------

<b>Descripción de la entidad</b>		Da una descripción a cada entrega del documento que se realiza.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_tipo_entrega	Llave primaria	serial	no			no	si
clasificación		varchar	no	20		no	si

Tabla 36. Descripción del nomenclador ntipo\_entrega. Fuente: Tomado del Diccionario de datos del Proyecto Plataforma Modular de Identificación y Control de Acceso.

<b>Nombre de la entidad</b>		dentrega_documento					
<b>Descripción de la entidad</b>		Acción en la que el ciudadano obtiene su documento de identificación.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_entrega	Llave primaria	uuid	no			no	si
fecha_hora		timestamp	no			si	no
id_documento_identificación	Llave foránea	uuid	no			no	si
id_ciudadano_recibe	Llave foránea	uuid	no			si	no
id_tipo_entrega	Llave foránea	integer	no			si	no

Tabla 37. Descripción de la entidad dentrega\_documento. Fuente: Tomado del Diccionario de datos del Proyecto Plataforma Modular de Identificación y Control de Acceso.

<b>Nombre de la entidad</b>		nestado_documento					
<b>Descripción de la entidad</b>		Descripción de los estados que puede tener un documento de identificación.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_estado_documento	Llave Primaria	serial	no			no	si
clasificación		varchar	no	20		no	si

Tabla 38. Descripción del nomenclador nestado\_documento. Fuente: Tomado del Diccionario de datos del Proyecto Plataforma Modular de Identificación y Control de Acceso.

<b>Nombre de la entidad</b>		nestado_orden					
<b>Descripción de la</b>		Descripción de los estados que puede tener una orden.					

entidad							
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_estado_orden	Llave primaria	serial	no			no	si
clasificación		varchar	no	20		no	si

**Tabla 39.** Descripción del nomenclador nestado\_orden. Fuente: Tomado del Diccionario de datos del Proyecto Plataforma Modular de Identificación y Control de Acceso.

Nombre de la entidad		nestado_solicitud_impresión					
Descripción de la entidad		Descripción de los estados que puede tener una solicitud de impresión.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_estado_solicitud_impresion	Llave primaria	serial	no			no	si
clasificación		varchar	no	20		no	si

**Tabla 40.** Descripción del nomenclador nestado\_solicitud\_impresión. Fuente: Tomado del Diccionario de datos del Proyecto Plataforma Modular de Identificación y Control de Acceso.

## Anexo 12: Interfaces del sistema.

Interfaces del sistema del componente de entrega correspondiente a los requisitos: RFG1. Buscar trámite, RFG2. Listar trámites, RFG3. Buscar información de la persona, RFG4. Listar personas, RFG5. Mostrar información del ciudadano, RFG6. Actualizar estado del documento, RFG7. Actualizar estado de las fases del trámite, RFE1. Registrar el tipo de entrega, RFE2. Mostrar personas autorizadas a recoger el documento, RFE4. Registrar entrega del documento, RFE5. Generar comprobante de entrega.

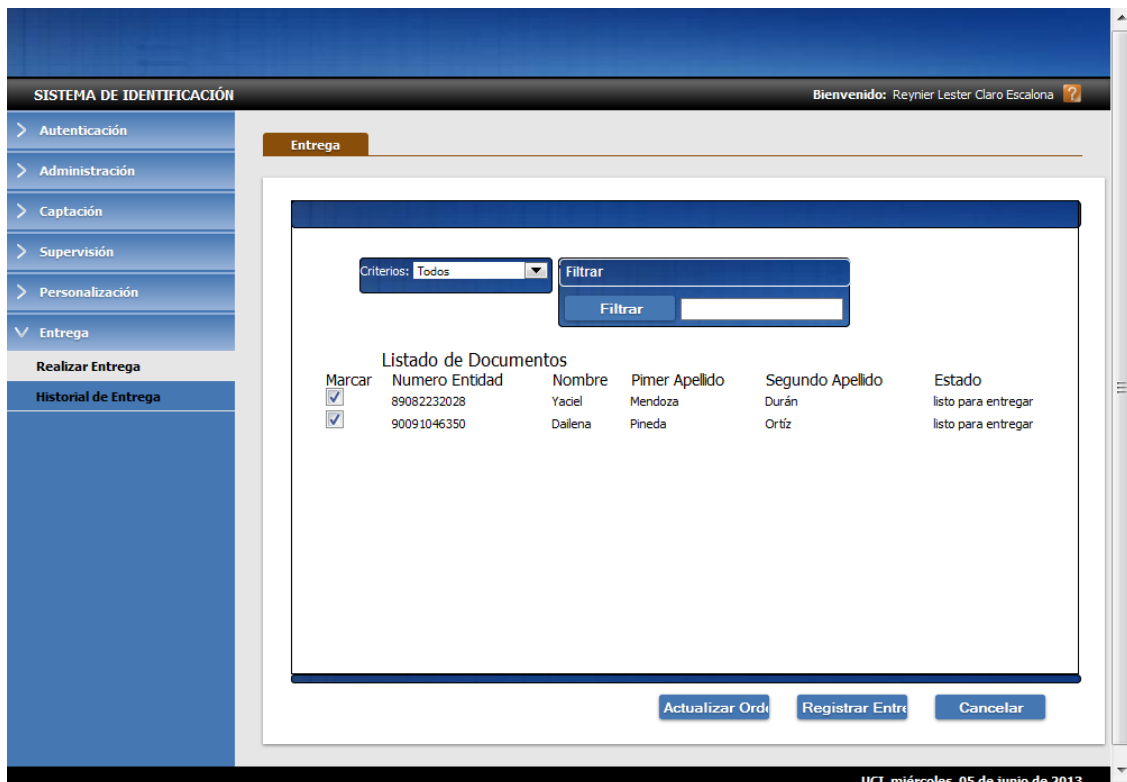


Figura 17.Registrar entrega del documento. Fuente: Elaboración propia.

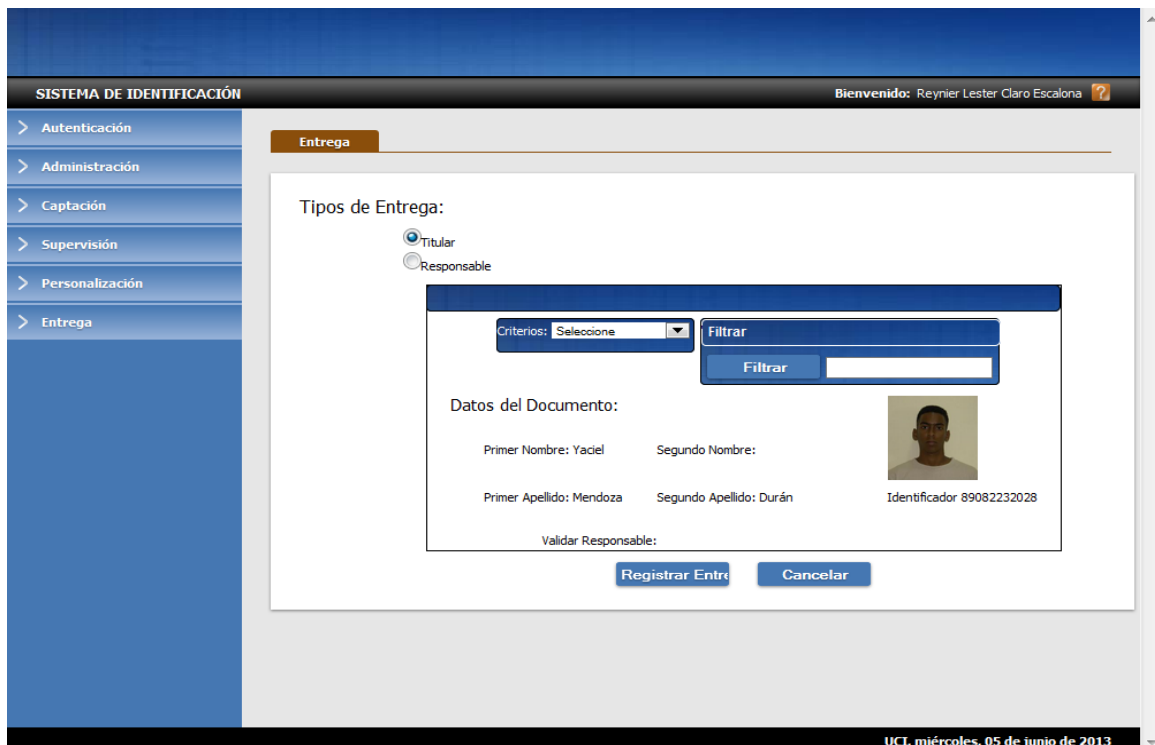


Figura 18.Registrar Entrega del documento a Titular. Fuente: Elaboración propia.



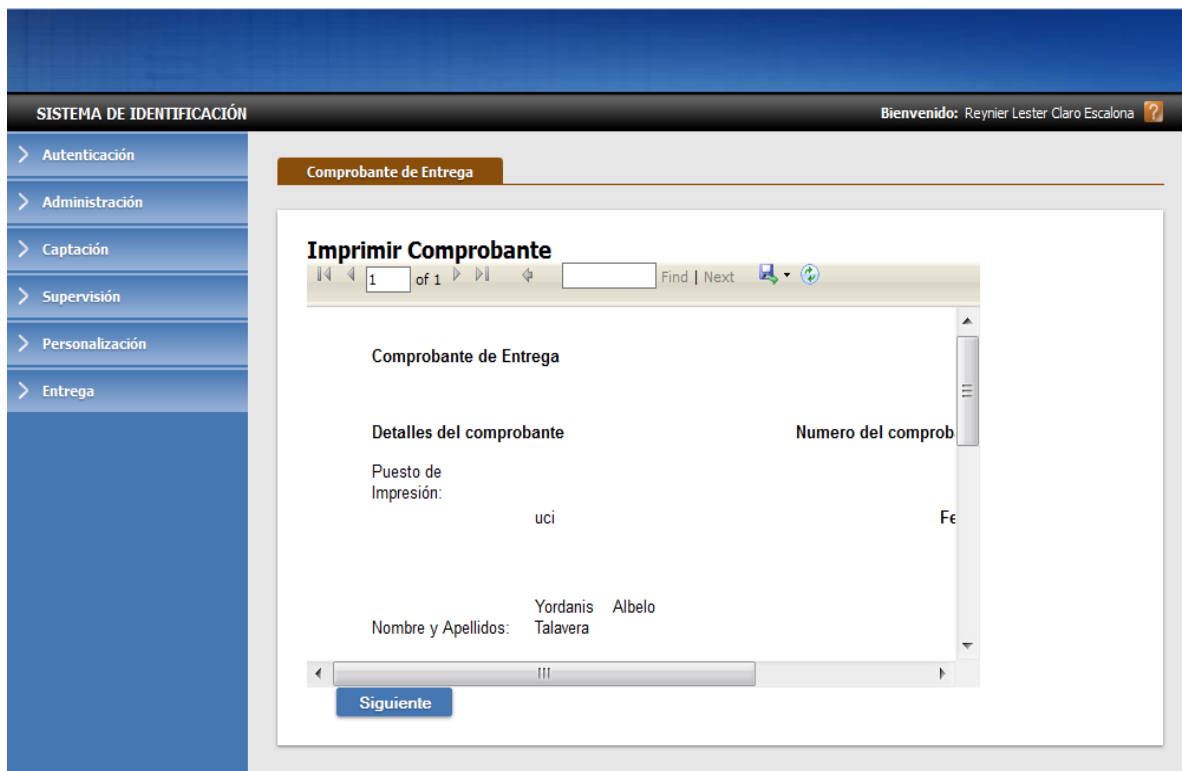


Figura 19. Imprimir comprobante. Fuente: Elaboración propia.

### Anexo 13. Diseño de los Casos de Prueba.

No	Nombre de Campo	Clasificación	Valor Nulo	Descripción
3	Marcar	Campo de selección	Sí	Este campo puede o no estar seleccionado.
4	Criterios_Entrega	Lista desplegable	Sí	Este campo contiene varios criterios como: Carnet de identidad, Nombre y Apellidos.
5	Filtrar_Entrega	Campo de texto	Sí	Este campo admite solamente letras si el criterio especificado fue: Carnet de identidad, Nombre y Apellidos. Admite solamente números si el criterio especificado fue Carnet de Identidad y admite solamente letras si el criterio especificado fue Nombre y Apellidos.
6	Tipo_Entrega	Campo de selección	Sí	Este campo puede o no estar seleccionado en dependencia de los valores que puede tomar:

				Titular y Responsable.
7	Criterios_Responsable	Lista desplegable.	Sí	Este campo contiene el criterio Carnet de identidad.
8	Filtrar_Responsable	Campo de texto	Sí	Este campo admite solamente números ya que el criterio especificado es el Carnet de identidad.

Tabla 41. Variables del diseño de caso de prueba: Entrega de documentos. Fuente: Elaboración propia.

Escenario	Descripción	V 4	V 5	V 3	V 6	V 7	V 8	Respuesta del sistema.	Flujo central
E.C 1.1 Filtrar	El usuario selecciona un criterio y luego introduce los datos por los que va a filtrar y luego oprime el botón filtrar.	V (Carnet de Identidad)	V (90120805774)	NA	NA	NA	NA	El sistema muestra todos los trámites que se correspondan con ese filtro.	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar
		I(Valor por defecto Selección)	V (90120805774)	NA	NA	NA	NA	El sistema no muestra ningún trámite ya que no se ha seleccionado ningún trámite.	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar
		V (Carnet de Identidad)	I(Se han introducido letras en el campo de texto)	NA	NA	NA	NA	El sistema muestra un mensaje de error ya que ese campo de texto no acepta letras y no muestra ningún trámite.	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar
		I(Valor por defecto Selección)	I( Campos vacíos)	NA	NA	NA	NA	No se muestra ningún	1. Entrega. 2. Realizar Entrega. 3. Criterios.

		one)						trámite.	4. Filtrar
		V (Nombre y Apellidos)	V (Yordanis Albelo Talavera)	NA	NA	NA	NA	El sistema muestra el trámite correspondiente al tipo de criterio especificado.	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar
		V (Nombre y Apellidos)	I (Yordanis) Se ha especificado un solo parámetro de entrada.	NA	NA	NA	NA	El sistema no muestra ningún trámite ya que solamente fue entrado un solo parámetro.	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar
		V (Nombre y Apellidos)	I (56Juan Pérez) Se han introducido números.	NA	NA	NA	NA	El sistema muestra un mensaje de error ya que no se deben introducir números	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar
		I (No se ha seleccionado ningún criterio)	V (Cualquier texto)	NA	NA	NA	NA	El sistema no muestra ningún trámite ya que no se ha seleccionado ningún trámite.	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar
		I (No se ha seleccionado ningún criterio)	I (Campos vacíos)	NA	NA	NA	NA	El sistema no muestra ningún trámite ya que no se ha seleccionado ningún trámite y los campos están vacíos.	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar

Tabla 42. Diseño de caso de prueba: Entrega de documentos. Escenario Filtrar. Fuente: Elaboración propia.

Escenario	Descripción	V 4	V 5	V 3	V 6	V 7	V 8	Respuesta del sistema.	Flujo central
E.C 1.2.Registrar Entrega	El usuario selecciona la opción Marcar, seleccionándose el documento por el que se va a realizar la entrega y luego oprime el botón Registrar Entrega.	NA	NA	V(Cuando se selecciona uno o varios elementos)	NA	NA	NA	El sistema muestra la interfaz que posee los Tipos de Entrega.	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar. 5. Actualizar orden. 6. Registrar Entrega.
		NA	NA	I(Cuando no se selecciona ningún elemento)	NA	NA	NA	El sistema muestra un error ya que no se ha seleccionado ningún documento.	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar. 5. Actualizar orden. 6. Registrar Entrega.

Tabla 43. Diseño de caso de prueba: Entrega de documentos. Escenario Registrar Entrega. Fuente: Elaboración propia.

Escenario	Descripción	V 4	V 5	V 3	V 6	V 7	V 8	Respuesta del sistema.	Flujo central
E.C 1.3. Seleccionar el Tipo de Entrega.	El usuario selecciona el campo de selección Titular, mostrándose por defecto un listado de	NA	NA	NA	V (Cuando se ha seleccionado la opción Titular)	NA	NA	El sistema actualizará el estado del documento, la fase del trámite y mostrará la interfaz de Imprimir Comprobant	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar. 5. Actualizar orden. 6. Registrar Entrega. 7. Titular.

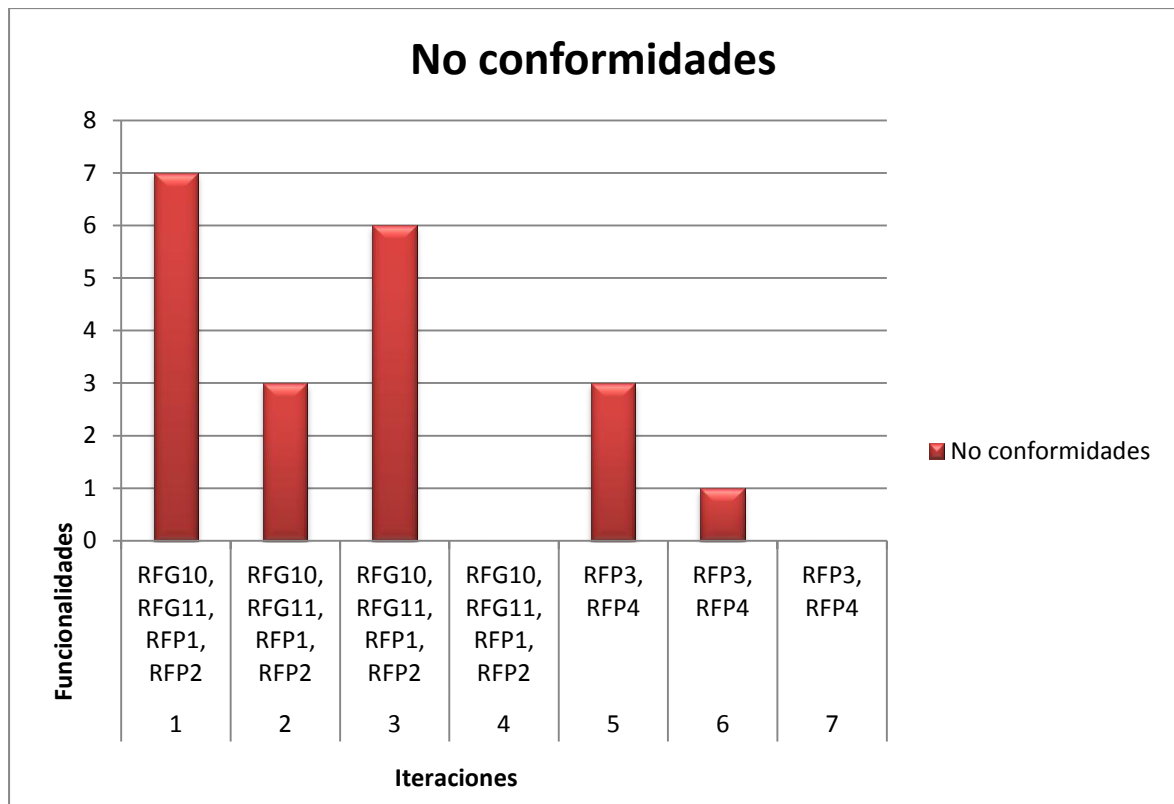
	<p>critérios, un campo de texto filtrar. Además de mostrar los datos principales del Titular. Luego oprime el botón Registrar Entrega.</p>				r). NA (Responsable)			e.	
		NA	NA	NA	I (Cuando no se ha seleccionado la opción Titular) . NA(Responsable)	NA	NA	El sistema no muestra ningún dato asociado al Titular, ya que no se ha seleccionado la opción.	<ol style="list-style-type: none"> <li>1. Entrega.</li> <li>2. Realizar Entrega.</li> <li>3. Criterios.</li> <li>4. Filtrar.</li> <li>5. Actualizar orden.</li> <li>6. Registrar Entrega.</li> <li>7. Titular.</li> </ol>
	<p>El usuario selecciona el campo de selección Responsable, mostrándose el listado de criterios y el campo de texto filtrar. Luego de realizar la búsqueda del</p>	NA	NA	NA	V (Se ha seleccionado la opción Responsable) . NA (Titular)	V (Carnet de Identidad)	V (90120805774)	El sistema muestra el trámite asociado a la persona Responsable . El sistema actualizará el estado del documento, la fase del trámite y mostrará la interfaz Imprimir Comprobant	<ol style="list-style-type: none"> <li>1. Entrega.</li> <li>2. Realizar Entrega.</li> <li>3. Criterios.</li> <li>4. Filtrar.</li> <li>5. Actualizar orden.</li> <li>6. Registrar Entrega.</li> <li>7. Responsable</li> </ol>

	responsable, oprime el botón Registrar Entrega.							e.	
		NA	NA	NA	I(No se ha seleccionado la opción Responsable) NA (Titular)	I(Nó se seleccionó ningún criterio)	I ( No se introdujo ningún dato en el campo de texto).	No se muestra ningún trámite ya que no se seleccionó la opción.	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar. 5. Actualizar orden. 6. Registrar Entrega. 7. Responsable .
		NA	NA	NA	V ( Se ha seleccionado la opción Responsable ) NA (Titular)	V (Carnet de Identidad)	I (Al 90120805774).	Muestra un mensaje de error ya que este campo no admite letras.	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar. 5. Actualizar orden. 6. Registrar Entrega. 7. Responsable .
		NA	NA	NA	V ( Se ha seleccionado la opción	V (Carnet de Identidad)	I ( Campo de texto	Muestra un mensaje de error ya que en el campo de texto no introdujo ningún dato.	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar. 5. Actualizar orden. 6. Registrar

					Responsable) NA (Titular)		vacío)		Entrega. 7. Responsable .
		NA	NA	NA	V (Se ha selec ciona do la opción n Resp onsa ble). NA(T itular )	I (No se sel ecc ion a nin gún cri terio de sel ecc ión)	V (90 120 805 774 )	Muestra un mensaje de error ya que no se seleccionó ningún criterio de selección.	1. Entrega. 2. Realizar Entrega. 3. Criterios. 4. Filtrar. 5. Actualizar orden. 6. Registrar Entrega. 7. Responsable .

Tabla 44. Diseño de caso de prueba: Entrega de documentos. Escenario seleccionar el tipo de entrega.  
Fuente: Elaboración propia.

### Resultados de las pruebas de los componentes de personalización y entrega.



**Figura 20.**Gráfico de los resultados a las pruebas funcionales del componente personalización. Fuente: Elaboración propia.

Iteraciones	Funcionalidades	No Conformidades
1	RFG11,RFG12	5
2	RFG11,RFG12	3
3	RFG11,RFG12	2
4	RFG11,RFG12	0
5	RFG1,RFG2,RFG3,RFG4,RFG5	7
6	RFG1,RFG2,RFG3,RFG4,RFG5	6
7	RFG1,RFG2,RFG3,RFG4,RFG5	4
8	RFG1,RFG2,RFG3,RFG4,RFG5	1
9	RFG1,RFG2,RFG3,RFG4,RFG5	0
10	RFE1,RFE2,RFE3,RFE4	8
11	RFE1,RFE2,RFE3,RFE4	7
12	RFE1,RFE2,RFE3,RFE4	6
13	RFE1,RFE2,RFE3,RFE4	3
14	RFE1,RFE2,RFE3,RFE4	0

**Tabla 45.**Resultados de las pruebas al componente de entrega. Fuente: Elaboración propia.



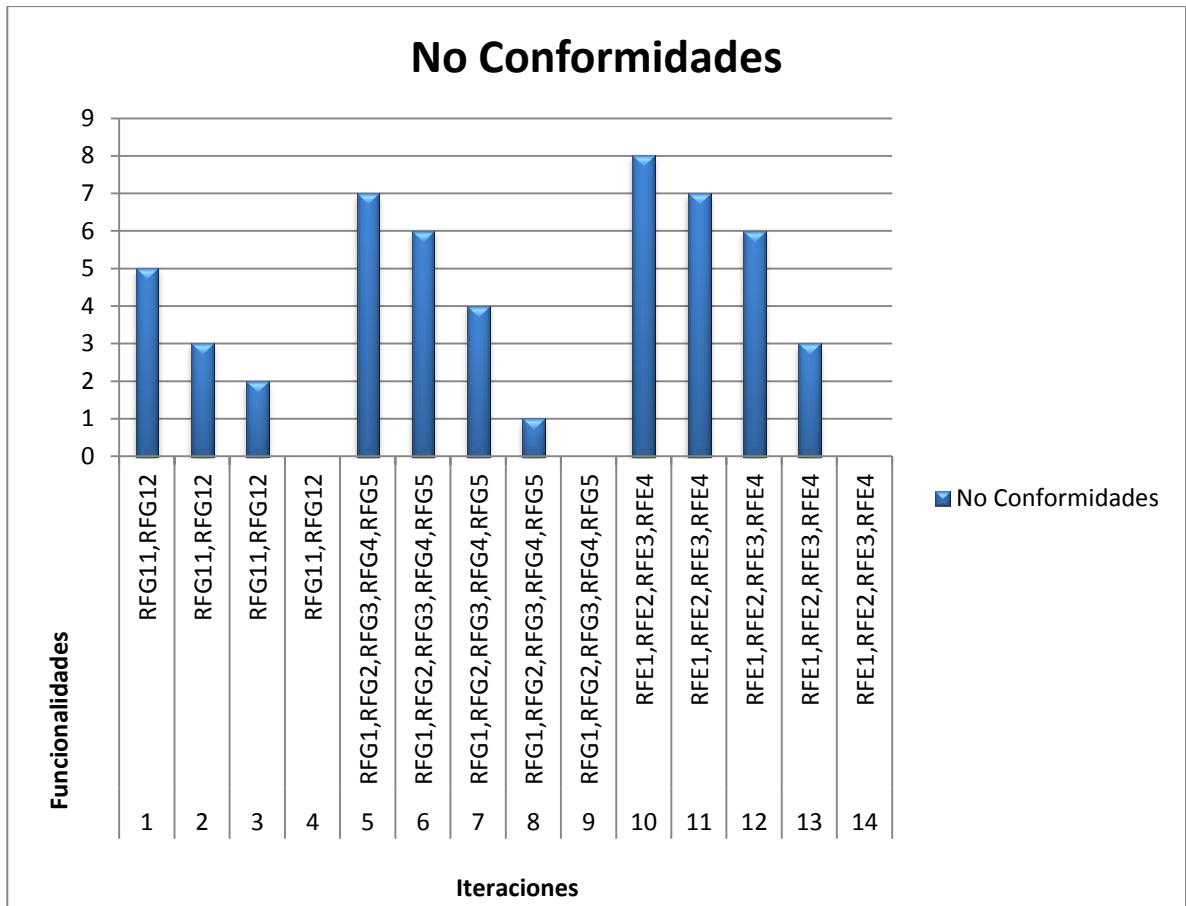


Figura 21. Gráfico de los resultados a las pruebas funcionales del componente Entrega.