

**Universidad de las Ciencias Informáticas
Facultad 1**



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

**Título: “Módulos de Captura de datos, imágenes y Supervisión para la
Plataforma Modular de Identificación y Control de Acceso”**



Autores: Mabel de la Caridad Ramos Delis
Danny Hernández Simons

Tutor: Ing. Renier Pérez García

La Habana, Junio de 2013



“Si los jóvenes fallan, todo fallará. Es mi más profunda convicción que la juventud cubana luchará por impedirlo. Creo en ustedes.”

Declaración de autoría:

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

Para que así conste firmamos la presente a los ____ días del mes de junio del año 2013.

Autores

Mabel de la Caridad Ramos Delis

Danny Hernández Simons

Firma del autor

Firma del autor

Tutor

Ing. Renier Pérez García

Firma del tutor

Dedico mi Trabajo de Diploma a mi mamita querida, por ser mi guía y haberme apoyado en todo momento de mi vida y de la carrera, te amo mami.

A mi hermanita Rosy que la quiero muchísimo.

A mi abuela por apoyarme en todo momento y por creer siempre en mí.

A mis padres por su amor y cariño.

Mabel de la Caridad Ramos Delis

Dedico mi Trabajo de Diploma a mi mamá y a mi abuela por ser mi luz, mi guía y haberme apoyado en todo.

Danny Hernández Simons

Mabel de la Caridad Ramos Delis

Les agradezco a todas las personas que de una forma me brindaron su apoyo y confiaron en mí.

A mi compañero de tesis que sin su ayuda no hubiese podido estar el día de hoy aquí.

A mi mami por estar siempre conmigo en todos los momentos de mi vida, por su apoyo y amor, por dármele todo en la vida y por confiar en todo momento en que yo lograría estar discutiendo mi tesis el día de hoy.

Te quiero mami.

A mi hermanita Rosy, porque te quiero con el alma malcriada.

A mi abuela por apoyarme siempre y tener tanta confianza en mí.

A mis padres y a mi familia por siempre estar presentes en mi vida.

A Annie por brindarme su ayuda incondicional siempre que la necesité en la realización de la tesis y por ser mi tutora de siempre.

A mis amistades, estas personitas que han significado mucho en mi vida por brindarme su amistad y estar presentes cuando las necesité, en especial: a Giselle mi otra hermana, gracias por tu amistad, apoyo y confianza, te quiero Yiyi, a Geny por esa confianza que siempre has tenido en mí, por la fuerza que me has dado en muchos momentos de la carrera para seguir adelante, a Nellis, Wendy, Yeimy, Roxana, Sairenys a Adys por su cariño y aguantar mis malcriadeces, mut, a Lienylla por su preocupación.

A Yaima por brindarme su ayuda siempre que la necesite y por su amistad.

A mis compañeros de grupo por los buenos momentos que pasamos: Aynel, Yam, Jose, Daniel, Herbert y Nilbert.

Agradecimientos

En general a todas las personas que de una forma u otra contribuyeron a que hoy yo pudiese estar aquí discutiendo mi trabajo de diploma.

Resumen

Con el avance de las nuevas Tecnologías de la Información y las Comunicaciones (TIC), se han logrado automatizar los procesos de identificación, a través de sistemas que permitan entre sus funcionalidades realizar la captura y supervisión de los datos e imágenes de las personas, para la identificación de las mismas.

La captura y supervisión de datos e imágenes son procesos que se realizan de forma diferente en cada sector que cuente con un sistema para la identificación de las personas pertenecientes al mismo, ya que el flujo de información de las personas, que se maneja en cada uno de ellos es diferente, además no todos disponen de los recursos necesarios para realizar dichos procesos. Con el desarrollo de un sistema de identificación genérico que cuente con los módulos de Captura y Supervisión de datos e imágenes que sean adaptables y configurables de forma rápida en diferentes sectores, como puede ser: el sector de la educación, el sector empresarial, el sector de la salud, el sector gubernamental, entre otros, se solucionarían estos problemas anteriormente mencionados.

La Universidad de las Ciencias Informáticas (UCI) es una universidad que cuenta con varios centros productivos dentro de los que se encuentra el Centro de Identificación y Seguridad Digital (CISED), en el mismo se han tenido experiencias en el desarrollo de sistemas de identificación como el Sistema automatizado para la Emisión del Pasaporte y Constancia de Cedulación en las Sedes Consulares de la República Bolivariana de Venezuela y el Sistema Único de Identificación Nacional de la República de Cuba, en estos sistemas se realizan los procesos de captura y supervisión de datos e imágenes de manera diferente, de acuerdo a las características de cada entidad. Por lo que no pueden ser empaquetados como un producto de *software* para su comercialización. La presente investigación tiene como objetivo el desarrollo de los módulos Captura de datos, Captura de imágenes y Supervisión que permitan gestionar la captura y supervisión de los datos e imágenes de un sistema de identificación lográndose adaptar a diferentes sectores. Para el desarrollo de la solución se emplearán las herramientas y tecnologías que permitan el desarrollo de la misma con la calidad y fiabilidad requerida para lograr los resultados esperados.

Palabras claves:

Captura de datos, Captura de imágenes, Supervisión, sistema de identificación.

Índice de contenido	
Introducción	1
Capítulo 1: Fundamentación teórica	6
Introducción	6
Conceptos asociados a los procesos de captura y supervisión de datos e imágenes	6
1.1. Soluciones existentes de sistemas de identificación en el ámbito internacional y nacional	6
1.2. Análisis del estándar internacional: Organización de Aviación Civil Internacional (OACI) asociado a los documentos de identificación de las personas.....	13
1.3. Tecnologías, metodología y herramientas a utilizar.....	15
1.3.1. Metodología FDD (<i>Feature Driven Development</i> / Desarrollo Basado en Funciones).....	15
1.3.2. Entorno integrado de desarrollo	16
1.3.3. Lenguaje de desarrollo	16
1.3.4. Herramienta de modelado	17
1.3.5. Notación de modelado: BPMN	17
1.3.6. Lenguaje de modelado: UML.....	17
1.3.7. Microsoft .NET Framework 4.0	18
1.3.8. Gestor de base de datos.....	18
1.3.9. Windows Workflow Foundation 3.5	19
1.3.10. Cliente TortoiseSVN-1.7.6.....	19
1.3.11. ORM NHibernate v3.0.....	20
1.3.12. BisonFramework	21
1.3.13. ASP.NET	21
1.3.14. UIToolsBox	22
1.3.15. JavaScript.....	22
1.3.16. CSS	22
Conclusiones parciales	22
Capítulo 2: Análisis y diseño de la propuesta de solución.....	24
Introducción	24
2.1. Flujo de procesos	24

2.2. Descripción de las actividades de los procesos captura de datos, captura de imágenes y supervisión.....	25
2.3. Requisitos funcionales.....	28
2.4. Requisitos no funcionales.....	35
2.5. Arquitectura propuesta	38
2.6. Patrones	40
2.7. Diseño de las funcionalidades.....	43
2.8. Diseño del <i>workflow</i> utilizado en la propuesta de solución.....	46
2.9. Modelo de datos.....	48
Conclusiones parciales	49
Capítulo 3: Implementación y pruebas.....	50
Introducción.....	50
3.1. Estándares de codificación.....	50
3.2. Tratamiento de errores	53
3.3. Vista de despliegue.....	53
3.4. Diagrama de componentes	54
3.5. Interfaces de usuarios	55
3.6. Pruebas.....	56
Conclusiones parciales	61
Conclusiones	62
Recomendaciones	63
Bibliografía referenciada	64
Bibliografía consultada	67
Glosario de términos	69
Anexos.....	70

Índice de figuras

Figura 1: Proceso captura de datos	25
Figura 2: Captar datos.....	33
Figura 3: Captar datos de una fuente externa.....	34
Figura 4: Vista lógica de la arquitectura.....	39
Figura 5: Ejemplo del uso del patrón Creador	41
Figura 6: Ejemplo del uso del patrón Singleton	42
Figura 7: Diagrama de clases del diseño: Registrar datos del ciudadano.	43
Figura 8: Diagrama de secuencia: Registrar datos del ciudadano.....	45
Figura 9: Workflow para la captura de los datos de un ciudadano	47
Figura 10: Modelo de datos.....	48
Figura 11: Estándar de codificación: Convención Pascal para el nombre de las clases.	50
Figura 12: Estándar de codificación: Prefijo “I” con la convención Camel para las interfaces.....	51
Figura 13: Estándar de codificación: Convención Pascal para el nombre de los métodos.	51
Figura 14: Estándar de codificación: Nombre de las variables.....	51
Figura 15: Estándar de codificación: Nombre de las variables	52
Figura 16: Estándar de codificación: Llaves y líneas en blanco	52
Figura 17: Estándar de codificación: Buenas prácticas de programación.....	53
Figura 18: Diagrama de despliegue.....	54
Figura 19 : Diagrama de componentes.....	54
Figura 20: Interfaz Registrar datos del ciudadano.	56
Figura 21: Resultado de la prueba unitaria realizada al método InsertarCiudadano.	57
Figura 22: Gráfico del resultado de las pruebas unitarias	59
Figura 23: Gráfico del resultado de las pruebas de caja negra.....	60

Índice de tablas

Tabla 1. Listado de datos de un documento de identificación de tamaño 1.....	13
Tabla 2. Descripción de las actividades principales de los procesos de negocio: Captura de datos, Captura de imágenes y Supervisión.	28
Tabla 3. Descripción de las funcionalidades del módulo Captura de datos.....	35
Tabla 4. Descripción de la clase BussinesFactory.....	44
Tabla 5. Descripción de la clase RepositoryFactory.....	44
Tabla 6. Descripción de la entidad CE_Ciudadano	45
Tabla 7. Resultados de las pruebas unitarias.....	58
Tabla 8: Resultados de las pruebas de caja negra.....	60

Introducción

Un documento de identificación es un elemento de suma importancia que permite identificar legal y oficialmente a un individuo. Este documento está compuesto por el nombre(s) y los apellidos de la persona, además de otros datos de relevancia para la identificación de la misma. Así mismo, permite realizar ciertas actividades cotidianas que sin un documento de identificación resultarían muy complejas de desarrollar. Para la creación de un documento de identificación es necesario realizar varios procesos, dentro de los que se encuentran los procesos de captura y supervisión de datos e imágenes, que constituyen elementos fundamentales para poder emitir un documento de identificación. Con el avance de las nuevas Tecnologías de la Información y las Comunicaciones (TIC), se han logrado automatizar los procesos de identificación, a través de la creación de sistemas que permitan entre sus funcionalidades realizar la captura y supervisión de los datos e imágenes de las personas para la confección de dicho documento.

Cada sector realiza la captura y supervisión de datos e imágenes de acuerdo a las características y propósitos del mismo, por lo que el flujo de información que se maneja de las personas en cada uno de ellos es diferente. Además no todos cuentan con los recursos necesarios para realizar estos procesos, por lo que resulta engorroso crear un sistema de identificación específico para cada institución. La necesidad de desarrollar un sistema de identificación genérico surge con la idea de tener un producto de *software* que sea configurable y adaptable a las características de los distintos sectores, como puede ser el sector de la educación: por ejemplo en las universidades para emitir las credenciales de sus estudiantes, en el sector empresarial para emitir las credenciales de los clientes, proveedores y empleados, en el sector de la salud para emitir el carnet del paciente o del médico, en el sector gubernamental para la emisión de pasaportes y carnet de identidad, entre otros sectores.

La Universidad de las Ciencias Informáticas (UCI), creada por el Comandante en Jefe Fidel Castro Ruz es una universidad que tiene entre sus principales objetivos la producción de *software*, contribuyendo así al desarrollo de la informática en el país. La UCI cuenta con varios centros productivos, dentro de los que se encuentra el Centro de Identificación y Seguridad Digital (CISED). En el mismo se han tenido experiencias en el desarrollo de sistemas de identificación como el Sistema automatizado para la Emisión del Pasaporte y Constancia de Cedulación en las Sedes Consulares de la República Bolivariana de Venezuela y el Sistema Único de Identificación Nacional de la República de Cuba. En estos sistemas se realizan los procesos de captura y supervisión de datos e imágenes de manera diferente ajustándose a las

características y condiciones de cada entidad. Por estas razones no pueden ser empaquetados como un producto de *software* para su comercialización, ya que fueron desarrollados con un propósito específico.

Para resolver esta situación surgió la idea de desarrollar en el CISED un sistema de identificación de personas que cuente con varios módulos, dentro de los que se encuentran los módulos de Captura y Supervisión de datos e imágenes que permitan de una forma rápida su instalación y configuración, y así lograr adaptarlo a diferentes sectores.

Teniendo en cuenta lo planteado anteriormente se identifica como **problema de investigación**: ¿Cómo gestionar la captura y supervisión de datos e imágenes durante un proceso de identificación de personas?

Según el problema identificado se define como **objeto de estudio**: Procesos de captura y supervisión de datos e imágenes.

Para darle solución al problema planteado se define como **objetivo general**: Desarrollar los módulos que permitan gestionar la captura y supervisión de datos e imágenes durante el proceso de identificación de personas.

Teniendo en cuenta el objetivo general planteado anteriormente se desglosan los siguientes **objetivos específicos**:

- ❖ Desarrollar el marco teórico de la investigación.
- ❖ Realizar análisis y diseño de los módulos Captura de datos, Captura de imágenes y Supervisión de un sistema de identificación de personas.
- ❖ Implementar los módulos Captura de datos, Captura de imágenes y Supervisión de un sistema de identificación de personas utilizando tecnología *web*.
- ❖ Realizar pruebas a la solución implementada.

Como **campo de acción** se define: Procesos de captura y supervisión de datos e imágenes para la identificación de personas.

Para regir la investigación se define como **idea a defender**: Con el desarrollo de los módulos de Captura de datos, Captura de imágenes y Supervisión se podrá gestionar la captura y supervisión de datos e imágenes en el proceso de identificación de personas.

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

- ❖ Análisis de los principales conceptos asociados a los procesos de identificación de personas para obtener una base teórica para el desarrollo de la solución.
- ❖ Estudio y caracterización de las tendencias y soluciones existentes en el campo de captura y supervisión de datos e imágenes para la identificación de personas.
- ❖ Análisis del sistema *Device Grid Manager (DGM)* para la integración de dispositivos de *hardware* en aplicaciones *web* desarrollado en el Centro de Identificación y Seguridad Digital (CISED).
- ❖ Análisis de los estándares internacionales de información asociados a la identificación de personas.
- ❖ Definición de las herramientas informáticas y metodología a usar para el desarrollo de los módulos Captura de datos, Captura de imágenes y Supervisión para un sistema de identificación de personas utilizando tecnología *web*.
- ❖ Identificación de los procesos de negocio de los módulos Captura de datos, Captura de imágenes y Supervisión.
- ❖ Modelación de los procesos de negocio de los módulos Captura de datos, Captura de imágenes y Supervisión.
- ❖ Descripción de los procesos de negocio de los módulos Captura de datos, Captura de imágenes y Supervisión.
- ❖ Definición de los requisitos funcionales y no funcionales de los módulos Captura de datos, Captura de imágenes y Supervisión.
- ❖ Descripción de los requisitos funcionales de los módulos Captura de datos, Captura de imágenes y Supervisión.
- ❖ Diseño de prototipos de interfaz de usuario utilizando técnicas de usabilidad para la *web*.
- ❖ Diseño de los módulos Captura de datos, Captura de imágenes y Supervisión.

- ❖ Implementación de los módulos que permitan gestionar la captura y supervisión de datos e imágenes.
- ❖ Aplicación de las pruebas de caja negra y pruebas unitarias a los módulos Captura de datos, Captura de imágenes y Supervisión en función de validar la propuesta de solución.

Para realizar las tareas de la investigación se emplearon los siguientes **métodos científicos**:

Métodos teóricos:

- ❖ Histórico- lógico: este método se utiliza para analizar como el desarrollo de las tecnologías ha evolucionado en los sistemas de identificación.
- ❖ Analítico–sintético: se utilizó para 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: se realizó la modelación de los procesos captura de datos, captura de imágenes y supervisión, ya que permite predecir las respuestas de los mismos ante variaciones de algunos de sus parámetros, sin tener que ejecutarlos en la realidad.

Métodos empíricos:

- ❖ Entrevista: se realizaron entrevistas a personas especializadas en el tema con el objetivo de obtener información acerca de los procesos de captura de datos, captura de imágenes y supervisión, para al final lograr un resultado satisfactorio.
- ❖ Observación: con el objetivo de obtener información real de cómo se realizan los procesos de captura de datos, captura de imágenes y supervisión en un sistema de identificación.

Se asume como **justificación de la investigación** que el presente trabajo de investigación es necesario para la Plataforma Modular de Identificación y Control de Acceso, ya que su aporte permitirá obtener la información necesaria y con la calidad requerida para poder emitir documentos de identificación.

El presente documento está estructurado por tres capítulos, los mismos se describen a continuación:

Capítulo 1: Fundamentación teórica: en este capítulo se estudian los conceptos fundamentales relacionados con el trabajo de diploma, además de realizar un estudio de los sistemas de identificación a

nivel nacional e internacional, así como de las herramientas, tecnologías y metodología a utilizar para el desarrollo de la solución.

Capítulo 2: Análisis y diseño de la propuesta de solución: se describen los procesos de negocio, son definidas y descritas detalladamente las funcionalidades de la solución, sirviéndole a los desarrolladores como guía durante la implementación. También son definidos los elementos del diseño con el que contará la propuesta de solución.

Capítulo 3: Implementación y pruebas: se describe el proceso de implementación de la solución. Se detallan las pruebas realizadas al *software* y se exponen sus resultados.

Capítulo 1: Fundamentación teórica

Introducción

Para una mejor comprensión de la presente investigación se hace necesario el estudio de los conceptos fundamentales relacionados con la misma. Un documento de identificación es un elemento en el que constan los datos de una persona para poder identificarla, por lo que es necesario estudiar en el presente trabajo sistemas que automaticen procesos de emisión de documentos de identificación para comprender como se gestiona la captura y supervisión de datos e imágenes para la identificación de personas. Además es fundamental realizar un estudio de las herramientas, tecnologías y metodología a utilizar para el desarrollo de la solución.

Conceptos asociados a los procesos de captura y supervisión de datos e imágenes

A continuación se describen algunos de los conceptos asociados a la presente investigación, la comprensión y el conocimiento de los mismos es de gran importancia para el desarrollo de la propuesta de solución y para el entendimiento de la investigación.

Identificación

La identificación es la acción y efecto de identificar o identificarse. También se conoce como identificación al documento oficial o la credencial que permite identificar a una persona, dicho documento puede ser DNI (Documento Nacional de Identidad), una cédula o un registro, dependiendo del caso (1).

Supervisión

Supervisión es la acción y efecto de supervisar; verbo que supone ejercer la inspección de un trabajo realizado por otra persona. Por lo tanto, la supervisión es el acto de vigilar ciertas actividades de tal manera que se realicen en forma satisfactoria (2).

Captura de datos

A consideración de los autores se entiende como captura de datos, cualquier forma de introducir datos a un sistema informático de forma sistematizada, por lo general para procesarlos y guardarlos.

1.1. Soluciones existentes de sistemas de identificación en el ámbito internacional y nacional

Actualmente en el mundo existen un gran número de sistemas de identificación de personas. Por lo que se hace necesario realizar un estudio de los mismos con el fin de conocer cómo funcionan y analizar si alguno de ellos aporta alguna funcionalidad a la propuesta de solución.

Soluciones existentes de sistemas de identificación en el ámbito internacional

Sistema Automatizado para la Emisión de Pasaporte y Constancia de Cedulación en las Sedes Consulares de la República Bolivariana de Venezuela

Para hacer cumplir el derecho de los venezolanos de poseer un pasaporte para su identificación, permitiendo de esta forma la seguridad y la no falsificación de documentos, se hace necesaria la creación de un Sistema Automatizado para la Emisión del Pasaporte y Constancia de Cedulación en las Sedes Consulares de la República Bolivariana de Venezuela. Donde en el mismo se le captan los datos e imágenes al ciudadano, se supervisan, y se verifica la identidad mediante el sistema AFIS¹ (3).

Las características con que cuenta este sistema en los procesos de captura de datos e imágenes son solamente para la emisión de pasaportes, además está desarrollado para una única entidad, por lo que no será utilizado para la solución.

Sistema Automatizado para la Emisión de Pasaporte Andino de Venezuela

Por la necesidad de que los ciudadanos andinos de la República Bolivariana de Venezuela gozaran de un pasaporte para su identificación, además para evitar el fraude y la falsificación de documentos se crea el Sistema Automatizado para la Emisión de Pasaporte Andino de Venezuela. Este sistema permite realizar la captura de datos a través de una planilla donde se plasman los datos más relevantes de identificación de los ciudadanos, incluyendo la captura de imagen, foto y firma.

- Para la captura de la foto se sigue el siguiente flujo:

El proceso inicia cuando se accede a la opción captar foto y se muestra el área para la captura de la misma, se coloca el área definida sobre la foto, se procede a obtener la imagen y se muestra el área seleccionada.

- El proceso de captura de firma del ciudadano cuenta con el siguiente flujo:

Se accede a la opción captar firma, luego se muestra una opción para seleccionar cuando el ciudadano este imposibilitado para firmar, si es así se selecciona esta opción y luego se inserta el texto "Imposibilitado Firma" en la imagen de la firma del ciudadano, de lo contrario se coloca el área definida sobre la foto, se procede a obtener imagen y finalmente se muestra el área seleccionada (4).

¹ El AFIS (*Automated Fingerprint Identification System*), es un sistema basado en la Identificación Automática de Huellas Dactilares.

Este sistema está desarrollado para una entidad específica, además no se captan los datos de fuentes externas existentes ni se introducen en el momento, por lo que no será utilizado para el desarrollo de la aplicación, ya que lo que se pretende desarrollar es un sistema genérico que pueda ser adaptable y configurable en las instituciones o empresas que requieran del mismo.

Desarrollo de la capa de procesos para el Sistema de Emisión de Pasaportes Diplomáticos, de Servicio y Acreditaciones de la República Bolivariana de Venezuela.

El desarrollo y el gran avance de las tecnologías que existen hoy en día para evitar la falsificación, el fraude y otros aspectos de los documentos de viaje, constituyen el motor impulsor de la creación y evolución de los pasaportes como método de seguridad, es por ello que el Ministerio del Poder Popular para Relaciones Exteriores (MPPRE) de la República Bolivariana de Venezuela y la Universidad de las Ciencias Informáticas, desarrollan el Sistema de Emisión de Pasaportes Diplomáticos, de Servicio y Acreditaciones con el propósito automatizar los procedimientos para emitir pasaportes electrónicos.

En el desarrollo de la capa de Procesos en el SEPYA, que permite la gestión de los procesos Solicitud y Enrolamiento de pasaportes diplomáticos, de servicio de este sistema se realiza la captura de datos, imágenes y la supervisión de la siguiente forma:

- Captura de datos: Si todos los datos son correctos, el captador de datos e imágenes procede a capturar los datos que no fueron registrados como parte de la solicitud.
- Captura de huellas, firma y foto: si la captura de datos se realiza de forma correcta se prosigue a la captura de huellas, firma y foto del ciudadano, según la variante seleccionada: adulto cedula/no cedula, menor cedula con progenitor/tutor/responsable, menor cedula con dos progenitores, menor no cedula con un progenitor/tutor/responsable, menor no cedula con dos progenitores.
- Supervisión de los datos: para la supervisión de los datos el supervisor revisa que los mismos hayan sido captados correctamente, este mismo proceso se realiza para la supervisión de las imágenes (5).

Luego de analizadas las características de este sistema se puede concluir que el estudio realizado sirvió como base para el desarrollo de la solución, ya que aportó nuevos conocimientos acerca de los procesos de captura de datos, captura de imágenes y supervisión, sin embargo no será utilizado para el desarrollo de la misma, ya que está concebido para responder solamente a las necesidades del MPPRE, además no realiza la supervisión de los datos e imágenes captadas.

Sistema de Identificación Civil, Cédula de Identidad y Pasaportes de Chile

Para que cualquier chileno pudiera viajar al exterior necesitaba visa para llegar a distintos destinos, la solicitud de cédula de identidad o pasaporte significaba un trámite complicado, además la pérdida de alguno de estos documentos podía convertirse en una gran pesadilla. Estos problemas desaparecen luego de que el proyecto SONDA se hiciera cargo del Sistema de Identificación Civil, Cédula de Identidad y Pasaportes de Chile que generó enormes beneficios para la comunidad.

Una de las principales innovaciones del proyecto fue la implementación de un sistema computacional central de identificación, que contiene una base de datos de identificación biométrica que almacena imágenes digitalizadas de la fotografía, firma e impresiones dactilares de la persona y además cuenta con nuevas estaciones de trabajo para la captura de datos donde se generan los documentos. El mismo es un sistema centralizado que permite verificar la identidad de un individuo (vivo o fallecido) desde cualquier parte del país de forma inmediata y automatizada, incluso posibilita confirmar la identidad de una persona que no porta su documento de identificación (6).

En este sistema no se realiza la captura de datos a través de fuentes externas, además no se realiza el proceso de supervisión de los datos e imágenes captadas y está desarrollado para satisfacer las necesidades de una única entidad, por lo que solo se estudia para obtener un mayor conocimiento de los procesos de captura de datos e imágenes.

Software de Identificación *Digital – Credcorp*

Este sistema sirve para diseñar identificaciones, vincular el diseño de las mismas a una base de datos con el fin de personalizar cada identificación y finalmente imprimirlas.

Permite el diseño en pantalla con la capacidad de utilizar campos de texto, foto, firma, huella, código de barras, imágenes y campos de textos rotados; con la libertad de relacionar dichos campos con información contenida en una base de datos existente, o bien, obtener la información capturándola en tiempo real y guardándola en una base de datos nueva por cada proyecto. Es compatible con cualquier impresora de tarjetas plásticas o de papel.

Ventajas

- Elabora identificaciones o credenciales en tarjetas plásticas o en papel.
- Genera sus propios diseños acorde a las necesidades.
- Reducción de costos por impresión de credenciales.
- Prestigio a la empresa.

- En caso de no contar con ella, creación de base de datos de empleados.

Características de Credcorp

- Diseño flexible de identificaciones con campos de texto, imágenes y códigos de barra.
- Puede usar la información proveniente de alguna otra fuente de datos como el sistema de recursos humanos o de nómina.
- Permite el uso de múltiples imágenes por registro y de distintos tipos.

Opciones de captura de imagen

- Importación desde archivo.
- Captura de imágenes mediante dispositivos compatibles con *TWAIN*².
- Captura de huellas mediante digitalizador de huella *Sagem MorphoTouch* y próximamente *Digital Persona U. ARE. U. 4000B*.
- Captura de firma mediante digitalizador *Hand HellProducts* y *EpadInk*.
- Interfaz gráfica sencilla al usuario.
- Compatible con *Microsoft Windows 98, Windows NT, Windows 2000 Professional, Windows Me y Windows XP (7)* .

Este sistema no realiza la supervisión de los datos e imágenes, además es un sistema privativo, por lo que no será utilizado para la propuesta de solución.

Soluciones existentes de sistemas de identificación en el ámbito nacional

Sistema Único de Identificación Nacional (SUIN)

El Sistema Único de Identificación Nacional de la República de Cuba (SUIN) es un sistema que permite la inscripción e identificación de las personas naturales. Está concebido con procesos y documentos seguros que incluyen el uso de la biometría; y a partir del mismo se crea la base de datos única de identificación de la población cubana como premisa indispensable para el Gobierno en Línea en el país.

Este sistema se integra a dispositivos para la captura en vivo de imágenes como: fotos, huellas decadaactilares y firma, bajo el cumplimiento de parámetros y normativas de calidad para la verificación

² Estándar destinado a la adquisición de imágenes desde un escáner.

biométrica automatizada. Permite además la captación de datos a partir de la lectura mecánica de los documentos de identidad.

En caso de no contarse con los dispositivos de captura requeridos debe brindarse la posibilidad de lectura de estas imágenes desde un soporte en papel, además de permitir verificaciones contra el CUBAFIS³.

El módulo de Supervisión de este sistema se considera opcional y solo será incluido para aquellos trámites que requieren una valoración de personal especializado o directivos, en función de realizar control de la calidad y de la aprobación del trámite para su conclusión. Puede estar vinculado a la misma oficina o estar en un nivel superior, acorde a los niveles de aprobación de los trámites en cuestión (8).

Este sistema no realiza la captura de datos a través de fuentes de datos existentes, ni el captador introduce los datos en el momento al sistema, sino que se realiza a través de la lectura mecánica de los documentos de identidad, lo que lo convierte en un proceso particular del SUIN, por estas razones no puede ser utilizado en la implementación de la propuesta de solución. Sin embargo el estudio del mismo sirvió de base para conocer cómo se realizan los procesos de captura y supervisión de datos e imágenes para el desarrollo del *software* de la presente investigación.

Sistema de identificación de la UCI

El Sistema de Identificación de la UCI está diseñado para el registro e identificación de las personas que pertenecen a la comunidad universitaria a través de la emisión de credenciales. Este sistema permite la captura de datos e imágenes mediante, base de datos existentes, además admite la captura de fotos en tiempo real con una cámara fotográfica, estas también pueden ser obtenidas desde dispositivos USB como memorias *flash*.

Este sistema no permite que la persona encargada de captar los datos los pueda introducir directamente en el sistema, pues solo los obtiene a través de base de datos existentes. Además no se captan otras imágenes como la huella digital, por lo que el estudio del mismo permitió ampliar el conocimiento sobre las características con que este cuenta para captar los datos e imágenes, pero no será utilizado para la solución, ya que está desarrollado específicamente para la UCI.

³Sistema AFIS desarrollado por el MININT y Datys para Cuba, actualmente llamado BIOMESYS (*Biometric System*).

Sistema para la interacción y control centralizado de dispositivos en aplicaciones web v2.0 (DGM)

El DGM ⁴v2.0 es un sistema que permite la interacción y control centralizado de los dispositivos de captación biométrica. Este sistema cuenta con tres subsistemas que interactúan como un todo: el Servicio local para el manejo de dispositivos, el *framework JavaScript DGMJS* y el subsistema para el control centralizado de dispositivos.

El *framework JavaScript DGMJS* es un componente, que permite la interacción con los servicios publicados en la estación de trabajo donde se esté ejecutando el Servicio local para el manejo de dispositivos. Establece una comunicación entre la aplicación *web* externa y el Servicio local, facilitando interactuar con los dispositivos. A su vez, el Servicio local gestiona las peticiones que provienen desde el navegador *web* que han sido solicitadas por el *Framework DGMJS* y reenvía estas peticiones al motor de manejo de los servicios para cuando se reciba una respuesta por parte del servicio que maneja el dispositivo, enviársela al navegador *web*. El Servicio local es quien contiene además al grupo de clientes y controladores, los clientes constituyen la interfaz para el uso de los dispositivos, quienes utilizan los controladores que son los que interactúan directamente con el dispositivo. Por su parte, el subsistema para el control centralizado de dispositivos brinda una interfaz visual en forma de aplicación *web*, que permite que el usuario pueda gestionar cada uno de los dispositivos de la red. Además de ver en tiempo real el estado de los mismos. Así mismo gestiona las localizaciones de los dispositivos (9).

El DGM v2.0 constituye un sistema que presenta características y funcionalidades que permiten el trabajo con los dispositivos de captura biométrica. Se encarga además, de garantizar su vinculación con los sistemas *web* independiente del navegador a utilizar. Es un sistema de fácil manejo y entendimiento que posibilita llevar a cabo la captura de datos e imágenes para su posterior procesamiento y supervisión. Fue desarrollado en el CISED, por lo que no constituye un *software* de carácter privativo o que tenga que ser costado su empleo para la propuesta de solución.

Por todo lo anteriormente expuesto, se determina utilizar el Sistema para la interacción y control centralizado de dispositivos en aplicaciones *web* v2.0, para llevar a cabo los procesos de captura de datos y de imágenes para su posterior supervisión en la propuesta de solución.

⁴ Por sus siglas en Inglés: *Device Grid Manager*

1.2. Análisis del estándar internacional: Organización de Aviación Civil Internacional (OACI) asociado a los documentos de identificación de las personas

Listado de los datos que debe tener un documento de identificación de tamaño 1 (dv1)

El documento 9303, parte 3, volumen 1 de la OACI⁵, define para los documentos de viajes oficiales de lectura mecánica de tamaño 1 (dv1), las especificaciones de la zona de inspección visual, donde se muestra el listado de datos que debe tener un documento de identificación, que los mismos deben ser claramente legibles. Estos datos se listan en la siguiente tabla (10).

Nro.	Datos	Descripción
1	Estado u organismo expedidor	Estado u organismo responsable de expedir el documento
2	Tipo de Documento	Tipo de documento
3	Nombre	Nombre completo del titular
4	Apellidos	Apellidos del titular
5	Sexo	Sexo del titular, se especificará mediante las iniciales utilizadas comúnmente: F (en caso femenino) y M (en caso masculino).
6	Nacionalidad	Nacionalidad del titular
7	Fecha de nacimiento	Fecha de nacimiento del titular según los registros del Estado u organismo expedidor
8	Datos personales opcionales	Datos personales opcionales ej. : identificación personal o huella digital
9	Número de documento	El que asigne el Estado u organismo expedidor para identificar unívocamente el documento
10	Fecha de caducidad	Fecha de caducidad del documento
11	Firma o marca habitual del titular	Firma o marca habitual del titular
12	Elemento de identificación	Este campo contendrá un retrato del titular
13	Datos opcionales	Datos opcionales adicionales a criterio del Estado u organismo expedidor

Tabla 1. Listado de datos de un documento de identificación de tamaño 1. Fuente: Elaboración propia

⁵ La Organización de Aviación Civil Internacional, también conocida como Organización Internacional de Aeronáutica Civil, OACI (o ICAO, por sus siglas en inglés International Civil Aviation Organization) es una agencia de la Organización de las Naciones Unidas creada en 1944 por la Convención de Chicago para estudiar los problemas de la aviación civil internacional y promover los reglamentos y normas únicos en la aeronáutica mundial.

Para la solución del presente trabajo los datos que se captarán de los que define la OACI para un documento de identificación de tamaño 1 son los siguientes:

- Estado u organismo expedidor.
- Tipo de documento.
- Nombre.
- Apellidos.
- Sexo.
- Nacionalidad.
- Fecha de nacimiento.
- Datos personales opcionales.
- Elemento de identificación.
- Datos opcionales.

Cumplimiento de las normas para los datos biométricos según la OACI para un documento de identificación de tamaño 1 (dv1)

En el documento 9303, parte 3, volumen 1 de la OACI se definen las características que deben tener los datos biométricos exhibidos para un documento de identificación de tamaño1 (dv1), las mismas se mencionan a continuación.

❖ Firma:

La firma exhibida tendrá dimensiones visibles al ojo humano(es decir, la reducción de tamaño será inferior al 50%), conservándose la proporción dimensional de la firma. El color con que se exhiba la firma contrastará claramente con el fondo.

❖ Huella digital de un solo dedo:

La huella digital de un solo dedo exhibida, si lo exige el Estado expedidor, será la original que haya creado el titular en el sustrato del documento, el Estado expedidor hará todo lo posible para asegurar que la huella sea a prueba de falsificaciones y sustituciones. El color con que se exhiba la huella contrastará claramente con el fondo.

❖ Foto:

- Las dimensiones del retrato serán de 35x45mm en ancho y alto, el Estado expedidor ajustará el tamaño adecuado para un dv1.

- La fotografía mostrará una toma de primer plano de la cabeza y los hombros del solicitante, el rostro ocupará hasta el 70%-80% de la dimensión vertical de la fotografía.
- El rostro debe de estar enfocado con total nitidez y no deberán aparecer manchas de tinta o arrugas.
- La fotografía mostrará al solicitante mirando directamente a la cámara.
- La fotografía mostrará al solicitante con los ojos abiertos y claramente visibles sin lentes de color.
- No se permiten tocados, cabello, peinados o adornos faciales que oculten el rostro.
- No se encontrarán otras personas u objetos faciales en la fotografía (11).

Para la solución del software de la presente investigación solo serán captadas las huellas y la foto del ciudadano para la emisión del documento de identificación.

1.3. Tecnologías, metodología y herramientas a utilizar

Las herramientas, tecnologías y metodología a utilizar para el desarrollo de la solución fueron definidas por el equipo de trabajo del proyecto Plataforma Modular de Identificación y Control de Acceso, a continuación se describen las características de las mismas.

1.3.1. Metodología FDD (*Feature Driven Development* / Desarrollo Basado en Funciones)

FDD por sus siglas en inglés *Feature Driven Development* es un enfoque ágil para el desarrollo de sistemas. Dicho enfoque no hace énfasis en la obtención de los requerimientos, sino en cómo se realizan las fases de diseño y construcción. Sin embargo, fue diseñada para trabajar con otras actividades de desarrollo de *software* y no requiere la utilización de ningún modelo de proceso específico. Además, hace énfasis en aspectos de calidad durante todo el proceso e incluye un monitoreo permanente del avance del proyecto. FDD afirma ser conveniente para el desarrollo de sistemas críticos.

Esta metodología propone tener etapas de cierre cada dos semanas, lo cual implica que los desarrolladores tendrán nuevas actividades que realizar en dicho período de tiempo. Esto hace que la motivación del equipo se mantenga durante todo el proyecto debido a que se ven los resultados periódicamente (12).

La metodología FDD establece 5 fases secuenciales:

1. Desarrollo de un modelo general.
2. Construcción de la lista de funcionalidades.
3. Planeación por funcionalidad.
4. Diseño por funcionalidad.
5. Construcción por funcionalidad.

1.3.2. Entorno integrado de desarrollo

Microsoft Visual Studio 2010 Ultimate es el exhaustivo paquete de herramientas de administración del ciclo de vida de las aplicaciones para equipos. Este paquete provee la calidad de los resultados, desde el diseño hasta la implementación. *Visual Studio 2010 Ultimate* admite un número cada vez mayor de plataformas y tecnologías. *Visual Studio Test Professional 2010*, parte de *Visual Studio 2010 Ultimate*, proporciona una herramienta única para capturar y actualizar requisitos de pruebas, automatizar la navegación de pruebas manuales, acelerar la solución y aceptar el ciclo mediante la captura de todo el contexto de la prueba. *Microsoft Visual Studio 2010 Ultimate* incluye potentes herramientas que simplifican todo el proceso de desarrollo de aplicaciones, de principio a fin. Provee una mayor productividad y ahorro de costes al utilizar herramientas de pruebas y depuración integradas que ayudarán a los desarrolladores a crear siempre un código de gran calidad.

Incorpora herramientas avanzadas de pruebas que permiten comprobar la calidad del código en todo momento. Dentro de estas pruebas se encuentran las pruebas de IU codificadas, que automatizan la realización de pruebas de la interfaz de usuario en aplicaciones basadas en *web* y en *Windows*, las pruebas manuales, *Test Professional*, pruebas de rendimiento de *web*, pruebas de carga, entre otras. *Visual Studio 2010* puede instalarse en los sistemas *Windows XP (x86)*, *Windows 7 (x86 y x64)*, entre otros (13).

1.3.3. Lenguaje de desarrollo

CSharp (C #)

C#⁶ es un lenguaje orientado a objetos elegante y con seguridad de tipos que permite a los desarrolladores crear una amplia gama de aplicaciones sólidas y seguras que se ejecutan en *.NET Framework*. Se puede utilizar este lenguaje para crear aplicaciones cliente para *Windows* tradicionales, servicios *Web XML*, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos y muchas tareas más.

La sintaxis de C# simplifica muchas de las complejidades de C++ y proporciona características eficaces tales como tipos de valor que admiten valores *NULL*.

C# admite métodos y tipos genéricos, que proporcionan mayor rendimiento y seguridad de tipos, además admite los conceptos de encapsulación, herencia y polimorfismo.

⁶ Lenguaje de Programación Orientado a Objetos

El proceso de compilación de C# es simple en comparación con el de C y C++ y es más flexible que en Java. No hay archivos de encabezado independientes, ni se requiere que los métodos y los tipos se declaren en un orden determinado (14).

1.3.4. Herramienta de modelado

Visual Paradigm

Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML ideal para Ingenieros de *Software*, Analistas de sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala, necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos (15).

Visual Paradigm for UML (VP-UML) es una herramienta de diseño UML y herramienta CASE UML diseñada para ayudar en el desarrollo del *software*. Es compatible con los equipos de desarrollo de *software* en la captura de requisitos, la planificación del *software*, ingeniería de código, modelos de clase, modelado de datos, etc (16).

1.3.5. Notación de modelado: BPMN

Business Process Model and Notation (BPMN) 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.

BPMN es un estándar internacional de modelado de procesos aceptado por la comunidad, permite modelar los procesos de una manera unificada y estandarizada permitiendo un entendimiento a todas las personas de una organización.

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, quienes diseñan, controlan y gestionan procesos. Dentro de un Diagrama de procesos de negocio BPD se utiliza un conjunto de elementos gráficos, agrupados en categorías, que permite el fácil desarrollo de diagramas simples y de fácil comprensión, pero que a su vez manejan la complejidad inherente a los procesos de negocio (17).

1.3.6. Lenguaje de modelado: UML

El Lenguaje de Modelado Unificado (UML - *Unified Modeling Language*) es un lenguaje que permite modelar, construir y documentar los elementos que forman un producto de software que responde a un

enfoque orientado a objetos. Este lenguaje fue creado por un grupo de estudiosos de la Ingeniería de *Software* formado por: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1995. Desde entonces, se ha convertido en el estándar internacional para definir organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos.

UML no es un lenguaje de programación sino un lenguaje de propósito general para el modelado orientado a objetos y también puede considerarse como un lenguaje de modelado visual que permite una abstracción del sistema y sus componentes (18).

1.3.7. Microsoft .NET Framework 4.0

.NET Framework es un componente integral de *Windows* que admite la compilación y la ejecución de aplicaciones y servicios *Web XML*. El diseño de *.NET Framework* está enfocado a cumplir los objetivos siguientes:

Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local.

Ofrecer un entorno de ejecución de código que promueva la ejecución segura del mismo, incluso del creado por terceras personas desconocidas o que no son de plena confianza.

Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en *Windows* o en *Web*.

Basar toda la comunicación en estándares del sector para asegurar que el código de *.NET Framework* se puede integrar con otros tipos de códigos.

.NET Framework contiene dos componentes principales: *Common Language Runtime* y la biblioteca de clases de *.NET Framework*.

La biblioteca de clases, es una completa colección orientada a objetos de tipos reutilizables que se pueden emplear para desarrollar aplicaciones que abarcan desde las tradicionales herramientas de interfaz gráfica de usuario (GUI) o de línea de comandos, hasta las aplicaciones basadas en las innovaciones más recientes proporcionadas por *ASP.NET*, como los formularios *Web Forms* y los servicios *Web XML* (19).

1.3.8. Gestor de base de datos

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. *PostgreSQL* utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para lograr la estabilidad del sistema.

Sus características técnicas la hacen una de las bases de datos más potente y robusta del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. Funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Es una base de datos 100% ACID además está disponible para *Linux* y *UNIX* en todas sus variantes (*AIX*, *BSD*, *HP-UX*, *SGI IRIX*, *Mac OS X*, *Solaris*, *Tru64*) y *Windows* 32/64bit (20).

1.3.9. Windows Workflow Foundation 3.5

Windows Workflow Foundation es el modelo de programación, motor y herramientas para generar con rapidez las aplicaciones habilitadas por flujo de trabajo en *Windows*. Está compuesto de un espacio de nombres, un motor de flujo de trabajo en proceso y diseñadores para *Visual Studio* 2005.

Windows Workflow Foundation es un marco que permite a los usuarios crear flujos de trabajo de sistema o humanos en sus aplicaciones para *Windows Vista*, *Windows XP* y la familia *Windows Server* 2003. Se puede utilizar para resolver los escenarios simples como mostrar los UI de controles basados en datos proporcionados por el usuario o los escenarios complejos que se producen en las empresas grandes, como procesamiento del orden y control de inventario.

Proporciona una experiencia de desarrollo coherente y familiar con otras tecnologías *.NET Framework* 3.0 como: *Windows Communication Foundation* y *Windows Presentation Foundation*. La API de la *Windows Workflow Foundation* proporciona la compatibilidad completa para *Visual Basic .NET* y *C#*, un compilador del flujo de trabajo especializado, depura dentro de un flujo de trabajo, un *Workflow Designer* gráfico y desarrolla completamente su flujo de trabajo en código o en marcado. También proporciona un modelo extensible y un diseñador para generar actividades personalizadas que encapsulan la funcionalidad del flujo de trabajo para los usuarios finales o para reutilizarse en varios proyectos (21).

1.3.10. Cliente TortoiseSVN-1.7.6

Subversion es un sistema de control de versiones libre/código fuente abierto. Maneja ficheros y directorios a través del tiempo.

El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Esto le permite recuperar versiones antiguas de sus datos, o examinar la historia de cómo han cambiado los mismos.

Subversion puede acceder a su repositorio a través de redes, lo que permite ser usado por las personas desde distintos ordenadores. En cierto modo, la capacidad para que varias personas puedan modificar y

administrar el mismo sistema de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápido sin un único conducto por el cual deban pasar todas las modificaciones. Es un sistema general que puede ser usado para administrar cualquier conjunto de ficheros.

Algunas de las características de *TortoiseSVN* se mencionan a continuación:

Versionado de directorios

CVS solamente sigue la historia de ficheros individuales, pero *Subversion* implementa un sistema de ficheros de versiones “virtual” que sigue los cambios por todo el árbol de directorios sobre el tiempo.

Historial real de versiones

Con *Subversion*, usted puede añadir, borrar, copiar y renombrar ficheros y directorios. Y cada fichero nuevo, agregado comienza con una nueva y limpia historia enteramente suya.

Envíos atómicos

Una colección de modificaciones, cualquiera entra en el repositorio completamente o no del todo.

Esto permite a los desarrolladores construir y enviar los cambios como fragmentos lógicos, y previene problemas que pueden ocurrir cuando solo una porción de cambios son enviados al repositorio satisfactoriamente (22).

1.3.11. ORM NHibernate v3.0

NHibernate es un marco que permite hablar de una base de datos relacional de una manera orientada a objetos. Permite almacenar (o como también se suele decir, "persistir") los objetos de una base de datos y cargarlos más adelante.

NHibernate genera las sentencias SQL necesarias para insertar, actualizar, eliminar y cargar datos. De manera más formal, se puede decir que es un Mapeador Relacional de Objetos (ORM⁷), herramienta o marco. *NHibernate* es un puerto *.NET* de *Hibernate Java*. No es el único marco ORM para *.NET*, existen otros marcos como: *Entity Framework* de *Microsoft*, *LLBLGen Pro*, *Subsonic* y *Genoma*, por nombrar sólo algunos (23).

⁷ Mapeo Objeto Relacional, componente de software que permite trabajar con los datos persistidos como si ellos fueran parte de una base de datos orientada a objetos (en este caso virtual).

1.3.12. BisonFramework

Es un *framework* para la orquestación de procesos de negocio con *Windows WorkFlow Foundation*. Su principal objetivo es proporcionar un componente que permita gestionar las instancias de *WorkFlow*. Además, encapsula un conjunto de actividades y servicios que le dan mayor dinamismo al desarrollo de sistemas centrados en la orquestación de procesos de negocio con WF, específicamente para un ambiente web. Entre los servicios con que cuenta el *Bison* se encuentra el servicio de persistencia, el mismo provee un mecanismo para salvar las instancias de *workflow* en base de datos para luego ser cargadas en memoria cuando sea necesario. Otro de los servicios que contiene este *framework* es el de navegación, tiene la responsabilidad de la navegación dentro de una aplicación *web*, además identifica las actividades que exhiben una interfaz de usuario y las que se encuentran pendientes de ejecución dentro de una instancia *workflow*.

Entre las ventajas que brinda *BisonFramework* se pueden mencionar las siguientes:

- ❖ Proporciona una mayor aproximación a los usuarios de negocio.
- ❖ Brinda rapidez y flexibilidad para modelar y cambiar los procesos según las necesidades.
- ❖ Aporta escalabilidad o capacidad de crecer.
- ❖ Fortifica el puente creado por el *Workflow* para la comunicación entre el analista y el desarrollador.
- ❖ Propone una arquitectura donde se encuentran bien definidas las capas de presentación y negocio, posee actividades y servicios especializados en la orquestación de interfaces de usuario que permiten definir su flujo de una manera gráfica dentro del *Workflow* (24).

1.3.13. ASP.NET

ASP.NET es un modelo de desarrollo *Web* unificado que incluye los servicios necesarios para crear aplicaciones *Web* empresariales con el código mínimo. Forma parte de *.NET Framework* y al codificar las aplicaciones tiene acceso a las clases en *.NET Framework*. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el *Common Language Runtime (CLR)*, entre ellos *Microsoft Visual Basic* y *C#*. Estos lenguajes permiten desarrollar aplicaciones *ASP.NET* que se benefician del *Common Language Runtime* (25).

Las nuevas características de *ASP.NET* ayudan a crear con facilidad páginas que son más coherentes y pueden ofrecer una experiencia más enriquecedora a los usuarios, permiten utilizar las páginas principales para crear un diseño de página coherente para un sitio *Web* o para un grupo de páginas relacionadas (26).

1.3.14. **UIToolsBox**

El *UIToolsBox* es una librería de componentes de interfaz de usuario que rige el modo en que se visualizan los controles en el navegador basado en las siguientes características:

- Define los controles básicos basándose en los controles de *ASP.NET* para mantener el comportamiento de las aplicaciones *ASP.NET*.
- Rige el comportamiento de las interfaces de usuario a partir de una capa basada en *JQuery* que le da un comportamiento más dinámico a los controles utilizados.
- Garantiza peticiones asíncronas al servidor a través de servicios *Web* que son consumidos desde *JavaScript*. (27)

1.3.15. **JavaScript**

JavaScript es un lenguaje de secuencias de comandos desarrollado por *Netscape Communications* y diseñado para desarrollar aplicaciones de *Internet*, cliente y de servidor (28).

El lenguaje *JavaScript* utiliza una sintaxis parecida a la de *C*, y admite construcciones estructuradas, como *if...else*, *for* y *do...while*. Se utilizan llaves (`{}`) para delimitar los bloques de instrucciones. El lenguaje admite varios tipos de datos, como *String*, *Number*, *Boolean*, *Object* y *Array*. Admite las características de fechas mejoradas, las funciones trigonométricas y las expresiones regulares.

JavaScript usa prototipos en vez de clases. Permite definir un objeto creando una función constructora (29).

1.3.16. **CSS**

Las hojas de estilos en cascada (*CSS*) definen reglas de estilo que un programador puede aplicar en los elementos de una página *web*. Por ejemplo, los estilos *CSS* pueden definir cómo se muestran los elementos y dónde se posicionan en la página. Se puede usar *CSS* junto con *ECMAScript* (*JavaScript*, *JScript*) y los elementos *HTML* definidos en la página maestra y las páginas individuales para personalizar la presentación de los sitios *web* (30).

Conclusiones parciales

Para una mejor visión y comprensión de la investigación se han expuestos los principales conceptos relacionados con la misma.

El estudio de sistemas que automatizan procesos de emisión de documentos de identificación que gestionen la captura y supervisión de datos e imágenes en el proceso de identificación de personas sirvió

como base para desarrollar los módulos Captura de datos, Captura de imágenes y Supervisión de la Plataforma Modular de Identificación y Control de Acceso.

Se definió como herramienta de modelado *Visual Paradigm* para diseñar la solución, como entorno integrado de desarrollo *Visual Studio 2010*, como Control de Versiones *Subversion Cliente TortoiseSVN-1.7.6*, como gestor de base de datos *PostgreSQL 9.1*, *Windows Workflow Foundation 3.5*, *Microsoft .NET Framework 4.0*, *ASP.NET* y FDD como metodología de desarrollo de *software*.

Capítulo 2: Análisis y diseño de la propuesta de solución

Introducción

Se realiza el estudio de los procesos captura de datos, captura de imágenes y supervisión dentro del negocio, para luego proceder con la modelación de los mismos. Partiendo del modelado se logran definir los requisitos funcionales que serán automatizados, los mismos se agrupan y clasifican para facilitar el diseño de las funcionalidades del sistema.

2.1. Flujo de procesos

En un sistema de identificación de personas para emitir un documento de identificación se realizan varios procesos, dentro de estos se encuentran: proceso de captura de datos, captura de imágenes y supervisión. A continuación se describe el flujo de proceso captura de datos, los restantes procesos pueden verse en el Anexo 1.

Proceso captura de datos

El proceso inicia cuando el ciudadano solicita el trámite que puede ser para la creación de una nueva credencial o la renovación de esta. El Captador de datos e imágenes verifica si el trámite es por corrección de datos. De no ser por corrección el Captador verifica si los datos se capturan de una fuente externa. Si no se capturan de una fuente externa se busca a la persona en el sistema. De no existir la persona, el sistema muestra un formulario donde serán captados los datos de la misma, el Captador de datos e imágenes introduce los datos correspondientes de la persona y el sistema registra los datos luego de ser introducidos. El Captador de datos e imágenes verifica si existe persona asociada al titular sin captura de datos. Si no existe persona asociada al titular el sistema cambia la fase "Captación de datos" a "Efectiva", si existe persona asociada al titular se realiza el mismo procedimiento anteriormente explicado. Para más detalle del proceso ver la Figura 1 que se encuentra a continuación.

Capítulo 2: Análisis y diseño de la propuesta de solución

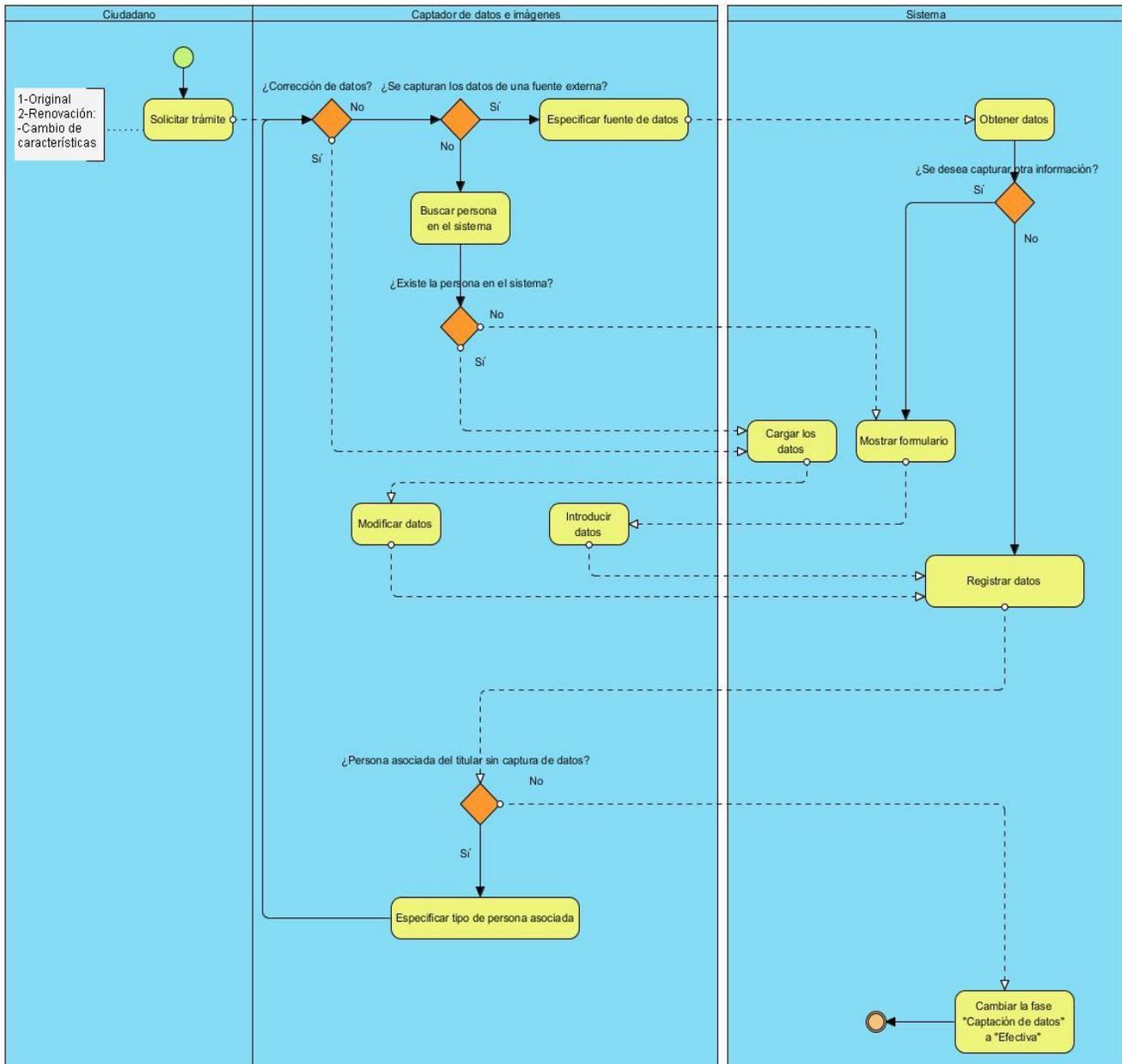


Figura 1: Proceso captura de datos. Fuente: Elaboración propia

2.2. Descripción de las actividades de los procesos captura de datos, captura de imágenes y supervisión

Para comprender con mayor claridad el flujo de acciones que se efectúa en los procesos captura de datos, captura de imágenes y supervisión, se realiza la descripción de las actividades principales que conforman

Capítulo 2: Análisis y diseño de la propuesta de solución

cada uno de los procesos, la misma se muestra a continuación en la Tabla 2. La descripción de todas las actividades se pueden ver en el Anexo 2.

Actividad	Descripción	Responsable	Entrada	Salida
Solicitar trámite	El ciudadano se presenta en la oficina y hace la solicitud para la creación de la credencial.	Ciudadano	Solicitud del ciudadano	Datos de la solicitud
Buscar persona en el sistema	El Captador de datos e imágenes busca a la persona en el sistema para verificar si se le captaron los datos anteriormente.	Captador de datos e imágenes	Datos de la solicitud.	Resultados de la búsqueda.
Mostrar formulario	En caso de que la persona no se encuentre en el sistema, el mismo muestra un formulario para captarle los datos al ciudadano.	Sistema	Resultados de la búsqueda.	Formulario mostrado.
Introducir datos	El Captador de datos e imágenes introduce los datos correspondientes del ciudadano.	Captador de datos e imágenes.	Formulario mostrado.	Datos del ciudadano introducidos
Registrar datos	El sistema registra los datos del ciudadano luego de ser introducidos.	Sistema	Datos del ciudadano introducidos.	Datos del ciudadano registrados en el sistema.
Cambiar la fase "Captación de datos" a "Efectiva"	Cuando los datos son captados y registrados correctamente se cambia la fase "Captación de datos" a "Efectiva".	Sistema	Datos del ciudadano registrados en el sistema.	Estado de la fase "Captación de datos" cambiada a "Efectiva".
Buscar imágenes en el sistema	Se busca si existen imágenes del ciudadano en el sistema para verificar si se le captaron con anterioridad.	Captador de datos e imágenes.	Datos del ciudadano registrados.	Resultados de la búsqueda.
Capturar foto	Se captura la foto del ciudadano a	Sistema	Información de	Foto capturada y

Capítulo 2: Análisis y diseño de la propuesta de solución

	través del dispositivo correspondiente.		que la foto no existe en el sistema.	registrada en el sistema.
Capturar huella	Se captura la huella del ciudadano con el dispositivo para este fin.	Sistema	Información de que no existe la huella en el sistema.	Huella capturada y registrada en el sistema.
Capturar firma	Se captura la firma del ciudadano.	Sistema	Información de que no existe la firma en el sistema.	Firma capturada.
Cambiar estado de la fase "Captura de imágenes" a "Efectiva"	Cuando las imágenes son captadas correctamente la fase "Captura de imágenes" se cambia a "Efectiva".	Sistema	Imágenes captadas.	Estado de la fase "Captura de imágenes" cambiada a "Efectiva".
Seleccionar opción de supervisión	Se selecciona la opción de supervisión luego de que se hayan captado los datos o las imágenes en dependencia de lo que haya sido configurado para la institución.	Supervisor	Datos e imágenes del ciudadano captadas.	Proceso de supervisión iniciado.
Mostrar los datos e imágenes captadas	Se le muestra al ciudadano sus datos e imágenes para que los revise, que puede ser a través de: (planilla de control, revisión desde el monitor, etc.).	Sistema	Datos e imágenes del ciudadano captadas.	Datos e imágenes mostrados por alguna de las opciones.
Revisar datos e imágenes personalmente	El ciudadano revisa sus datos e imágenes personalmente con el fin de detectar algún error.	Ciudadano	Datos e imágenes mostrados por alguna de las opciones.	Datos e imágenes revisadas por el propio ciudadano.
Revisar datos e imágenes	El Supervisor revisa los datos e imágenes del ciudadano luego de	Supervisor	Datos e imágenes mostrados por	Datos e imágenes revisadas por el

Capítulo 2: Análisis y diseño de la propuesta de solución

	pasar por la primera revisión del propio ciudadano.		alguna de las opciones.	Supervisor.
Cambiar estado de la fase "Supervisión" a "Efectiva"	Si luego de la revisión de los datos e imágenes del ciudadano se concluye que toda la información está correcta se cambia la fase "Supervisión" a "Efectiva".	Sistema	Datos e imágenes revisadas.	Estado de la fase "Supervisión" cambiada a "Efectiva".

Tabla 2. Descripción de las actividades principales de los procesos de negocio: Captura de datos, Captura de imágenes y Supervisión. Fuente: Elaboración propia

2.3. Requisitos funcionales

La definición de los requisitos funcionales es una fase que tiene gran importancia en el ciclo de vida del desarrollo del software. Los requisitos funcionales especifican las funcionalidades con que debe cumplir el sistema, por eso la importancia de identificarlos claramente y con una buena calidad, para evitar que tengan que ser definidos nuevamente, provocando un atraso en la entrega del producto e insatisfacción de los usuarios.

Luego del análisis del Modelo de proceso de negocio de la solución, fueron definidas las principales actividades que serán objeto de automatización, las mismas permitirán definir los requerimientos del sistema. Estos requerimientos fueron agrupados según los procesos correspondientes. A continuación son enunciados los requerimientos funcionales identificados:

Requisitos Generales

RF1. Buscar trámite.

1.1 Mostrar criterios para buscar el trámite.

1.2 Mostrar los datos del trámite.

RF2. Listar trámites.

RF3. Buscar información de la persona.

3.1 Mostrar criterios para buscar información de la persona.

3.2 Mostrar la información de la persona.

RF4. Listar personas.

RF5. Mostrar información del ciudadano.

RF6. Obtener datos de una fuente externa.

6.1 Mostrar fuentes de datos existentes.

6.2 Establecer conexión con la fuente de datos.

6.3 Registrar datos obtenidos de la fuente externa.

RF7. Actualizar estado de las fases del trámite.

RF8. Verificar conexión con los dispositivos.

RF9. Emitir notificaciones.

RF10. Cancelar trámite.

Módulo Captura de datos

RF11. Registrar datos del ciudadano.

11.1 Mostrar formulario para introducir los datos del ciudadano.

- Estado u organismo expedidor
- Tipo de Documento
- Nombre
- Primer Apellido
- Segundo Apellido
- Sexo
- Nacionalidad
- Fecha de nacimiento
- Número de documento
- Datos opcionales (Formularios dinámicos)

11.2 Validar los datos introducidos.

11.3 Registrar los datos del ciudadano en el sistema.

RF12. Permitir corrección de datos del ciudadano.

12.1 Cargar datos del ciudadano.

12.2 Señalar los datos incorrectos.

12.3 Actualizar los datos corregidos.

Capítulo 2: Análisis y diseño de la propuesta de solución

Módulo Captura de imágenes

RF13. Permitir la selección de las imágenes existentes que van a utilizarse.

13.1 Registrar imágenes seleccionadas.

RF14. Registrar la foto capturada.

RF15. Registrar huellas capturadas.

Módulo Supervisión

RF16. Imprimir planilla para la supervisión de datos.

RF17. Permitir la selección de los datos e imágenes con errores.

RF18. Registrar los datos e imágenes que fueron seleccionados como incorrectos.

Descripción de las funcionalidades

La metodología de desarrollo **FDD** utilizada para la realización del software, plantea que los requisitos funcionales se deben dividir en subconjuntos en dependencia de la relación que tengan entre ellos. Luego se describen detalladamente para lograr un mejor entendimiento por parte de los desarrolladores a la hora de implementarlos.

En la Tabla 3 se describen detalladamente los requisitos que corresponde al módulo Captura de datos, el resto de las descripciones se pueden ver en el Anexo 3.

Precondiciones	Debe existir una solicitud para realizar un trámite.
Funcionalidades asociadas	RF3, RF4,RF5, RF6,RF9, RF10
Conceptos tratados	Ciudadano, Datos
Descripción básica	1. El sistema muestra la interfaz principal con un menú con la opción “Captación” con la siguientes opciones a seleccionar: <ul style="list-style-type: none">• Captación de datos externos• Captación de datos• Corrección de datos 1.1. Si el Captador de datos e imágenes selecciona la opción

“Captación de datos externos” el sistema muestra la interfaz Captar Datos Externos donde serán cargados los datos del ciudadano.

1.1.1. El Captador especifica la fuente externa, de donde se obtienen los datos del ciudadano mediante un Select, selecciona el criterio para buscar a la persona y presiona el botón Buscar.

1.1.2. El sistema se comunica con la fuente de datos especificada por el Captador de datos e imágenes y muestra la interfaz Captar Datos Externos, donde los datos obtenidos de la fuente externa aparecen previamente cargados en el formulario. Si en la administración se especificó que alguno de los datos obtenidos de la fuente no puede ser modificado, el sistema los carga en el formulario pero de forma no editable.

1.1.3. Si existen datos del formulario que no fueron cargados de la fuente externa el Captador de datos e imágenes los introduce en el sistema.

1.1.4. El Captador de datos e imágenes presiona el botón “Registrar”. (En caso de que se Cancele la acción ver descripción alterna “Cancelar”).

1.1.5. El sistema valida los datos, verificando que no se encuentre ningún campo vacío y que los datos introducidos se correspondan con el formato del campo.

1.1.6. Si todos los datos están correctos y no existen campos obligatorios vacíos, el sistema registra los datos del ciudadano. (En caso de existir campos obligatorios vacíos ver descripción alterna “Campos Vacíos”).

1.1.7. Si el Captador de datos imágenes selecciona la opción “Captación de datos”.

1.1.8. El sistema muestra la interfaz Captar Datos con un formulario para introducir los siguientes datos:

- Estado u organismo expedidor

- Tipo de Ciudadano
- Tipo de Documento
- Primer Nombre
- Segundo Nombre
- Primer Apellido
- Segundo Apellido
- Sexo
- Carnet de identidad
- Nacionalidad
- Fecha de nacimiento
- Datos opcionales(Formularios dinámicos)
 - ✓ Área
 - ✓ Número de documento

1.1.9 El Captador de datos e imágenes introduce los datos del ciudadano y presiona el botón “Registrar”. A continuación se ejecutan los pasos 1.1.5 y 1.1.6 descritos anteriormente.

2. Permitir corrección de datos del ciudadano.

2.1. El sistema muestra una interfaz con el listado de los ciudadanos que se detectaron con problemas en los datos durante la supervisión con las siguientes opciones:

- Corregir datos
- Filtrar

2.1.1. El Captador de datos e imágenes selecciona la opción “Filtrar”.

2.1.1.1. El Captador de datos e imágenes introduce los datos por los que va a filtrar:

- Nombre del titular
- Carnet de Identidad

2.1.1.2. El Captador de datos e imágenes selecciona el botón “Filtrar”.

2.1.1.3. El sistema actualiza el listado según los resultados de la búsqueda. (En caso de no encontrarse la persona ver descripción alterna "Persona no encontrada").

2.1.2. El Captador de datos e imágenes selecciona a la persona del resultado de la búsqueda y presiona la opción Corregir datos, el sistema muestra una interfaz con los datos del ciudadano y señalados con una x en rojo aquellos que se detectaron con problemas durante la supervisión.

2.1.2.1. El Captador de datos e imágenes modifica los datos del ciudadano y selecciona el botón "Actualizar". A continuación se ejecutan los pasos 1.1.5 y 1.1.6 descritos anteriormente.

Prototipos de interfaz de usuario

The screenshot displays a web application interface for a 'SISTEMA DE IDENTIFICACIÓN'. The top navigation bar includes 'Autenticación', 'Administración', 'Captación', 'Supervisión', and 'Entrega'. The main content area is titled 'Captar Datos' and contains a form with the following fields:

Datos Generales			
Estado u Organismo Expedidor	Tipo de Documento	Tipo de Ciudadano	No. Carné
UCI	solapín	estudiante	89040542974
Primer Nombre	Segundo Nombre	Primer Apellido	Segundo Apellido
Mabel	de la Caridad	Ramos	Delis
Fecha de Nacimiento	Sexo	Nacionalidad	
5/4/1989	Femenino	cubana	

Below the 'Datos Generales' section is the 'Datos Opcionales' section, which includes:

Area	No. Documento
F 1	1

At the bottom right of the form are two buttons: 'Registrar' and 'Cancelar'.

Figura 2: Captar datos. Fuente: Elaboración propia

Figura 3: Captar datos de una fuente externa. Fuente: Elaboración propia

Descripción Alternativa

Descripción alternativa 1 "Campos vacíos"

1. Si existen campos obligatorios vacíos el sistema muestra el mensaje "Existen campos vacíos".

Descripción alternativa 2 "Persona no encontrada"

1. Si la persona no existe el sistema muestra un mensaje de error

Descripción alternativa 3 "Cancelar"

1. Si se selecciona la opción Cancelar el sistema redirecciona a la interfaz principal.

Validaciones

1. En el **RF Registrar Datos del ciudadano**, cuando se obtienen los datos de una fuente externa se deben cargar todos los datos que

	<p>se definieron en la administración que deben cargarse de la fuente externa, y en la misma interfaz deben mostrarse los datos opcionales para que el Captador de datos e imágenes los capte.</p> <p>2. En el RF Permitir corrección de datos del ciudadano en caso de que no se corrijan todos los datos especificados cuando el Captador de datos e Imágenes seleccione el botón “Corregir” se debe mostrar un mensaje especificando que deben corregirse todos los datos.</p>
Postcondiciones	Quedan captados los datos de la persona titular del documento de identificación

Tabla 3. Descripción de las funcionalidades del módulo Captura de datos. Fuente: Elaboración propia

2.4. Requisitos no funcionales

Los requisitos no funcionales describen las características del sistema que no implican funcionalidades del mismo, es decir son requisitos que imponen restricciones en el diseño o la implementación del producto. Definen las condiciones que debe tener el sistema para su correcto funcionamiento.

Requisitos no funcionales de *software* para estaciones clientes

RnF1. Sistema Operativo: *Windows XP SP3* o superior con antivirus actualizado.

RnF2. Navegador *web Mozilla Firefox v17.0* o superior, *Google Chrome v20.0* o superior.

RnF3. *Microsoft .Net framework v4.0* o superior.

RnF4. *Device Grid Manager (DGM)* para la interacción y control de dispositivos en aplicaciones web.

Requisito no funcionales de *software* para estaciones servidores

RnF5. Sistema Operativo: *Windows Server 2008 R2* con SP2, antivirus actualizado.

RnF6. *Microsoft .Net framework v4.0. IIS 7.0.*

RnF7. *PostgreSQL 9.1.*

Requisitos no funcionales de *hardware* para estaciones clientes

RnF8. *PC Pentium 4* a *1 GHz* o superior, mínimo *512 MB* de *RAM*, *40 GB* o superior de disco duro.

Capítulo 2: Análisis y diseño de la propuesta de solución

RnF9. Dispositivo para la lectura de huella (Lector de Huellas), dispositivo para la captura de foto (*Webcam*) y dispositivo de impresión (Impresora).

Requisitos no funcionales de *hardware* para estaciones servidores

RnF10. *PC Pentium 4* a 2 GHz o superior, mínimo 2 GB de RAM, 250 GB o superior de disco duro.

RnF11. El servidor para la instalación del Sistema para el control centralizado de dispositivos debe tener como mínimo 4 Gb de RAM, 20 GB de disco duro disponible para el almacenamiento de la base de datos y procesador superior 2.40 Ghz.

Requisitos no funcionales de usabilidad

RnF12. Facilidad de uso por parte de los usuarios: el sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada. Además, debe ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencias una rápida adaptación.

RnF13. 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.

RnF14. El sistema será distribuido en idioma español.

RnF15. Los términos utilizados se establecerán acorde al negocio correspondiente para facilitar la comprensión de la herramienta de trabajo.

RnF16. El sistema poseerá estructura y diseño homogéneos en todas sus pantallas, que facilite la navegación:

- Menús laterales y desplegables que permitan el acceso rápido a la información por parte de los usuarios.

Requisitos no funcionalidades de fiabilidad

RnF17. 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.

RnF18. El tiempo medio entre fallos no debe ser superior a 1 hora.

RnF19. El sistema estará 1 hora fuera de operación luego de haber fallado 3 veces.

Capítulo 2: Análisis y diseño de la propuesta de solución

Requisitos no funcionales de eficiencia

RnF20. El sistema debe ser capaz de soportar una cantidad escalable de peticiones y dar respuestas efectivas y rápidas.

RnF21. El sistema debe hacer un uso eficiente de los recursos de hardware donde se encuentre desplegado.

Restricciones en el diseño y la implementación

RnF22. Plataforma de desarrollo *.NET 4.0* utilizando *Visual Studio Ultimate 2010*.

RnF23. Biblioteca *jQuery*.

RnF24. *Framework Bison* para la orquestación de procesos de negocio con *Windows Workflow Foundation 3.5*.

RnF25. *ASP.NET Web Forms 4.0* para el desarrollo de la capa de presentación.

RnF26. Para el acceso a datos se utilizará el ORM *NHibernate 3.0*.

RnF27. Los artefactos de diseño del *software* se desarrollarán en *Visual Paradigm*.

RnF28. El sistema gestor de bases de datos, será *Postgre SQL 9.1*.

RnF29. El sistema debe implementarse usando el lenguaje *C#*.

RnF30. *Device Grid Manager* para el control e interacción centralizado de dispositivos de *hardware* en aplicaciones *web*.

Requisitos no funcionales de soporte

RnF31. El soporte de la aplicación y los componentes utilizados en el desarrollo del sistema será responsabilidad de los proveedores del mismo.

Requisitos no funcionalidades de seguridad

RnF32. La seguridad se define a nivel de roles, con el fin de mantener la integridad de los datos en función del acceso de cada uno de ellos, trayendo consigo además la protección de la información.

RnF33. Autenticación segura para acceder a la aplicación.

Requisitos no funcionales de interfaz

RnF34. El sistema dispondrá de un diseño sencillo, accesible, ameno e intuitivo acorde a las características de la institución.

Estándares Aplicables

RnF35. ICAO: Estándar Internacional para especificaciones normativas para documentos de identificación.

2.5. Arquitectura propuesta

El diseño de la arquitectura de un sistema es el proceso por el cual se define una solución para los requisitos técnicos y operacionales del mismo. Este proceso define qué componentes forman el sistema, cómo se relacionan entre ellos, y cómo mediante su interacción llevan a cabo la funcionalidad especificada, cumpliendo con los criterios de calidad como seguridad, disponibilidad, eficiencia o usabilidad (31).

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.

La solución se basa en un estilo arquitectónico en capas, que presenta una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver.

Se definió como patrón de arquitectura de *software*, Modelo Vista Controlador (MVC), el cual separa los datos de una aplicación, la interfaz de usuario y la lógica de control. Este patrón de llamada y retorno se ve en aplicaciones *web*, donde las peticiones que realiza el usuario son atendidas por un controlador que desencadena un flujo guiado por un *Workflow*, el cual define las vistas a mostrar.

La arquitectura del sistema se encuentra representada por seis capas lógicas 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 superiores sin realizar grandes cambios en las demás capas inferiores. La comunicación entre las capas se realizara a nivel de interfaces permitiendo trabajar de manera transparente a las instancias reales. En la Figura 4 se muestra la vista lógica de la arquitectura del sistema. Además se realiza la descripción de las seis capas que la componen.

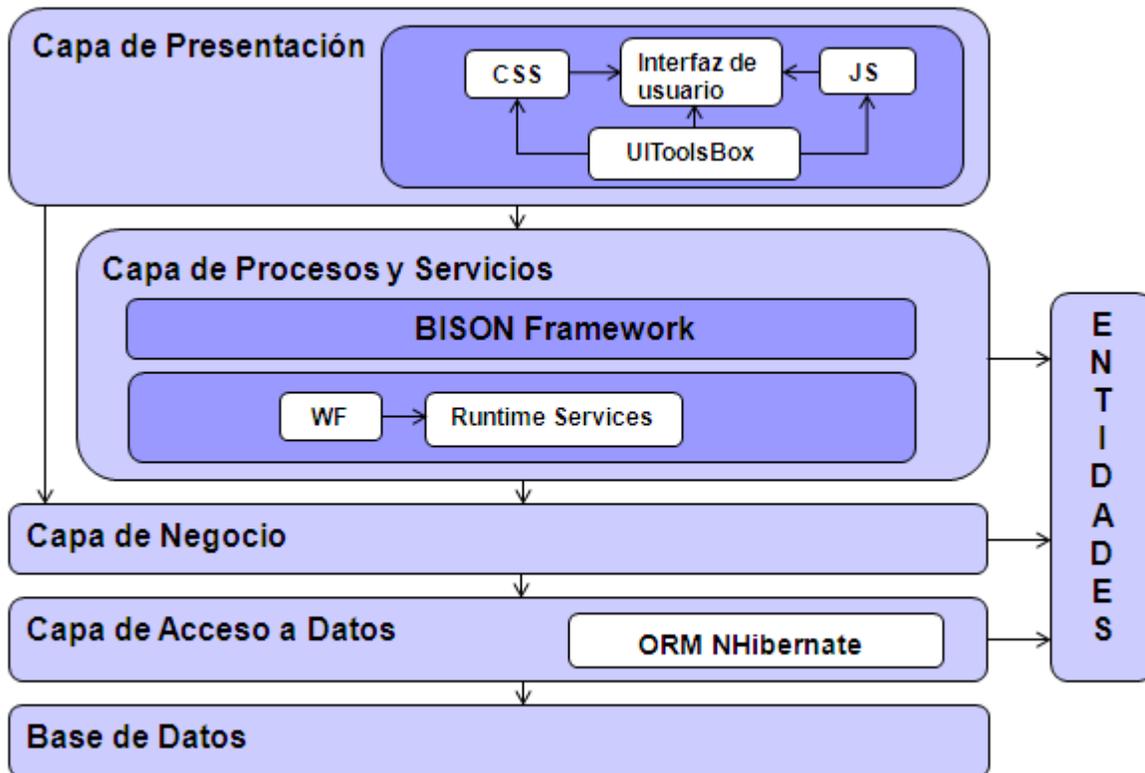


Figura 4: Vista lógica de la arquitectura. Fuente: Elaboración propia

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*, *JQuery*, los componentes *UIToolsBox* y archivos *CSS* que contienen los estilos de la aplicación.

Capa de Procesos y Servicios:

Es la capa donde se modela el negocio mediante procesos, haciendo uso de *Windows Workflow Foundation*. Además se definen los servicios que darán cumplimiento a las actividades del proceso.

Capa de Negocio:

Está conformada por las acciones del negocio que no pueden ser definidas a nivel de *Workflow*. Además mantienen separadas las acciones atómicas del negocio de la definición del proceso.

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. En esta capa se encuentra incluido el ORM *NHibernate*. 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, persisten en la base de datos y la información que estas contienen se muestra en la capa de 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.

2.6. Patrones

❖ Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (32).

A continuación se describen los patrones utilizados durante la implementación de las clases del sistema:

Experto. ¿Quién asumirá la responsabilidad en el caso general?

Asignar una responsabilidad al experto en la información: la clase que posee la información necesaria para cumplir con la responsabilidad (33).

Ejemplo: La clase *wrkfTrámite* es la clase que maneja todo el proceso de captura y supervisión de datos e imágenes a través de *workflow*, ya que es la clase experta para cumplir con estas responsabilidades.

Creador. ¿Quién crea?

Asigna a la clase *BussinesFactory* la responsabilidad de crear una instancia de la clase Ciudadano.

Ejemplo:

```
var ciudadano = new Ciudadano
{
    CIdentidad = result.CarneIDentidad,
    Fnacimiento = result.Fechanacimiento,
    IdCiudadano = result.IdCiudadano,
    Pnombre = result.Pnombre,
    Snombre = result.Snombre,
    Papellido = result.Papellido,
    Sapellido = result.Sapellido,
    Sexo = result.Sexo,
    tipoCiudadano = null
};
```

Figura 5: Ejemplo del uso del patrón Creador. Fuente: Elaboración propia

❖ Patrones de creación

Los patrones de diseño de creación abstraen el proceso de creación de instancia. Ayudan a hacer a un sistema independiente de cómo se crea, se compone y se representan sus objetos. Un patrón de creación de clases usa la herencia para cambiar la clase de la instancia a crear, mientras que un patrón de creación de objetos delega la creación de la instancia en otro objeto. Estos patrones encapsulan el conocimiento sobre las clases concretas que usa el sistema y ocultan como se crean y se asocian las instancias de estas clases (34).

A continuación se describe el patrón de diseño de creación utilizado para la implementación de las clases del sistema.

Singleton: Determina que una clase solo tenga una instancia y proporciona un punto de acceso global a ella.

Ejemplo:

```
public class BusinessFactory
{
    private static BusinessFactory instance = null;
    public static BusinessFactory Instance
    {
        get
        {
            if (BusinessFactory.instance == null)
                BusinessFactory.instance = new BusinessFactory();

            return BusinessFactory.instance;
        }
    }
}
```

Figura 6: Ejemplo del uso del patrón Singleton. Fuente: Elaboración propia

❖ Patrones *workflow*

En un *workflow*, los patrones son la secuencia lógica de diagramación de un requisito del negocio. Estos patrones se encuentran agrupados de la siguiente manera:

1. Patrones básicos de control de flujo.
2. Patrones de sincronización y enrutamiento avanzado.
3. Patrones estructurales.
4. Patrones que involucran múltiples instancias.
5. Patrones que se basan en el estado del sistema.
6. Patrones de cancelación.

A continuación se muestran los patrones utilizados para el diseño del *workflow* de la solución:

Patrones básicos de control de flujo: Están presentes en la mayoría de los lenguajes de *workflow*, y sirven para modelar procesos secuenciales, paralelos o aquellos que incluyan alguna decisión.

Patrones estructurales: Permiten terminar un subproceso cuando ya no haya nada que hacer o permiten definir ciclos de forma arbitraria.

2.7. Diseño de las funcionalidades

Luego de realizar la descripción de las funcionalidades se realizan los diagramas de clases del diseño donde se identifican las clases y se especifican los métodos que serán implementados en las mismas.

A continuación se muestra el diagrama de clases correspondiente a la funcionalidad “Registrar datos del ciudadano” del módulo Captura de datos. En el Anexo 4 se muestran otros diagramas de clases.

Diagrama de clases del diseño: Registrar datos del ciudadano

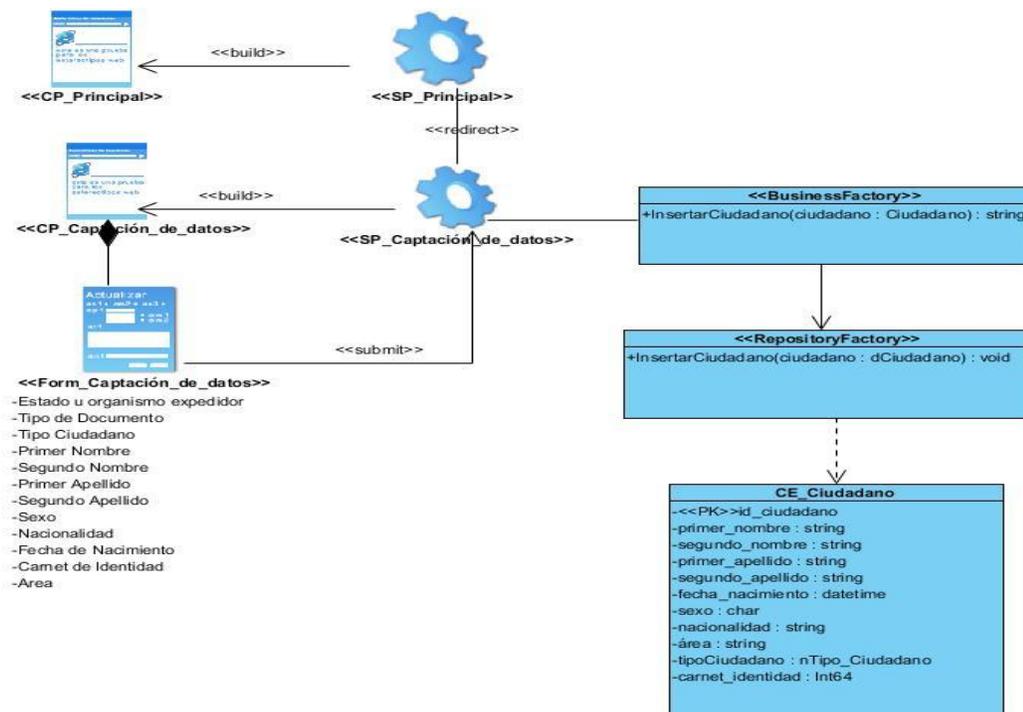


Figura 7: Diagrama de clases del diseño: Registrar datos del ciudadano. Fuente: Elaboración propia

Capítulo 2: Análisis y diseño de la propuesta de solución

Descripción de las clases principales

Nombre	<i>BussinesFactory</i>	
Descripción	Es la clase donde se encuentra el método que permite insertar al ciudadano en la base de datos luego de realizado el trámite.	
Métodos	Parámetros	Descripción
<i>string</i> InsertarCiudadano	Ciudadano ciudadano	Adiciona un ciudadano en la base de datos luego de realizado el trámite.

Tabla 4. Descripción de la clase BussinesFactory. Fuente: Elaboración propia

Nombre	<i>RepositoryFactory</i>	
Descripción	Es la clase que se encarga de la conexión a la base de datos	
Parámetros	Métodos	Descripción
dCiudadano ciudadano	void InsertarCiudadano	Inserta un ciudadano

Tabla 5. Descripción de la clase RepositoryFactory. Fuente: Elaboración propia

Descripción de las entidades

A continuación se muestra la descripción de la entidad CE_Ciudadano para ver la descripción de las restantes entidades ver Anexo 5.

Nombre	CE_Ciudadano	
Descripción	Representa la entidad que contiene los datos del ciudadano	
Nombre del atributo	Tipo de dato	Descripción
id_ciudadano	uuid	Identificador del ciudadano
primer_nombre	varchar	Primer nombre del ciudadano
segundo_nombre	varchar	Segundo nombre del ciudadano
primer_apellido	varchar	Primer Apellido del ciudadano
segundo_apellido	varchar	Segundo Apellido del ciudadano
Sexo	char	Sexo del ciudadano
fecha_Nacimiento	timestamp	Fecha de nacimiento del ciudadano

Capítulo 2: Análisis y diseño de la propuesta de solución

Nacionalidad	varchar	Nacionalidad del ciudadano
carnet_identidad	Int64	Carnet de identidad del ciudadano
id_tipo_ciudadano	serial	Identificador tipo de ciudadano
id_tipo_dato_opcional	serial	Identificador tipo dato opcional
Área	varchar	Área a la que pertenece el ciudadano

Tabla 6. Descripción de la entidad CE_Ciudadano. Fuente: Elaboración propia

Diagramas de secuencias

Un diagrama de secuencia es una forma de diagrama de interacción, que muestra los objetos como líneas de vida a lo largo de la página, con sus interacciones en el tiempo, representadas con mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes arrojan esas comunicaciones (35). A continuación se muestra el diagrama de secuencia de la funcionalidad “Registrar datos del ciudadano” del módulo Captura de datos, el resto de los diagramas se encuentran registrados en la documentación oficial del proyecto Plataforma Modular de Identificación y Control de Acceso.

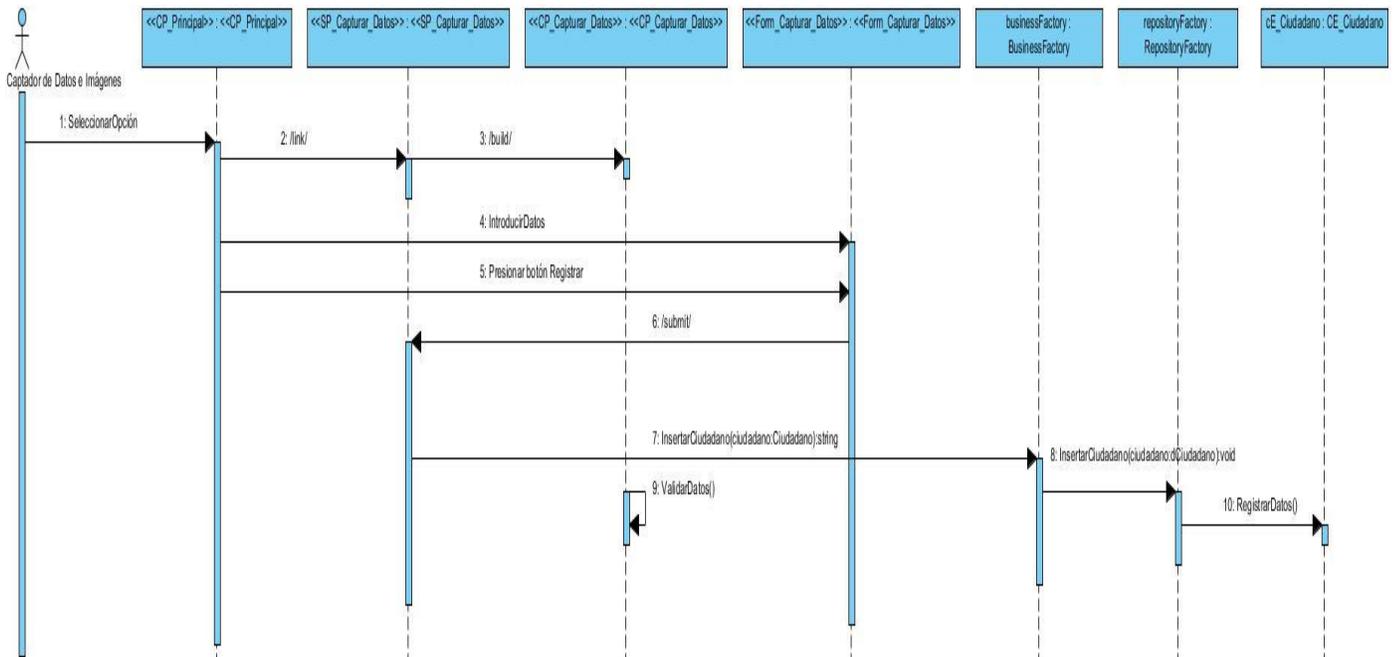


Figura 8: Diagrama de secuencia: Registrar datos del ciudadano. Fuente: Elaboración propia

2.8. Diseño del *workflow* utilizado en la propuesta de solución

Los flujos de trabajo (*workflow*) modelan los procesos de negocio identificados para el desarrollo de la aplicación. La solución está conformada por un *workflow* que maneja la captura de los datos de un ciudadano ya sea desde una fuente externa o introduciendo los mismos en el sistema directamente, también maneja la captura de las imágenes de un ciudadano. Estas imágenes al igual que los datos pueden ser captadas desde una fuente externa o en el momento. Además este flujo de trabajo realiza la supervisión de los datos e imágenes que fueron captadas al ciudadano. En la Figura 9 se muestra una parte del diseño del *workflow*. Para ver el *workflow* con más detalle ver Anexo 6.

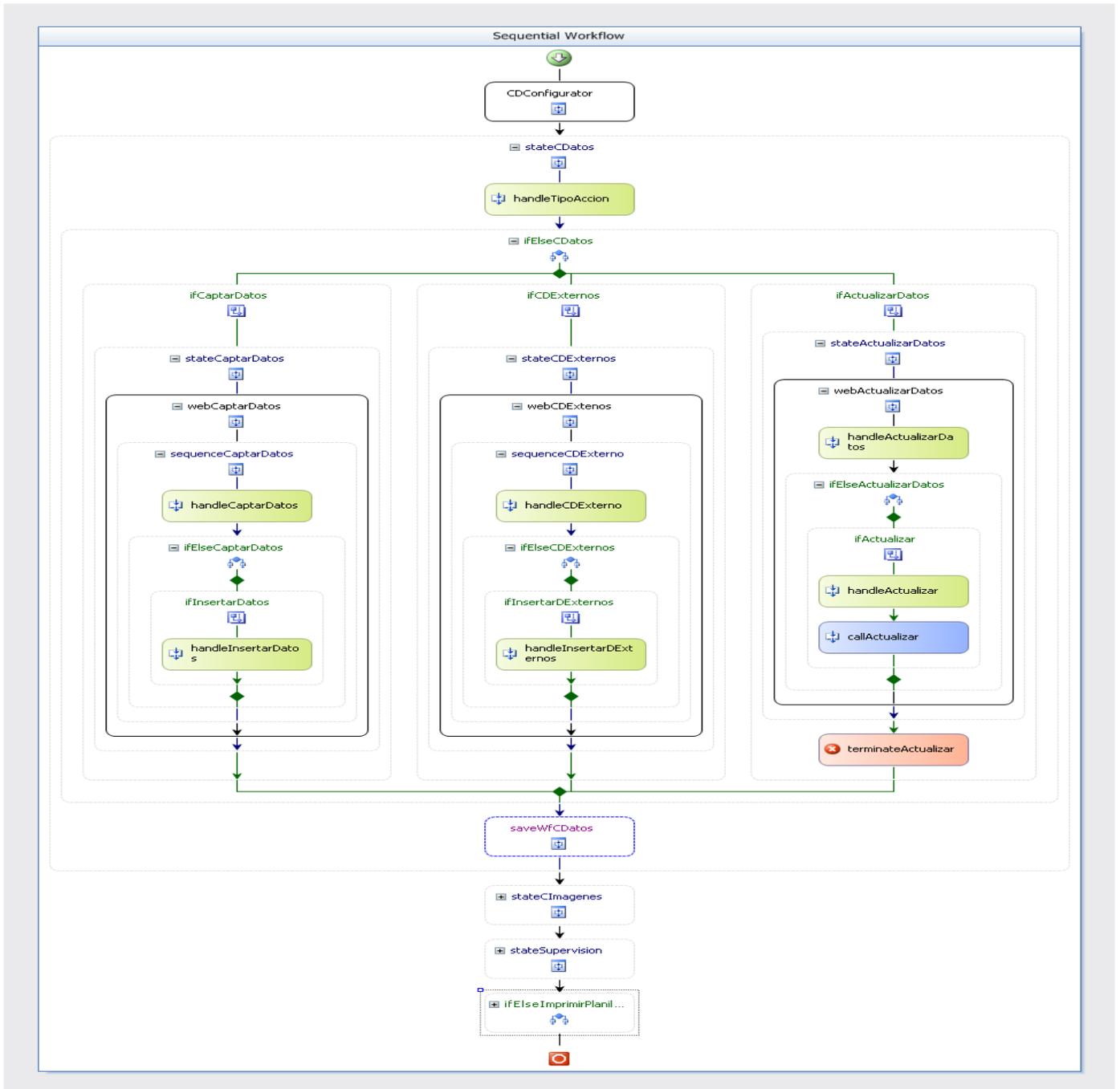


Figura 9: Workflow para la captura de los datos de un ciudadano. Fuente: Elaboración propia

2.9. Modelo de datos

La utilización de un modelo de datos tiene lugar durante el proceso de diseño de una base de datos. El mismo constituye la base del modelo de datos final de la aplicación, en el cual se definen detalladamente todas las entidades persistentes, así como los atributos y relaciones entre ellas.

A continuación se muestra el modelo de datos del sistema. Este esquema está compuesto por las entidades que almacenan los datos e imágenes captados a un ciudadano, los mismos constituyen la entrada fundamental para poder emitir un documento de identificación. También se encuentran las entidades que almacenan los datos opcionales de un ciudadano, estos datos son definidos por cada institución. Dentro de estas entidades se encuentra el nomenclador **ntipo_datos_opcional** que representa el nombre del conjunto de datos opcionales que se le captaran en este caso al ciudadano, la entidad **datributo_datos_opcional** almacena el atributo del dato opcional, que se refiere al nombre del mismo, también se encuentra la entidad **dvalor_datos_opcional**, la misma es la que almacena el valor correspondiente a cada atributo, esta entidad tiene la particularidad de ir creciendo a medida que se van añadiendo los valores para cada ciudadano. Para ver el modelo de datos más detallado consultar el Anexo 7.

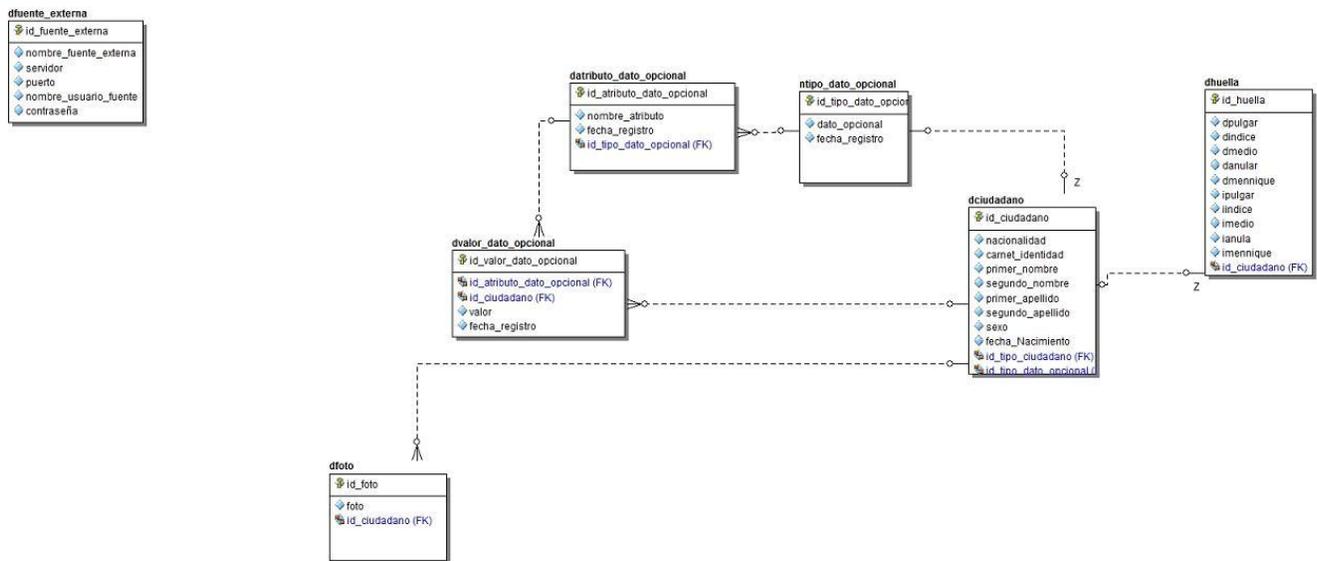


Figura 10: Modelo de datos. Fuente: Elaboración propia

Capítulo 2: Análisis y diseño de la propuesta de solución

Conclusiones parciales

El análisis del flujo de los procesos captura de datos, captura de imágenes y supervisión permitió una mayor visión de las actividades a automatizar.

La descripción de las actividades de los procesos captura de datos, captura de imágenes y supervisión, permitió definir las entradas y salidas de las mismas, así como los roles involucrados en cada momento del flujo.

La creación del listado de requisitos permitió definir las funcionalidades a automatizar para la solución del *software* de la presente investigación.

La descripción de las funcionalidades permitió realizar los diagramas de clases del diseño correspondientes a cada funcionalidad del sistema.

Se definió el modelo de datos, donde se identificaron las entidades persistentes que contienen la información que se procesa en el sistema, especificándose sus atributos y las relaciones entre ellas.

Capítulo 3: Implementación y pruebas

Introducción

La implementación y prueba constituyen una de las acciones más importantes para el desarrollo de un producto de *software*. El proceso de implementación incluye el establecimiento de los estándares de codificación, el desarrollo de cada una de las interfaces, entre otros elementos. Por otra parte las pruebas permiten detectar los errores cometidos durante el desarrollo del producto.

3.1. Estándares de codificación

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad, es de gran importancia para el *software*, logrando además un buen rendimiento del mismo (36).

El uso de estándares de codificación mejora la interpretación del código de programación entre los integrantes del equipo de desarrollo. A continuación se reflejan los estándares definidos para la propuesta solución.

- ❖ **Convención Pascal:** El primer caracter de cada palabra es en mayúscula y el resto en minúscula.
- ❖ **Convención Camel:** El primer caracter de cada palabra es en mayúscula (excepto la primera palabra) y el resto en minúscula.
- ❖ **Convención Pascal para el nombre de las clases**

Ejemplo:

```
public class BusinessFactory
{
    private static BusinessFactory instance = null;
    public static BusinessFactory Instance
    {
        get
        {
            if (BusinessFactory.instance == null)
                BusinessFactory.instance = new BusinessFactory();

            return BusinessFactory.instance;
        }
    }
}
```

Figura 11: Estándar de codificación: Convención Pascal para el nombre de las clases. Fuente: Elaboración propia

❖ Prefijo “I” con la convención Camel para las interfaces

Ejemplo:

(Ejemplo: `ITramiteService`)

Figura 12: Estándar de codificación: Prefijo “I” con la convención Camel para las interfaces. Fuente: Elaboración propia

❖ Convención Pascal para el nombre de los métodos

Ejemplo:

```
public string InsertarCiudadano(Ciudadano ciudadano)
{
    try
    {
        var tciudadano = new nTipoCiudadano
        {
            IdTipoCiudadano = ciudadano.tipoCiudadano.IdTipoCiudadano,
            Clasificacion = ciudadano.tipoCiudadano.Clasificacion
        };
    }
}
```

Figura 13: Estándar de codificación: Convención Pascal para el nombre de los métodos. Fuente: Elaboración propia

❖ Variables: Usar el significado de las palabras para el nombre de las variables, no usar abreviaturas.

Bien:

```
string nombre
char sexo
```

Figura 14: Estándar de codificación: Nombre de las variables. Fuente: Elaboración propia

Mal:

```
string nam
char sex
```

Figura 15: Estándar de codificación: Nombre de las variables. Fuente: Elaboración propia

❖ Llaves de apertura y cierre

- Las llaves deben ser utilizadas sobre una nueva línea y no sobre la misma línea, como en if, for, etc.

Ejemplo:

```
if (fpcHuella.RightHand.Amputated && fpcHuella.LeftHand.Amputated)
{
    var emptyHuella = new Huella {
        ciudadano=ciudadano.dCiudadano,
        idHuella=SessionInfo.InstanceId
    };
};
```

Figura 16: Estándar de codificación: Llaves y líneas en blanco. Fuente: Elaboración propia

❖ 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.

Ejemplo:

```
public void InsertarFoto(dFoto foto)
{
    try
    {
        UnitOfWork unidadtrabajo = null;
        Repository<Guid, dFoto> repositorio;

        unidadtrabajo = new UnitOfWork(NHibernateSessionManager.Instance.GetSessi
        repositorio = new Repository<Guid, dFoto>(unidadtrabajo.Session);

        repositorio.Add(foto);
        unidadtrabajo.Commit();
    }

    catch (Exception ex)
    {
        throw ex;
    }
}
```

Figura 17: Estándar de codificación: Buenas prácticas de programación. Fuente: Elaboración propia

3.2. Tratamiento de errores

Para mantener un correcto funcionamiento, lograr una mayor integridad y confiabilidad de la información que se procesa en el sistema y para evitar errores que no se conozcan siquiera sus orígenes, se realiza el tratamiento de errores partiendo de diferentes acciones las cuales se mencionan a continuación:

- El tratamiento de las excepciones se realiza a través de las sintaxis `try {...} catch (Exception ex) {...}`.
- 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.
- Validación de los datos introducidos verificando que los mismos se correspondan con el formato del campo.

3.3. Vista de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de *hardware* (nodos) y muestra cómo los elementos y artefactos del *software* se trazan en esos nodos (37). Para la propuesta de solución se estableció el diagrama de despliegue representado en la Figura 18, el cual está compuesto por una computadora cliente en la cual estará instalado el Servicio Local del sistema *Device Grid Manager (DGM)* para la interacción y control de dispositivos en aplicaciones *web*. Además esta PC tendrá instalado *Microsoft .NET Framework 4.0*, la misma contendrá un dispositivo para la lectura de huellas (Lector de Huellas), otro para la captura de fotos (*Webcam*), y una impresora. La comunicación de estos dispositivos será a través de puerto USB, la comunicación de la PC con el Servidor *Web* será a través del protocolo HTTPS.

Se representan además dos servidores, el primero de ellos es el Servidor *Web* que tendrá instalado *Internet Information Service 7.0* y *Microsoft .Net framework v4.0*. El segundo servidor contendrá la base de datos del sistema y deberá poseer como gestor de base de datos *PostgreSQL 9.1*. La comunicación entre dichos servidores será a través de *TCP/IP*.

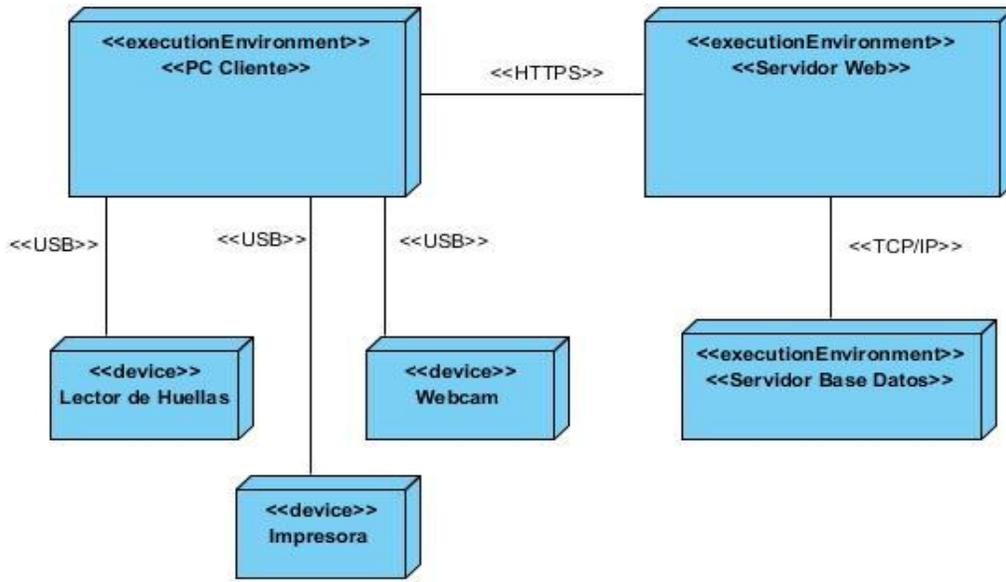


Figura 18: Diagrama de despliegue. Fuente: Elaboración propia

3.4. Diagrama de componentes

Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces (38).

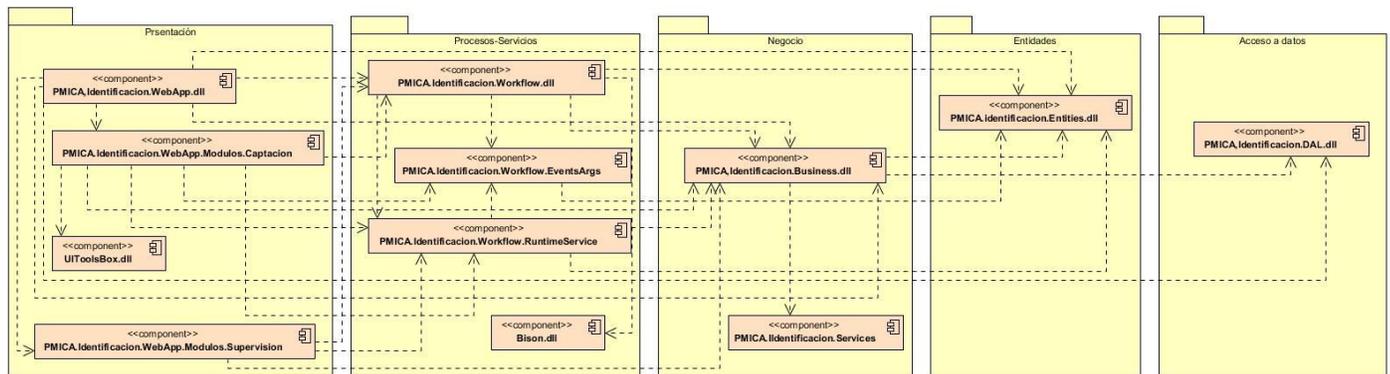


Figura 19 : Diagrama de componentes. Fuente: Elaboración propia

Descripción del Diagrama de componentes

Los componentes para la solución del presente trabajo fueron desarrollados cumpliendo con la arquitectura propuesta por el proyecto Plataforma Modular de Identificación y Control de Acceso. La capa de Presentación contiene el componente **PMICA.Identificación.WebApp.dll** este representa la Aplicación web donde se encuentran los **UserControls**, ficheros ascx que contienen los componentes gráficos y los formularios de las páginas web, además contiene el componente **UIToolsBox.dll** que posibilita la inclusión de controles web en las páginas que conforman las vistas, esta capa se comunica con la capa de Procesos y Servicios donde se encuentran los componentes **PMICA.Identificacion.Workflow.EventsArgs**, **PMICA.Identificacion.Workflow.RuntimeService**, **Bison.dll** y **PMICA.Identificacion.Workflow.dll**, en este último se encuentra el *workflow* que modela el negocio a través de procesos.

El componente **PMICA.Identificacion.Workflow.EventsArgs** contiene los eventos que permiten el funcionamiento del *workflow*, que este es llamado en el componente **PMICA.Identificacion.Workflow.RuntimeService**. En la capa de negocio se encuentra el componente **PMICA.Identificacion.Business.dll** que es donde se realizan las acciones del negocio que no pueden ser definidas a nivel de *workflow*. El componente **PMICA.Identificacion.Entities.dll** perteneciente a la capa Entidades contiene las clases entidades que persisten en la base de datos y la información que estas contienen fluye a través de las capas de Presentación, Procesos y Servicios y Negocio. Para el acceso a los datos se encuentra el componente **PMICA.Identificacion.DAL.dll** que se encarga de operar con la información persistente y enviarla a las clases entidades.

3.5. Interfaces de usuarios

La aceptación final de un producto de *software* por parte del usuario depende en gran medida de la apreciación que este se lleve del sistema, esto se logra a través de las interfaces, el diseño de las mismas debe ser sencillo, accesible, con colores adecuados, simple de usar, que le permita al usuario interactuar de forma intuitiva y sin dificultad con la aplicación. A continuación se muestra la interfaz asociada a la funcionalidad “Registrar datos del ciudadano” del módulo Captura de datos.

SISTEMA DE IDENTIFICACIÓN Bienvenido: Danny Hernández Simons

Captar Datos

Datos Generales

Estado u Organismo Expedidor	Tipo de Documento	Tipo de Ciudadano	No. Carné
UCI	solapín	estudiante	89040542974
Primer Nombre	Segundo Nombre	Primer Apellido	Segundo Apellido
Mabel	de la Caridad	Ramos	Delis
Fecha de Nacimiento	Sexo	Nacionalidad	
5/4/1989	Femenino	cubana	

Datos Opcionales

Area	No. Documento
F1	1

Registrar Cancelar

Figura 20: Interfaz Registrar datos del ciudadano. Fuente: Elaboración propia

3.6. Pruebas

Las pruebas de *software* son los procesos que permiten verificar y revelar la calidad de un producto antes de su puesta en marcha. Básicamente, es una fase en el desarrollo del *software* que consiste en probar las aplicaciones construidas.

Las pruebas se integran dentro de las diferentes fases del ciclo de vida del *software*. En este sentido, se ejecuta el aplicativo a probar y mediante técnicas experimentales se trata de descubrir qué errores tiene.

Según IEEE y SWEBOK una prueba es una actividad en la que un sistema o un componente es ejecutado bajo condiciones especificadas, los resultados son observados o registrados, realizándose una evaluación del aspecto del sistema o componente.

Una prueba es una actividad ejecutada para evaluar y mejorar la calidad del producto a través de la identificación de defectos y problemas (39).

Pruebas unitarias

Las pruebas unitarias constituyen el primer paso para detectar errores en el código, pues se centran en la menor unidad de diseño del *software*: el módulo, por ejemplo un método de una clase o una clase. El objetivo principal de estas pruebas es detectar errores en cada uno de los módulos del *software* al ser ejecutados independientemente del resto de los componentes (40).

Las pruebas unitarias permiten verificar a través de la ejecución de un fragmento de código si la aplicación cumple con las especificaciones esperadas. En la Figura 21 se muestra la prueba realizada a la funcionalidad “InsertarCiudadano (Ciudadano ciudadano)” y el resultado arrojado por la misma.

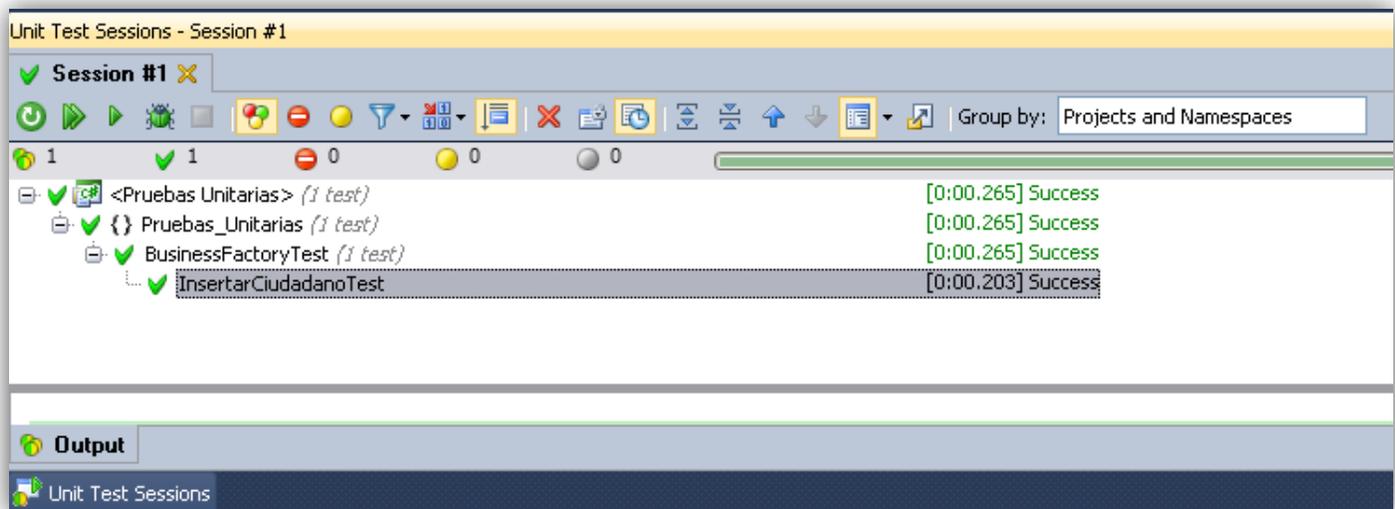


Figura 21: Resultado de la prueba unitaria realizada al método InsertarCiudadano. Fuente: Elaboración propia

En la figura anterior se muestran las pruebas ejecutadas que permiten comprobar que los datos del ciudadano son registrados correctamente a través de la clase *BusinessFactoryTest*. El icono verde que se encuentra al lado del nombre de la prueba significa que se completó satisfactoriamente, en caso de que hubiese fallado se mostraría en rojo. La barra de progreso en verde confirma que la prueba se ejecutó satisfactoriamente.

Pruebas de caja negra

Las pruebas de caja negra, se realizan dándole distintos valores a las entradas. Los datos de prueba se escogerán atendiendo a las especificaciones del problema, a fin de verificar que el programa corra bien.

Este tipo de prueba se centra en los requisitos funcionales del *software* y permite obtener entradas que prueben todos los flujos de una funcionalidad (41).

Para la aplicación de las pruebas de caja negra se desarrollaron los casos de prueba para cada funcionalidad del sistema. Estos están constituidos por un conjunto de variables a través de las cuales se determina si el requisito de una aplicación es parcial o completamente satisfactorio. Los casos de prueba, fueron realizados y se encuentran registrados en la documentación oficial del proyecto Plataforma Modular de Identificación y Control de Acceso.

Resultado de las pruebas

Pruebas unitarias

Durante la ejecución de las pruebas unitarias para la primera, segunda y tercera iteración de desarrollo, se realizaron tres iteraciones de prueba.

En la siguiente tabla se muestran los resultados alcanzados de las pruebas unitarias realizadas durante las tres etapas de desarrollo.

Iteraciones	Funcionalidades		
	Funcionalidades con errores	Funcionalidades correctas	Funcionalidades no implementadas
1	1	1	1
1,1	0	2	1
1,2	0	3	0
2	1	1	1
2,1	2	1	0
2,2	0	3	0
3	1	0	2
3,1	1	1	1
3,2	0	3	0

Tabla 7. Resultados de las pruebas unitarias. Fuente: Elaboración propia

En la siguiente figura se muestra de forma gráfica los resultados especificados anteriormente

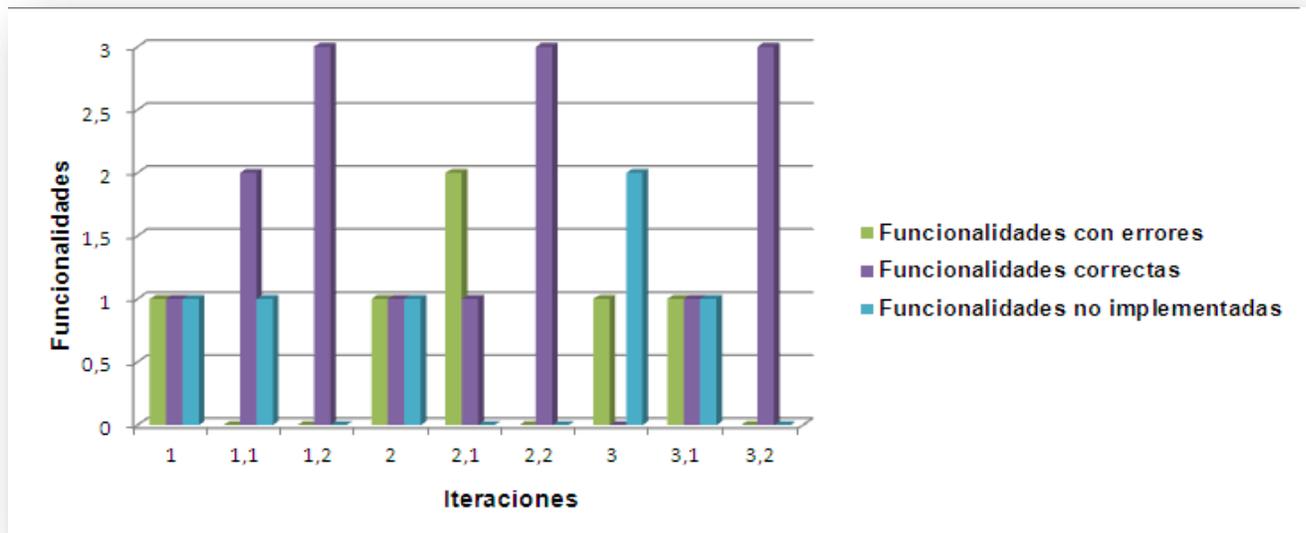


Figura 22: Gráfico del resultado de las pruebas unitarias. Fuente: Elaboración propia

Las pruebas unitarias realizadas permitieron comprobar el correcto funcionamiento de las funcionalidades definidas en cada uno de los módulos de la aplicación, demostrando que la misma cumple con las especificaciones esperadas.

Pruebas de caja negra

Las pruebas de caja negra fueron realizadas luego de la implementación de los módulos Captura de datos, Captura de imágenes y Supervisión. Para cada módulo se realizaron cuatro iteraciones de prueba. A continuación se muestran los resultados arrojados por la misma.

Iteraciones	Módulos		
	Captación de datos	Captación de imágenes	Supervisión
1	10	15	20
2	5	10	10
3	3	2	5
4	0	0 </td <td>0</td>	0

Tabla 8: Resultados de las pruebas de caja negra. Fuente: Elaboración propia

En la siguiente figura se muestra de forma gráfica los resultados especificados anteriormente

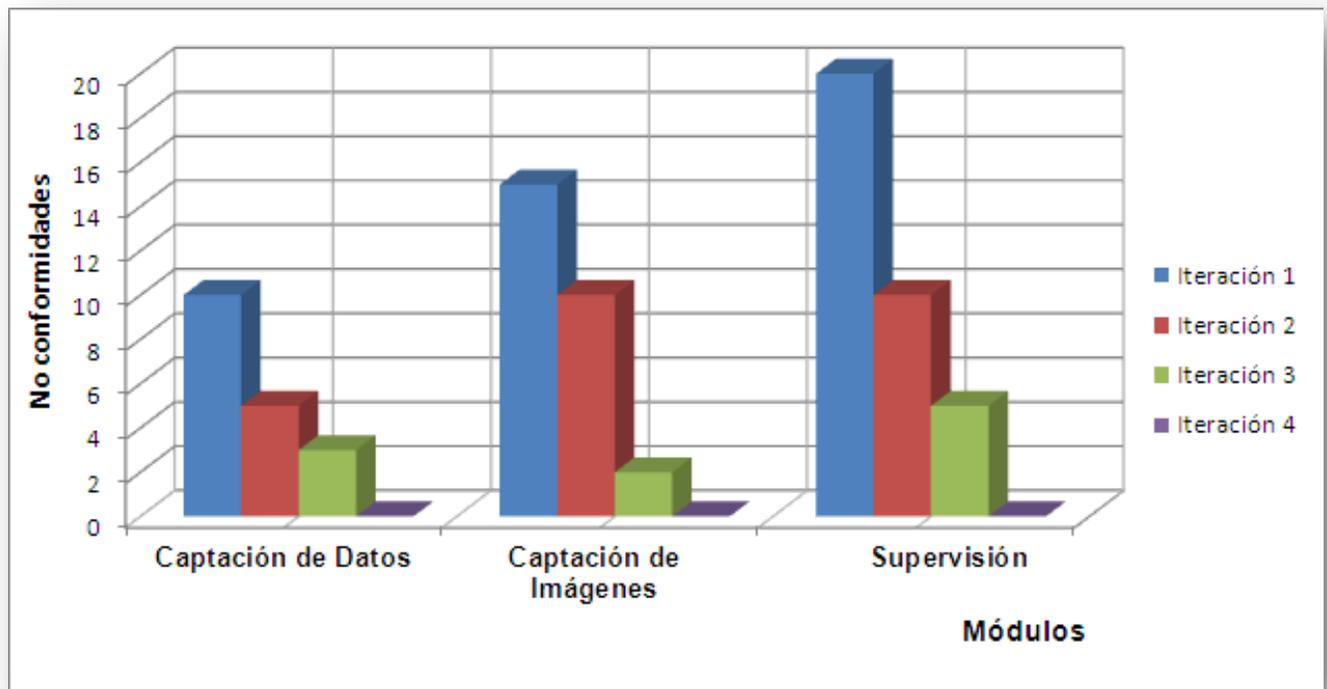


Figura 23: Gráfico del resultado de las pruebas de caja negra. Fuente: Elaboración propia

Las pruebas de caja negra realizadas permitieron comprobar que la aplicación cumple con los requisitos definidos.

Con los resultados obtenidos queda demostrado el correcto funcionamiento de los módulos Captura de datos, Captura de imágenes y Supervisión para la Plataforma Modular de Identificación y Control de Acceso.

Conclusiones parciales

Se generó el diagrama de despliegue, donde se indican los elementos de *hardware* con que cuenta la aplicación y los protocolos de comunicación que se establecen entre ellos.

Con la implementación de los módulos Captura de datos, Captura de imágenes y Supervisión, se cumplió con los requisitos definidos para la solución del *software*.

Las pruebas unitarias y de caja negra realizadas a la aplicación, permitieron comprobar su correcto funcionamiento y la evaluación de la calidad del *software*.

Conclusiones

Como resultado de la investigación realizada se arriba a las siguientes conclusiones:

- ❖ De los sistemas estudiados, fue utilizado en la solución del *software* el sistema *Device Grid Manager (DGM)*, para el manejo y control de dispositivos de *hardware*.
- ❖ La modelación y descripción detallada de los procesos captura de datos, captura de imágenes y supervisión, facilitó identificar las entradas y salidas de todas las actividades presentes en los mismos, así como los roles involucrados en cada momento del flujo.
- ❖ La identificación de las actividades que conforman el objeto de automatización, permitió definir los requisitos funcionales del sistema.
- ❖ La implementación de los módulos Captura de datos, Captura de imágenes y Supervisión, permitió el cumplimiento de los requisitos definidos para la solución del *software*.
- ❖ Durante el desarrollo de la aplicación, se realizaron pruebas unitarias y de caja negra que permitieron comprobar el correcto funcionamiento de la misma.
- ❖ La solución obtenida permitió que en la Plataforma Modular de Identificación y Control de Acceso, se gestione la captura de datos e imágenes que constituyen elementos fundamentales para poder emitir un documento de identificación, realizándose además la supervisión para verificar que los mismos se hayan captado correctamente.

Recomendaciones

- ❖ Se recomienda para futuras versiones del sistema implementar para el módulo Captura de imágenes la funcionalidad que permita la captura de firma.

Bibliografía referenciada

1. **Larousse Editorial, SL** . Definición.de. [En línea] <http://definicion.de/identificacion/>.
2. **Definicion.de**. Definición.DE. [En línea] 2008. [Citado el: 26 de febrero de 2013.] <http://definicion.de/supervision/>.
3. **Dayana Hernández Sayú, Yonnys Rodríguez Hernández**. *Sistema automatizado para la emisión del Pasaporte y Constancia de Cedulación en las Sedes Consulares de la República Bolivariana de Venezuela*. 2008.
4. **Membribes, Yamilka Aviles**. *Sistema Automatizado para la Emisión de Pasaporte Andino de Venezuela*. 2006.
5. **Idia Herrera Rivero, Annie Cubas González**. *Desarrollo de la capa de Procesos para el Sistema de Emisión de Pasaportes Diplomáticos, de Servicio y Acreditaciones de la República Bolivariana de Venezuela*. 2011.
6. **SONDA S.A.** SONDA. [En línea] 2012. [Citado el: 9 de enero de 2013.] <http://www.sonda.com/caso/13/>.
7. Identificación Digital@LONEXI. [En línea] [Citado el: 23 de noviembre de 2012.] http://www.lonexi.com/identificacion_digital.html.
8. **Arza, Javier**. *Proyecto Técnico Modernización del Sistema Nacional de Identificación de Cuba*. 24 de octubre de 2012.
9. **Rodriguez, Sandy Campos, Hector Luis Rodríguez Sánchez**. *Sistema para la interacción y control centralizado de dispositivos en aplicaciones web v2.0*. La Habana : s.n., 2012.
10. **ICAO**. *Documentos de viaje de lectura mecánica*. 2008.
11. —. *Documentos de viaje de lectura mecánica*. 2008.
12. **Universidad ORT Uruguay, Catedra de Ingeniería de Software**. *Metodología FDD*. Uruguay : s.n.
13. **OSC Digital**. Intercambios Virtuales. [En línea] 2013. [Citado el: 18 de mayo de 2013.] www.intercambiosvirtuales.org/software/microsoft-visual-studio-2010-ultimate-espanol.
14. **Microsoft**. MSDN. [En línea] 2013. <http://msdn.microsoft.com/>.
15. **Targetware Informática S.A.C.** software.com.ar. [En línea] 2012. [Citado el: 19 de mayo de 2013.] <http://www.software.com.ar/visual-paradigm-para-uml.html>.
16. **Visual Paradigm**. Visual Paradigm. [En línea] 19 de mayo de 2013. <http://www.visual-paradigm.com/product/vpuml/>.
17. **bizagi**. *Bizagi Process Modeler*.

18. **eumednet (SEJ-309)**. eumed.net. [En línea] [Citado el: 1 de marzo de 2013.] <http://www.eumed.net/libros-gratis/2009c/587/Lenguaje%20de%20Modelado%20Unificado.htm>.
19. **Microsoft**. MSDN. [En línea] [Citado el: 19 de mayo de 2013.] <http://msdn.microsoft.com/es-es/library/zw4w595w%28v=vs.100%29.aspx>.
20. Sobre PostgreSQL-es. [En línea] [Citado el: 9 de diciembre de 2012.] http://www.postgresql.org.es/sobre_postgresql.
21. **Microsoft**. MSDN. [En línea] 2013. [Citado el: 2 de marzo de 2013.] <http://msdn.microsoft.com/>.
22. **Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato**. *Control de versiones con Subversion*.
23. **Schenker, Dr. Gabriel Nicolas, Aaron Cure**. *NHibernate 3 Beginners Guide*. 2011.
24. **Linnett Galens Ameneiro, Danner Sierra Obregón**. *Sistema de Personalización de Documentos de Identificación de la República de Cuba*. La Habana : s.n., 2010.
25. **Microsoft**. MSDN. [En línea] [Citado el: 19 de mayo de 2013.] <http://msdn.microsoft.com/es-es/library/4w3ex9c2.aspx>.
26. —. MSDN. [En línea] [Citado el: 19 de mayo de 2013.] <http://msdn.microsoft.com/es-es/library/fa1h9d0d%28v=VS.80%29.aspx>.
27. **Cesar, Ing. Ivan Campos**. *Módulo de análisis de trazas de procesos del Sistema Único de Identificación Nacional*. La Habana : s.n., 2012.
28. **Microsoft**. Soporte. [En línea] 2013. [Citado el: 19 de mayo de 2013.] <http://support.microsoft.com/kb/154585/es>.
29. —. MSDN. [En línea] 2013. [Citado el: 19 de mayo de 2013.] <http://msdn.microsoft.com/es-es/library/ie/6974wx4d%28v=vs.94%29.aspx>.
30. —. MSDN. [En línea] 2013. [Citado el: 19 de mayo de 2013.] <http://msdn.microsoft.com/es-es/library/ms438349%28v=office.14%29.aspx>.
31. **Cesar de la Torre Llorente, Unai Zorrilla Castro, Javier Calvarro Nelson, Miguel Ángel Ramos Barroso**. *Guía de Arquitectura N-Capas orientada al Dominio con .Net 4.0*. 2010.
32. **Larman, Craig**. *UML y Patrones*.
33. —. *UML y Patrones*.

34. **Erich Gamma, Richard Helm, Raph Johnson, John Vlissides.** *Patrones de Diseño. Elementos de software orientado a objetos reutilizables* . 2003.
35. **Ltd, Sparx Systems Pty.** Sparx Systems. [En línea] 2007. <http://www.sparxsystems.com.ar/about.html>.
36. MSDN. [En línea] <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
37. **S.A, SOLUS.** Sparx Systems. [En línea] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.
38. **Microsoft.** MSDN. *MSDN*. [En línea] [Citado el: 5 de mayo de 2013.] <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.
39. **CIBERTEC.** *Pruebas de Software*.
40. **Javier Tuya, Isabel Ramos Roman, Javier Dolado Cosín.** *Técnica Cuantitativa para la gestión en la ingeniería del software*. 2007.
41. **CIBERTEC.** *Pruebas de Software*.

Bibliografía consultada

Dayana Hernández Sayú, Yonnys Rodríguez Hernández. *Sistema automatizado para la emisión del Pasaporte y Constancia de Cedulación en las Sedes Consulares de la República Bolivariana de Venezuela.* 2008.

Membrives, Yamilka Aviles. *Sistema Automatizado para la Emisión de Pasaporte Andino de Venezuela.* 2006.

Idia Herrera Rivero, Annie Cubas González. *Desarrollo de la capa de Procesos para el Sistema de Emisión de Pasaportes Diplomáticos, de Servicio y Acreditaciones de la República Bolivariana de Venezuela.* 2011.

Arza, Javier. *Proyecto Técnico Modernización del Sistema Nacional de Identificación de Cuba.* 24 de octubre de 2012.

Rodriguez, Sandy Campos, Hector Luis Rodríguez Sánchez. *Sistema para la interacción y control centralizado de dispositivos en aplicaciones web v2.0.* La Habana : s.n., 2012.

ICAO. *Documentos de viaje de lectura mecánica.* 2008.

Targetware Informática S.A.C. software.com.ar. [En línea] 2012. [Citado el: 19 de mayo de 2013.] <http://www.software.com.ar/visual-paradigm-para-uml.html>.

Visual Paradigm. Visual Paradigm. [En línea] 19 de mayo de 2013. <http://www.visual-paradigm.com/product/vpuml/>.

Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato. *Control de versiones con Subversion.*

Schenker, Dr. Gabriel Nicolas, Aaron Cure. *NHibernate 3 Beginners Guide.* 2011.

Linnett Galens Ameneiro, Dannier Sierra Obregón. *Sistema de Personalización de Documentos de Identificación de la República de Cuba.* La Habana : s.n., 2010.

Cesar, Ing. Ivan Campos. *Módulo de análisis de trazas de procesos del Sistema Único de Identificación Nacional.* La Habana : s.n., 2012.

Cesar de la Torre Llorente, Unai Zorrilla Castro, Javier Calvarro Nelson, Miguel Ángel Ramos Barroso. *Guía de Arquitectura N-Capas orientada al Dominio con .Net 4.0.* 2010.

Larman, Craig. *UML y Patrones.*

Erich Gamma, Richard Helm, Raph Johnson, John Vlissides. *Patrones de Diseño. Elementos de software orientado a objetos reutilizables .* 2003.

CIBERTEC. *Pruebas de Software.*

Javier Tuya, Isabel Ramos Roman, Javier Dolado Cosín. *Técnica Cuantitativa para la gestión en la ingeniería del software.* 2007.

TICS. TICS- Tecnologías de la Información y las Comunicaciones. *TICS- Tecnologías de la Información y las Comunicaciones.* [En línea] 2011. [Citado el: 19 de mayo de 2013.] <http://www.tics.org.ar/home/index.php/noticias-destacadas-2/157-definicion-de-tics>.

Glosario de términos

Captación de datos: Es la acción de introducir datos en una aplicación, es decir, el momento en que se introduce información en el sistema se denomina proceso de captación, y esta información será procesada por dicho sistema posteriormente.

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).

Fuente externa: Lugar de donde se obtendrán los datos a utilizar en el sistema. Puede ser una base de datos, fichero, servicio web, etc.

Huella dactilar: Señal que deja en una superficie la yema de un dedo.

Identificación: Reconocimiento de la identidad de un individuo. Documento de identidad.

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

Planilla de control: Inventario que se llena mientras se lleva a cabo una supervisión.

Supervisión: Proceso de inspección del desarrollo de una o varias actividades por parte de personas autorizadas.

Supervisor: Persona encargada de la dirección y vigilancia del desarrollo de una actividad.

Anexos

Anexo 1. Flujo de los procesos captura de imágenes y supervisión

1.1. Proceso captura de imágenes

El proceso inicia cuando el Captador de datos e imágenes busca si existen imágenes del ciudadano en el sistema captadas con anterioridad. Si no existen imágenes, se verifica si se obtienen de una fuente externa. Si no se obtienen de una fuente externa, se verifica si se desea captar la foto. Si se desea captar la foto el sistema la capta a través del dispositivo correspondiente. Si se desea captar la huella el sistema capta la huella del ciudadano. Si se desea captar la firma el sistema capta la firma con el dispositivo para realizar dicha función. El Captador de datos e imágenes verifica si existe persona asociada al titular sin captura de imágenes. Si no existe persona asociada al titular el sistema cambia la fase "Captura de imágenes" a "Efectiva" si existe persona asociada al titular se realiza el mismo procedimiento mencionado anteriormente. Para más detalle del proceso ver la Figura 1 que se encuentra a continuación.

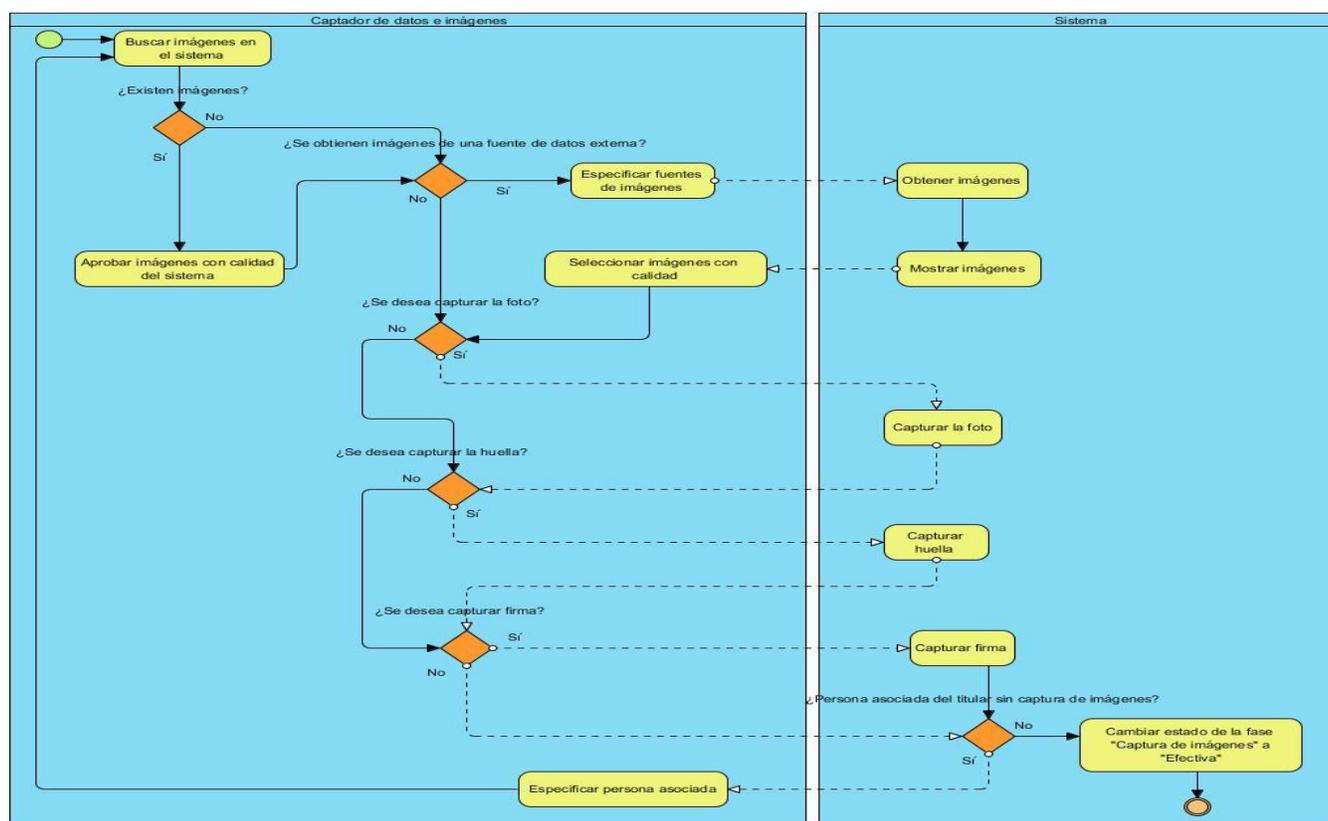


Figura 1: Proceso captura de imágenes

1.2. Proceso supervisión

El proceso inicia cuando el Supervisor selecciona la opción Supervisión luego de que se hayan captado los datos y las imágenes en dependencia de lo que haya sido configurado para la institución. El sistema le muestra al ciudadano sus datos e imágenes para que los revise (planilla de control, revisión desde el monitor, etc.). El ciudadano revisa sus datos e imágenes personalmente con el fin de detectar algún error. El Supervisor revisa los datos e imágenes luego de pasar por la primera revisión del propio ciudadano, verifica que los datos e imágenes estén correctos, si los datos e imágenes están correctos se cambia la fase "Supervisión" a "Efectiva". De lo contrario el Supervisor selecciona los datos e imágenes con problemas y cambia la fase "Supervisión" a "No Efectiva". Para más detalle del proceso ver la Figura 2.

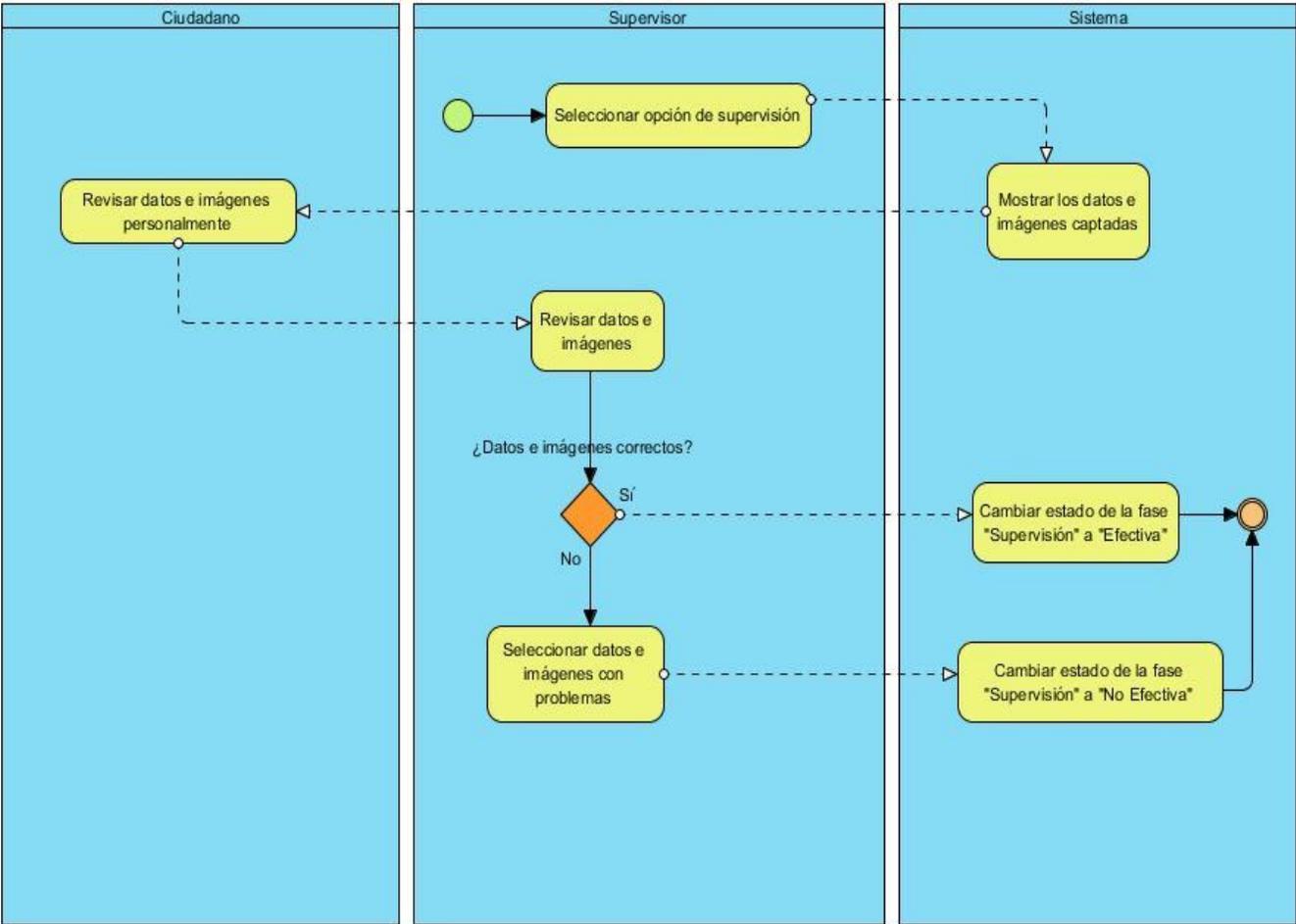


Figura 2: Proceso supervisión

Anexo 2 Descripción de todas las actividades de los procesos captura de datos, captura de imágenes y supervisión

Actividad	Descripción	Responsable	Entrada	Salida
Solicitar trámite	El ciudadano se presenta en la oficina y hace la solicitud para la captura de datos e imágenes.	Ciudadano	Solicitud del ciudadano	Datos de la solicitud
Buscar persona en el sistema	El Captador de datos e imágenes busca a la persona en el sistema para verificar si se le captaron los datos anteriormente.	Captador de datos e imágenes	Datos de la solicitud.	Resultados de la búsqueda.
Mostrar formulario	En caso de no haberse encontrado a la persona en el sistema, se muestra el formulario para la captura de datos del ciudadano.	Sistema	Resultados de la búsqueda.	Formulario mostrado.
Introducir datos	El Captador de datos e imágenes introduce los datos correspondientes del ciudadano.	Captador de datos e imágenes.	Formulario mostrado.	Datos del ciudadano introducidos
Registrar datos	El sistema registra los datos del ciudadano después que son introducidos en el sistema.	Sistema	Datos del ciudadano introducidos.	Datos del ciudadano registrados en el sistema.
Cambiar la fase "Captación de datos" a "Efectiva"	Cuando los datos son captados y registrados correctamente se cambia la fase "Captación de datos" a "Efectiva".	Sistema	Datos del ciudadano registrados en el sistema.	Estado de la fase "Captación de datos" cambiada a "Efectiva".

Buscar imágenes en el sistema	Se buscan las imágenes en el sistema para verificar si se le captaron con anterioridad al ciudadano.	Captador de datos e imágenes.	Datos del ciudadano registrados.	Resultados de la búsqueda.
Captar la foto	Se capta la foto del ciudadano a través del dispositivo correspondiente.	Sistema	Información de que las imágenes no existen en el sistema.	Foto captada y registrada en el sistema.
Captar huella	Se capta la huella del ciudadano.	Sistema	Información de que no existe la huella en el sistema.	Huella captada.
Captar firma	Se capta la firma del ciudadano.	Sistema	Información de que no existe la firma en el sistema.	Firma captada.
Cambiar estado de la fase "Captura de imágenes" a "Efectiva"	Cuando las imágenes son captadas correctamente la fase "Captura de imágenes" se cambia a "Efectiva".	Sistema	Imágenes captadas.	Fase del trámite "Captura de imágenes" cambiada a "Efectiva".
Seleccionar opción de supervisión	Se selecciona la opción de supervisión luego de que se hayan captado los datos o las imágenes en dependencia de lo que haya sido configurado para la institución.	Supervisor	Datos e imágenes del ciudadano captadas.	Proceso de supervisión iniciado.
Mostrar los datos e imágenes	Se le muestra al ciudadano sus datos e imágenes para que los revise (planilla de	Sistema	Datos e imágenes del ciudadano captadas.	Datos e imágenes mostrados por alguna de las opciones.

captadas	control, revisión desde el monitor, etc.).			
Revisar datos e imágenes personalmente	El ciudadano revisa sus datos e imágenes personalmente con el fin de detectar algún error.	Ciudadano	Datos e imágenes mostrados por alguna de las opciones.	Datos e imágenes revisadas por el propio ciudadano.
Revisar datos e imágenes	El Supervisor revisa los datos e imágenes del ciudadano luego de pasar por la primera revisión del propio ciudadano.	Supervisor	Datos e imágenes mostrados por alguna de las opciones.	Datos e imágenes revisadas por el supervisor.
Cambiar estado de la fase "Supervisión" a "Efectiva"	Si luego de la revisión de los datos e imágenes del ciudadano se concluye que toda la información está correcta se cambia la fase "Supervisión" a "Efectiva".	Sistema	Datos e imágenes revisadas.	Fase del trámite "Supervisión" cambiada a "Efectiva".
Especificar fuente de datos	El Captador de datos e imágenes especifica cuál de las fuentes de datos que están configuradas en el sistema va a utilizar para obtener la información del ciudadano.	Captador de datos e imágenes	Datos de la solicitud	Identificada la fuente de datos externa a utilizar para obtener los datos básicos del ciudadano
Obtener datos	El sistema se comunica con la fuente de datos externa especificada y guarda los datos obtenidos en la BD.	Sistema	Identificada la fuente de datos externa a utilizar para obtener los datos básicos del ciudadano	La información obtenida de la fuente de datos externa registrada en el sistema
Cargar los datos	Si la información del ciudadano se encuentra en el sistema se cargan sus datos en el formulario correspondiente	Sistema	Resultados de la búsqueda	Datos cargados en el formulario

Modificar datos	Se modifican los datos en el caso que lo requiera	Captador de datos e imágenes	Datos cargados en el formulario	Datos modificados
Especificar tipo de persona asociada	Cuando el titular tiene personas asociadas se especifica el tipo de persona asociada en cada caso	Captador de datos e imágenes	Datos del titular registrados en el sistema	Tipo de persona asociada especificada
Aprobar imágenes con calidad del sistema	Si las imágenes se encuentran en el sistema el Captador de Datos e Imágenes aprueba si las mismas tienen la calidad requerida	Captador de datos e imágenes	Resultados de la búsqueda	Imágenes aprobadas
Especificar fuentes de imágenes	El Captador de datos e imágenes especifica cuál de las fuentes de datos que están configuradas en el sistema va a utilizar para obtener las imágenes del ciudadano	Captador de datos e imágenes	Datos registrados del ciudadano	Identificada la fuente de datos externa a utilizar para obtener los datos básicos del ciudadano
Obtener imágenes	El sistema se comunica con la fuente externa especificada y guarda las imágenes obtenidas en la BD	Sistema	Identificada la fuente de datos externa a utilizar para obtener las imágenes del ciudadano	Imágenes obtenidas de la fuente externa registradas en el sistema
Mostrar imágenes	El sistema muestra las imágenes obtenidas tanto de la fuente externa como las que seleccionó el Captador de datos e imágenes con calidad en caso de haber seleccionado alguna del propio sistema	Sistema	Imágenes obtenidas de la fuente externa registradas en el sistema	Imágenes obtenidas de la fuente externa y las seleccionadas con calidad en caso de haber obtenido alguna del propio sistema
Seleccionar imágenes con calidad	Luego de que el sistema muestre las imágenes obtenidas de la fuente externa	Captador de datos e imágenes	Mostradas las imágenes obtenidas de la fuente externa	Imágenes seleccionadas

	y las seleccionadas con calidad en caso de haber obtenido alguna del propio sistema, el Captador de datos e imágenes selecciona la que crea que tiene mejor calidad, incluso puede no seleccionar ninguna y las desee captar		y las seleccionadas con calidad en caso de haber obtenido alguna del propio sistema	
Especificar persona asociada	En caso de que en la captura de datos se hayan introducido personas asociadas al titular, el Captador de datos e imágenes selecciona una de esas personas para realizarle la captura imágenes	Captador de datos e imágenes	Imágenes del titular captadas	Imágenes de la persona asociada al titular capturadas
Seleccionar datos e imágenes con problemas	Si luego de la revisión se detectan datos o imágenes incorrectos, el supervisor selecciona la información incorrecta para que quede registrado en el sistema	Supervisor	Datos e imágenes revisadas	Registrados los datos o imágenes con problemas
Cambiar estado de la fase "Supervisión" a "No Efectiva"	Si luego de la revisión se detectan datos o imágenes con problemas se cambia el estado de la fase "Supervisión" a "No Efectiva".	Sistema	Registrados los datos o imágenes con problemas	Estado de la fase "Supervisión" cambiada a "No Efectiva".

Tabla 1. Descripción de las actividades de los procesos de negocio: captura de datos, captura de imágenes y supervisión

Anexo 3 Descripción de las funcionalidades

Precondiciones	Deben estar registrados en el sistema los datos del titular.
Funcionalidades asociadas	RF2, RF8,RF9,RF13, RF14, RF15

Conceptos tratados	Ciudadano, Foto, Huella
Descripción básica	<p>1. El Captador de datos e imágenes selecciona la opción “Capturar imágenes” en el menú principal.</p> <p>2. El sistema muestra la interfaz “Captura de Imágenes” con un listado de los trámites que están en la fase “Captura de Imágenes” con los siguientes datos y opciones:</p> <p>Datos:</p> <ul style="list-style-type: none"> • Número de Documento • Primer Nombre • Segundo Nombre • Primer Apellido • Segundo Apellido <p>Opciones:</p> <ul style="list-style-type: none"> • Captar imágenes <p>3. Si el Captador de datos e imágenes selecciona la opción “Captar imágenes” el sistema muestra la interfaz “Buscar imágenes” con los siguientes datos y opciones:</p> <p>Datos:</p> <ul style="list-style-type: none"> • Imágenes en el sistema (El sistema busca si existen imágenes de la persona registradas y las muestra) <p>Fuente externa</p> <ul style="list-style-type: none"> • Fuente (Listado de fuentes que están configuradas en el sistema para obtener las imágenes) • Imágenes (Imágenes obtenidas de una fuente externa) <p>Las imágenes son mostradas con un checkbox que las permita seleccionar.</p> <p>Opciones:</p> <ul style="list-style-type: none"> • Guardar imágenes <p>4. El Captador de datos e imágenes selecciona las imágenes que tienen la calidad requerida y selecciona la opción “Guardar imágenes”.</p> <p>5. El sistema registra las imágenes seleccionadas con buena calidad como imagen definitiva para esa persona.</p>

6. Si se requiere foto y no fue seleccionada en el paso 3, el sistema muestra la interfaz “Capturar foto” con los siguientes datos y opciones:

Datos:

- Cuadro donde se muestra la imagen en vivo que se está captando
- Cuadro donde se muestra la foto después que ser captada

Opciones:

- Capturar
- Registrar

6.1. Si el Captador de datos e imágenes selecciona la opción Capturar el sistema le envía una señal al dispositivo para la captura de la foto indicando que debe tirar la foto y la visualiza en el cuadro derecho de la interfaz.

6.2. Si la foto captada cuenta con los parámetros de calidad requeridos, el Captador de datos e imágenes selecciona la opción “Registrar”. (En caso de que no se desee realizar la acción ver descripción alterna “Cancelar”).

6.3. El sistema registra la foto correspondiente.

7. Si se requieren las huellas y no fueron seleccionadas en el paso 3, el sistema muestra la interfaz “Capturar huellas” con los siguientes datos y opciones:

Datos:

- Cuadro donde se muestra la imagen en vivo que está captando el lector de huellas
- Cuadro donde se muestra la huella después que ser captada

Opciones:

- Capturar
- Registrar

7.1. Si el Captador de datos e imágenes selecciona la opción

Capturar el sistema obtiene una imagen de la huella y la visualiza en el cuadro derecho de la interfaz.

7.2. Si el Captador de datos e imágenes determina que la huella posee la calidad requerida selecciona la opción "Registrar". (En caso de que no se desee realizar la acción ver descripción alterna "Cancelar").

7.3. El sistema registra las huellas correspondientes.

Prototipos de interfaz de usuario

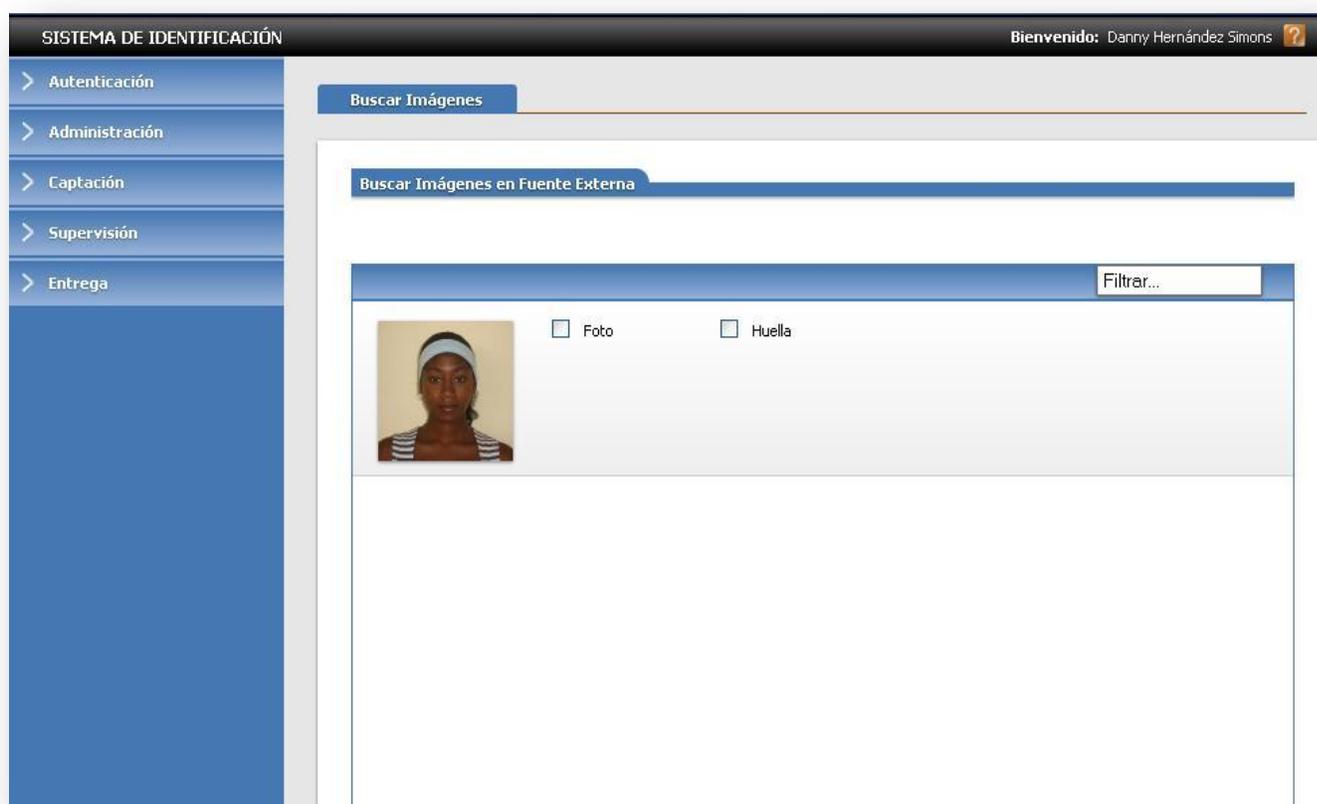


Figura3: Buscar Imágenes en Fuente Externa

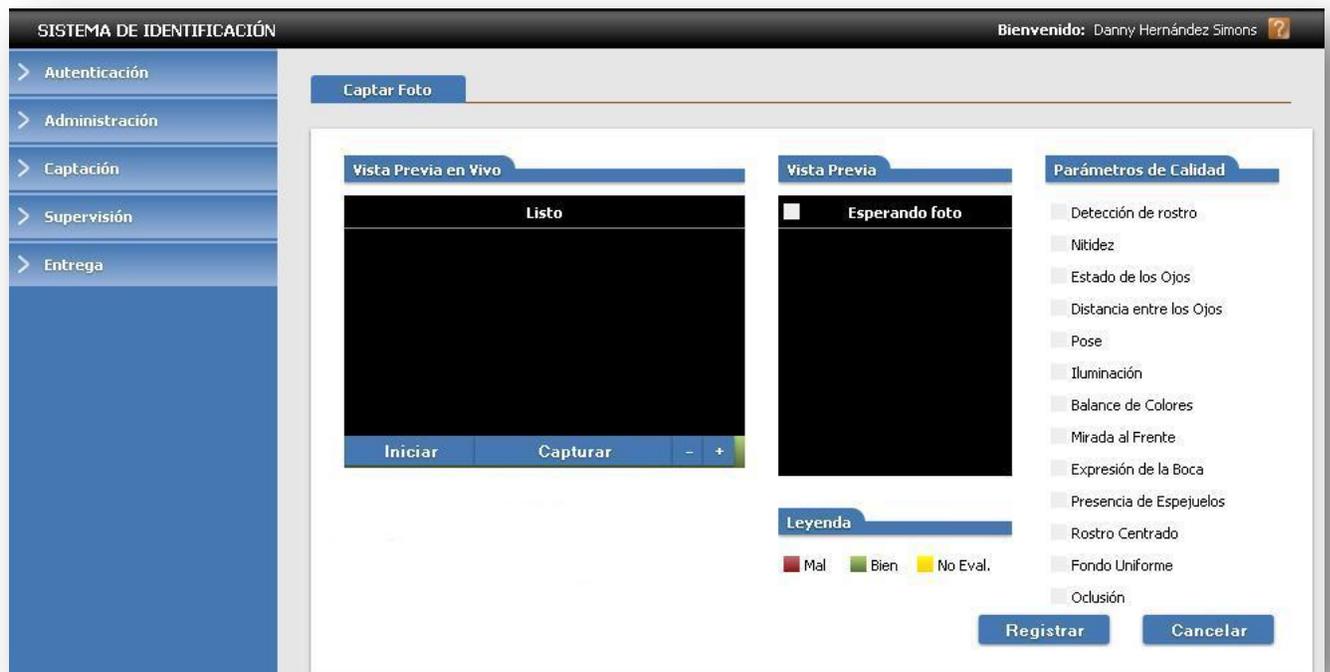


Figura 4: Capturar Foto

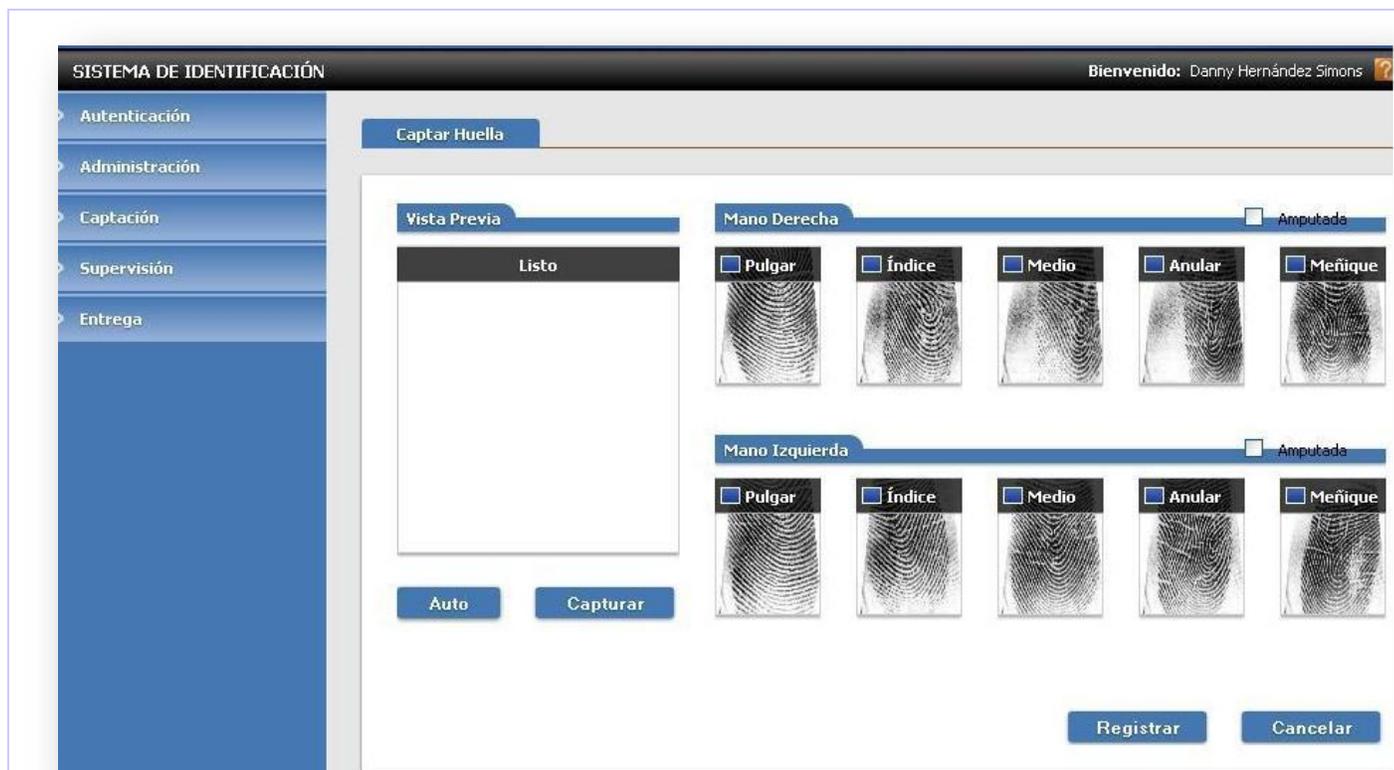


Figura 5: Captar Huella

Descripción alterna	
Descripción alterna 1 "Cancelar"	
	1. Si se selecciona la opción Cancelar el Sistema redirecciona a la interfaz principal.
Validaciones	
Postcondiciones	Quedan captadas las imágenes de la persona titular del documento

Tabla 2. Descripción de las funcionalidades del módulo Captura de Imágenes

Precondiciones	En qué estado debe estar el sistema para que pueda ser
----------------	--

	ejecutado
Funcionalidades asociadas	RF2, RF3, RF7,RF9, RF16, RF17,RF18
Conceptos tratados	Ciudadano, Datos, Imágenes
Descripción básica	<p>1. El Supervisor selecciona en la interfaz principal de la aplicación la opción “Supervisión” y selecciona Supervisión de Datos:</p> <p>2. El sistema muestra la interfaz Supervisar Trámite con las siguientes opciones:</p> <ul style="list-style-type: none"> • Imprimir Planilla (En caso de que se requiera) • Trámites en supervisión <p>3. Si el supervisor selecciona la opción “Imprimir planilla” el sistema muestra la interfaz “Imprimir Planilla” con el listado de los trámites pendientes a imprimir y un listado de los trámites impresos.</p> <p>3.1 El Supervisor selecciona el trámite al cual se le va a imprimir la planilla”.</p> <p>3.2 El sistema muestra una interfaz donde se muestra la planilla del ciudadano con sus datos e imágenes.</p> <p>3.3 El Supervisor selecciona el botón “Imprimir”.</p> <p>3.4 El sistema imprime la planilla.</p> <p>3.5 El sistema elimina el trámite del listado de los trámites pendientes a imprimir y lo coloca en el listado de trámites impresos.</p> <p>4. Si el Supervisor selecciona la opción “Trámites en supervisión”, el sistema muestra la interfaz “Supervisión” con un listado de los trámites que están en la fase “Supervisión” con los siguientes datos y opciones:</p> <p>Datos:</p> <ul style="list-style-type: none"> • Número de Documento • Primer Nombre • Segundo Nombre • Primer Apellido

- Segundo Apellido

Opciones:

- Supervisar

4. El Supervisor introduce el nombre del ciudadano y selecciona el botón Buscar.

5. El Supervisor selecciona el trámite que va a Supervisar

6. Si el Supervisor selecciona la opción "Supervisar", el sistema muestra una interfaz con todos los datos e imágenes que fueron registrados de la persona a través de un checkbox para permitir la selección de los datos e imágenes con errores.

6.1. El Supervisor revisa los datos e imágenes.

6.2. En caso de haber errores el supervisor selecciona los datos o imágenes correspondientes y selecciona la opción "Guardar".

6.3. El sistema registra los datos e imágenes seleccionados con errores y cambia la fase "Supervisión" a "No Efectiva".

Prototipos de interfaz de usuarios

The screenshot displays a web application interface for 'SISTEMA DE IDENTIFICACIÓN'. The top right corner shows a welcome message: 'Bienvenido: Danny Hernández Simons'. On the left, a navigation menu includes options like 'Autenticación', 'Administración', 'Captación', 'Supervisión', 'Supervisión de Datos', 'Imprimir Planillas', and 'Entrega'. The main content area is titled 'Supervisión de Datos' and features a sub-section 'Trámites sin Supervisar' with a search filter. Below this, a table lists five items, each with a small profile picture, name, ID number, and a date/status box.

Nombre	Carné de Identidad	2013
Alisbet Fernández Rojas	91030744334	6 1
Ernesto Sabuquet Hurtado	85120627165	6 11
Elizabet Córdova Viera	91100729698	6 12
Yaciel Leyva Góngora	89100339308	6 12
Yaiquelys Ponce Verdecia	89072541572	6 12

Buttons for 'Supervisar' and 'Salir' are located at the bottom right of the table area.

Figura 6: Trámites en Supervisión

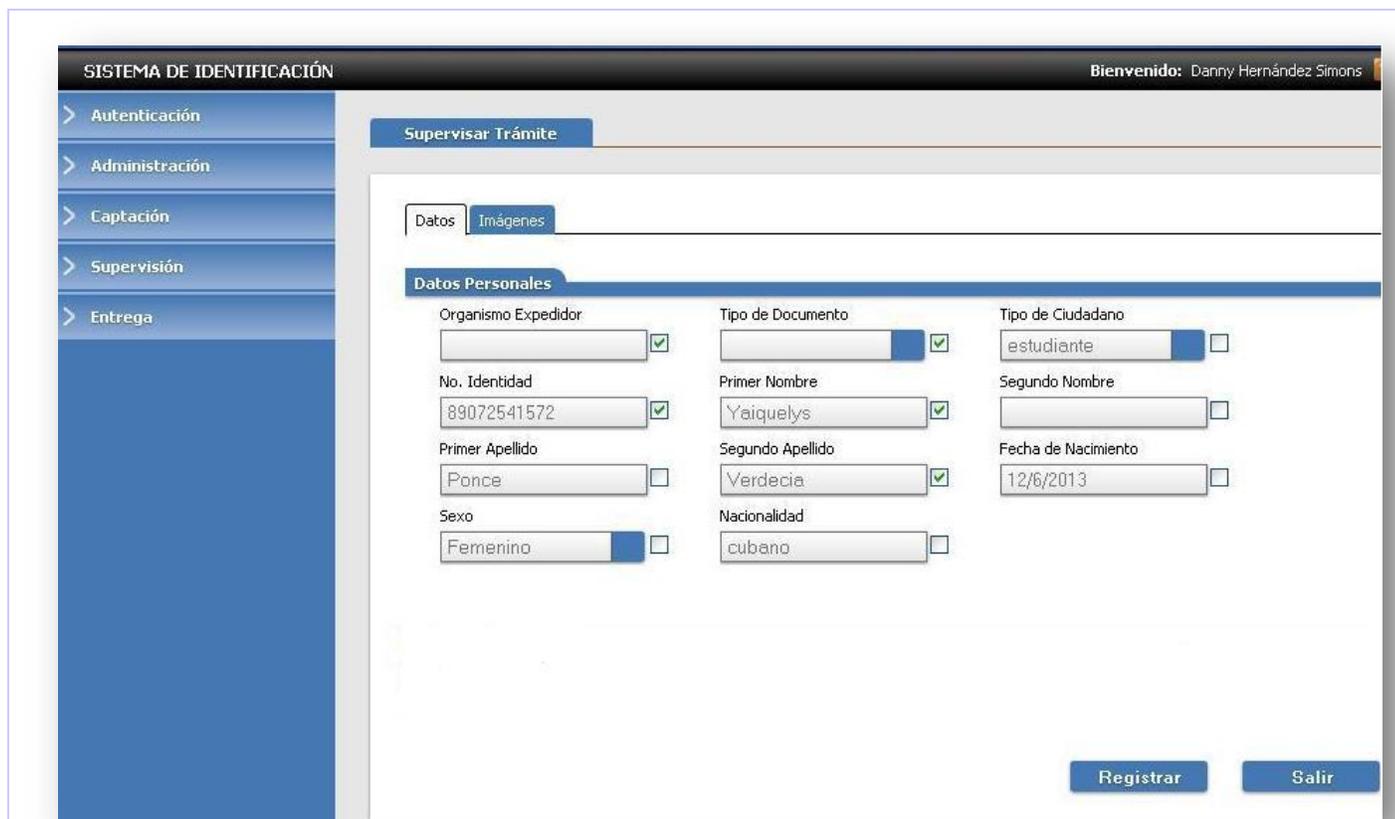


Figura 7: Supervisar Trámite

Postcondiciones

Los datos e imágenes quedan supervisados y listos para imprimir o enviados a corrección en caso de haberse detectado errores.

Tabla 3.Descripción de las funcionalidades del Módulo Supervisión

Anexo 4 Diagramas de clases del diseño

Diagrama de clases del diseño: Corregir datos

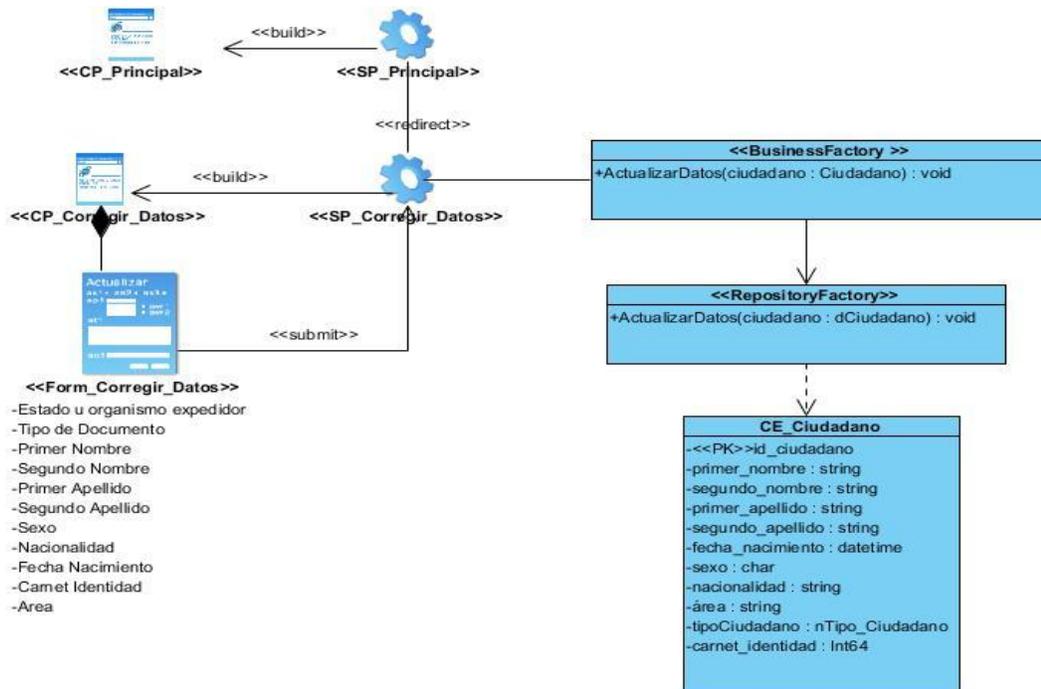


Figura 8: Diagrama de clases del diseño: Corregir datos

Diagrama de clases del diseño: Captar foto

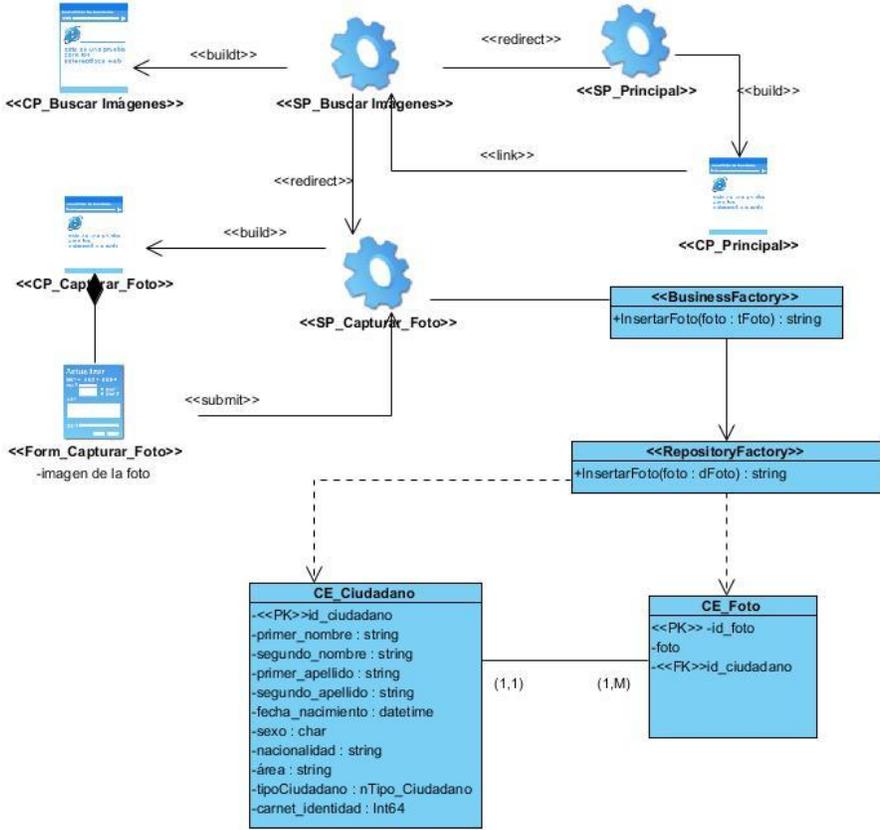


Figura 9: Diagrama de clases del diseño: Capturar foto

Diagrama de clases del diseño: Captar huella

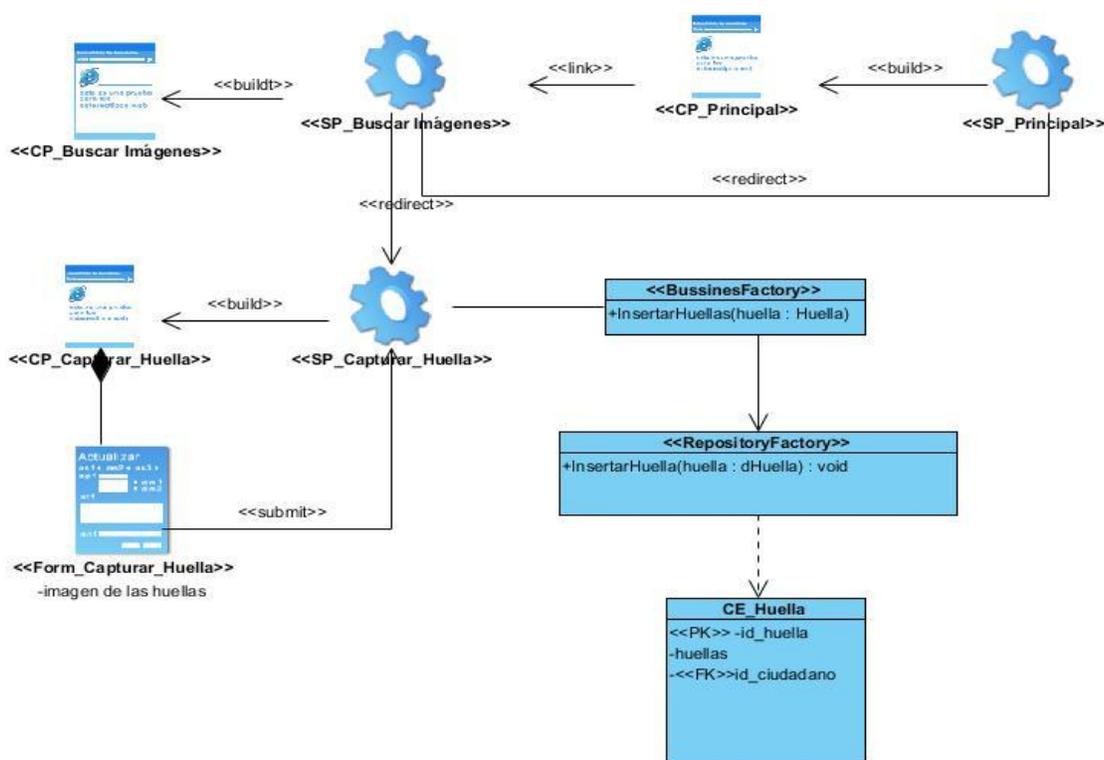


Figura 10: Diagrama de clases del diseño: Capturar huella

Anexo 5 Descripción de las entidades

Nombre	CE_Foto	
Descripción	Representa la entidad que contiene la fotografía del ciudadano	
Nombre del atributo	Tipo de dato	Descripción
id_foto	uuid	Identificador de la entidad CE_Imágenes
foto	byte	Foto del ciudadano
id_ciudadano	uuid	Identificador del ciudadano

Tabla 4. Descripción de la entidad CE_Foto

Nombre	CE_Huella	
Descripción	Representa la entidad que contiene la fotografía del ciudadano	
Nombre del atributo	Tipo de dato	Descripción

id_huella	uuid	Identificador de la entidad CE_Huella
dpulgar	byte	Dedo pulgar derecho
dindice	byte	Dedo índice derecho
dmedio	byte	Dedo medio derecho
danular	byte	Dedo anular derecho
dmennique	byte	Dedo meñique derecho
ipulgar	byte	Dedo pulgar izquierdo
iindice	byte	Dedo índice izquierdo
imediao	byte	Dedo medio izquierdo
ianular	byte	Dedo anular izquierdo
imennique	byte	Dedo meñique izquierdo
id_ciudadano	uuid	Identificador del ciudadano

Tabla 5. Descripción de la entidad CE_Huella

Nombre	CE_Huella	
Descripción	Representa la entidad que contiene la fotografía del ciudadano	
Nombre del atributo	Tipo de dato	Descripción
id_atributo_dato_opcional	uuid	Identificador del atributo
nombre_atributo	varchar	Nombre del atributo
fecha_registro	timestamp	Fecha de registro
id_tipo_dato_opcional	uuid	Identificador del dato opcional

Tabla 6. Descripción de la entidad CE_AtributoDatoOpcional

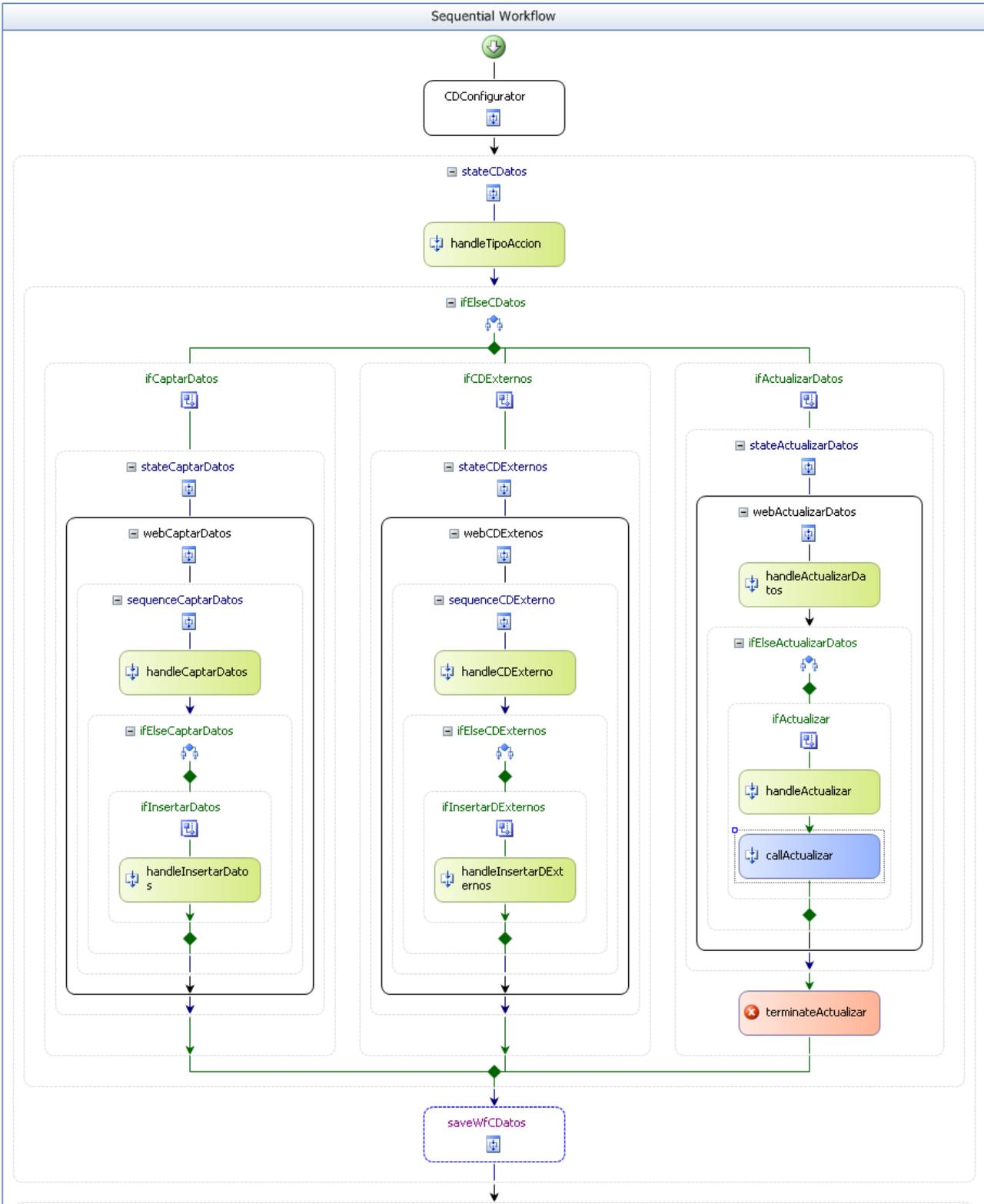
Nombre	CE_ValorDatoOpcional	
Descripción	Representa la entidad que contiene la fotografía del ciudadano	
Nombre del atributo	Tipo de dato	Descripción
id_atributo_dato_opcional	uuid	Identificador del atributo
id_ciudadano	uuid	Identificador del ciudadano
valor	varchar	Valor del dato opcional
fecha_registro	timestamp	Fecha de registro

Tabla 7. Descripción de la entidad CE_ValorDatoOpcional

Nombre	CE_FuenteExterna	
Descripción	Representa la entidad que contiene la fotografía del ciudadano	
Nombre del atributo	Tipo de dato	Descripción
id_fuente_externa	uuid	Identificador de la fuente externa
nombre_fuente_externa	varchar	Nombre de la fuente externa
servidor	varchar	Servidor donde se encuentra ubicada la fuente externa
puerto	numeric	Puerto para la conexión a la fuente externa
usuario	varchar	Nombre del usuario
Contraseña	varchar	Contraseña del ciudadano

Tabla 8. Descripción de la entidad CE_FuenteExterna

Anexo 6 Diseño general del workflow



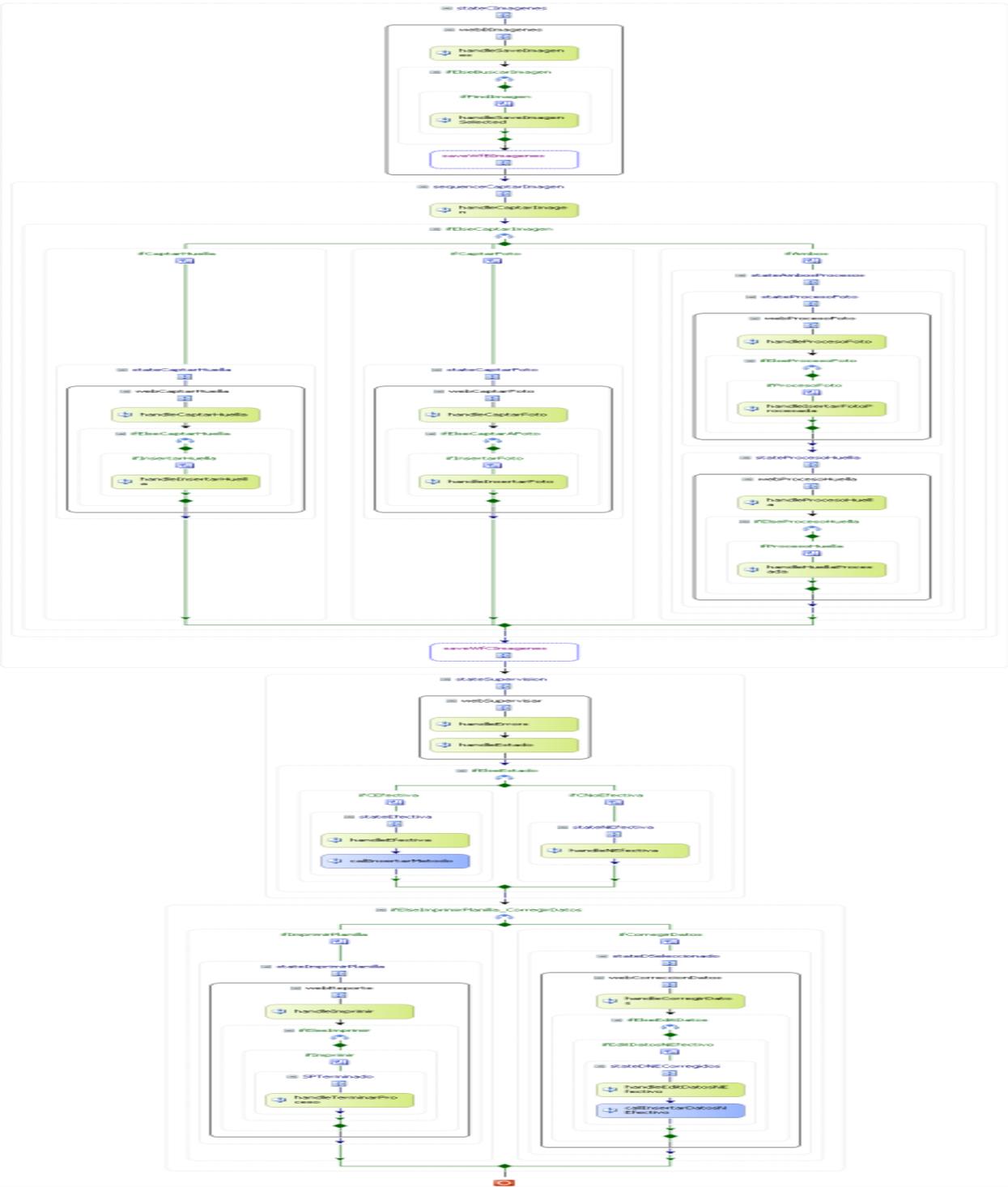


Figura 11: Workflow general utilizado para manejar la captura y supervisión de los datos e imágenes de un ciudadano

Anexo 7: Modelo de datos

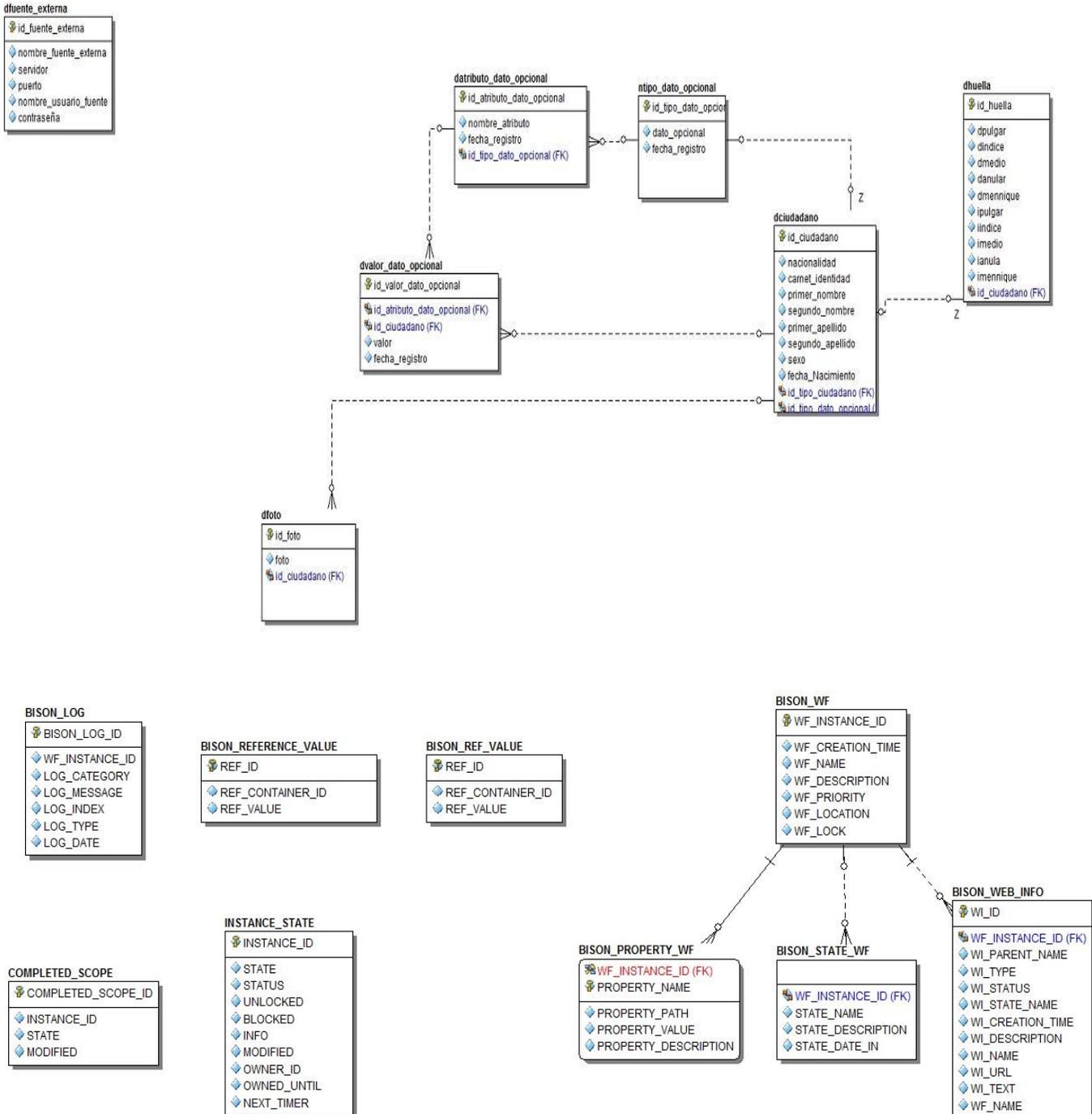


Figura 12: Modelo de Datos