Universidad de las Ciencias Informáticas

Facultad 1

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Título: Módulo para la comunicación de los sistemas no interconectados XABAL IDBIOACCESS y SPDI.

Autores

Haileén Alicia Romero Sanabria

Ernesto Carmona Escalona

Tutores

Ing. Reynier Lester Claro Escalona

Ing. Annie Cubas González

Consultante

Ing. Mayleidis López Fernández

La Habana, Junio 2014

"Lo que hace crecer el mundo no es el descubrir cómo está hecho, sino el esfuerzo de cada uno para descubrirlo."

José Martí

DECLARACIÓN DE AUTORÍA

Declaramos, Haileén Alicia Romero Sanabria y Ernesto Carmona Escalona, ser autores del presente trabajo de diploma y concedemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo con carácter exclusivo.

Para que así conste firmamos a los ___días del mes de _____ del año_____.

_____	_____
Firma del Autor	Firma del Autor
Haileén Alicia Romero Sanabria	Ernesto Carmona Escalona
_____	_____
Firma del Tutor	Firma del Tutor
Ing. Annie Cubas González	Ing. Reynier Lester Claro Escalona

SÍNTESIS DE LOS TUTORES

Ing. Reynier Lester Claro Escalona

Ingeniero en Ciencias Informáticas, UCI 2008.

Categoría docente: Profesor Instructor.

Profesor de la asignatura Proyecto de Investigación y Desarrollo de la Facultad 1. Arquitecto de Software del Departamento de Desarrollo del Centro de Identificación y Seguridad Digital.

Correo electrónico: rlclaro@uci.cu

Ing. Annie Cubas González

Ingeniero en Ciencias Informáticas, UCI 2011.

Analista del Proyecto Plataforma Modular de Identificación y Control de Acceso del Departamento de Desarrollo del Centro de Identificación y Seguridad Digital.

Correo electrónico: acubas@uci.cu

Ing. Mayleidis López Fernández

Ingeniera Industrial Especializada en Organización de Empresas, CUJAE 2003.

Categoría docente: Profesor Asistente.

Profesora de la Facultad 1. Jefa del Departamento de Práctica Profesional.

Correo electrónico: mayleidis@uci.cu

DEDICATORIA Y AGRADECIMIENTOS

En primer lugar doy gracias a Dios por haber pensado en mí cuando nadie lo había hecho, por darme aliento de vida, por escogerme la familia ideal y gracias a ÉL estoy aquí hoy.

En segundo lugar agradezco a esa familia que Dios me dio, a mi mamá, a mi papá, a mi hermano, a mi sobrino, a mi familia de crianza, por tanta preocupación y dedicación a lo largo de toda mi vida y sobre todo en estos 5 años.

A mi novio por soportarme todo este tiempo.

A mis tutores Annie y Reynier, por brindarme su apoyo y sus conocimientos para poder hacer este trabajo.

A mis hermanos en Cristo, por todos los domingos que pasamos juntos.

A mis amigas desde el IPI, Diano, Dami y Nelsy, que a pesar de tantas malcriadeces nos queremos como hermanas.

En fin, a todas las personas que han colaborado de una u otra forma en la realización de este trabajo de diploma.

Por todo Gracias, Haileén Alicia.

.:DEDICATORIA Y AGRADECIMIENTOS:.

Agradezco a mis padres por brindarme la guía para andar el camino que hoy termina. Un camino que ha sido empezado por éstos y por mi abuelo (al cual le dedico esta tesis) en primer lugar, continuado por familiares y amigos, y todo aquel que ha tocado mi vida y ha sabido aportarle sabiduría y luz en estos tiempos que corren marcados por el egoísmo y la oscuridad del hombre.

Son familiares, amigos, profesores y otros que se me han podido olvidar que con sus enseñanzas han forjado y moldeado la persona que soy; parafraseando a mi profesora de Filosofía que nos decía: - el hombre es un ente que es moldeado por la sociedad en la que vive y por aquellos que lo rodean -.

Como resultado de este proceso nace hoy un hombre nuevo que está listo para devolver lo que ha recibido y contribuir a la construcción da la sociedad futura haciendo de ésta un lugar mejor para los que están y los que vendrán.

A todos gracias, Ernesto.

RESUMEN

Un documento de identificación es un documento que contiene datos personales de un individuo y el proceso para la emisión de los mismos cuenta con cinco pasos: Captura de datos, Captura de Imágenes, Supervisión, Personalización y Entrega del Documento.

En la Universidad de las Ciencias Informáticas, específicamente en el Centro de Identificación y Seguridad Digital se desarrolló el sistema XABAL IDBIOACCESS con el objetivo de emitir documentos de identificación en la institución, este sistema para realizar la personalización se apoya en el Sistema de Personalización de Documentos de Identificación.

Cuando el XABAL IDBIOACCESS fue presentado en la feria de productos informáticos de la UCI, los representantes de las empresas interesadas dijeron que no les resultaba factible adquirir el SPDI además del XABAL IDBIOACCESS, debido a que incurre en gastos adicionales, por lo que la UCI propuso separar estos dos sistemas y ofrecer el servicio de personalización a cualquier entidad que lo solicite. Una vez separados los sistemas y para satisfacer las necesidades de los clientes se propone el desarrollo del "Módulo para la comunicación de los sistemas no interconectados XABAL IDBIOACCESS y SPDI", el cual permite que estos dos software se comuniquen sin que exista conexión directa entre ellos y brinda además, la posibilidad de que las empresas puedan emitir sus credenciales utilizando solamente el sistema XABAL IDBIOACCESS sin tener que adquirir el SPDI.

Palabras claves: comunicación, documento de identificación, no interconectados, personalización.

ÍNDICE DE CONTENIDO

.:ÍNDICE DE CONTENIDO:.

.:ÍNDICE DE CONTENIDO:.

ÍNDICE DE FIGURAS

.:ÍNDICE DE FIGURAS:.

ÍNDICE DE TABLAS

INTRODUCCIÓN

Las Nuevas Tecnologías de la Información y las Comunicaciones (NTIC) han impactado la vida cotidiana del hombre en las últimas décadas y la sociedad se ha visto transformada con la evolución acelerada de las mismas. El desarrollo científico-técnico alcanzado en el mundo en los últimos años ha llevado al hombre a un progreso eminente donde casi es imposible prescindir de la informática, permitiendo un perfeccionamiento de herramientas y tecnologías puestas a disposición de la sociedad.

Estas tecnologías y herramientas han permitido ahorrar tiempo y esfuerzo humano automatizando todos los procesos que una vez fueron engorrosos y lentos. Su utilización en diversos sectores de la sociedad ha mostrado éxito trayendo resultados beneficiosos para la economía y satisfacción del hombre.

Las NTIC han logrado acomodar al hombre hasta con los documentos de identificación, antiguamente se tenía que llevar siempre consigo un conjunto de papeles que permitían a los ciudadanos identificarse, mientras que hoy, sólo una simple tarjeta con un minúsculo código los identifica, código en el que se puede guardar toda la información referente a un individuo, desde su nombre hasta su grupo sanguíneo.

El **proceso de emisión de documentos de identificación** cuenta generalmente con los siguientes pasos:

- **Captura de Datos:** Donde se captan los datos biográficos de la persona.
- **Captura de Imágenes:** Donde se captan los datos biométricos de la persona.
- **Supervisión:** El proceso de revisión de la información captada en los pasos anteriores.
- **Personalización:** Consiste en la impresión de los datos del titular en el documento de identificación.
- **Entrega:** Es el proceso donde se le entrega el documento personalizado al titular correspondiente.

En el Centro de Identificación y Seguridad Digital (CISED) perteneciente a la Universidad de las Ciencias Informáticas (UCI), se desarrolló el sistema XABAL IDBIOACCESS, su objetivo es emitir documentos de identificación en la institución.

Después de desplegado este sistema fue presentado en la feria de productos informáticos de la UCI, interesándose en el mismo empresas como Servicios a la Aviación Civil S.A (SERVAC), Laboratorios Biológicos Farmacéuticos (LABIOFAM) e Instituto de Meteorología (INSMET).

El sistema XABAL IDBIOACCESS en el proceso de personalización de las credenciales utiliza el Sistema de Personalización de Documentos de Identificación (SPDI) y a las entidades interesadas no les resulta factible adquirir el SPDI pues implicaría la compra de equipamiento como: impresoras y máquina de

plastificar, además de contratar al personal capacitado para el empleo de dicho sistema, representando esto un costo adicional al software de emisión de las credenciales. Debido a esta situación y a que la UCI desea comercializar sus productos, la universidad propuso prestar un servicio de personalización de documentos de identificación a cualquier empresa que lo solicite, donde la entidad interesada tendría que adquirir solamente el resto de los módulos del sistema XABAL IDBIOACCESS quedando éstos aislados del SPDI.

Teniendo en cuenta lo anteriormente planteado, se presenta como **problema de investigación:** *¿Cómo lograr la comunicación de manera segura entre los sistemas no interconectados XABAL IDBIOACCESS y SPDI?*

Por lo que se define como **objeto de estudio:** *El proceso de comunicación entre sistemas no interconectados.*

Teniendo como **objetivo general:** *Desarrollar un módulo que permita la comunicación segura de los sistemas no interconectados XABAL IDBIOACCESS y SPDI para la emisión de documentos de identificación.*

Enmarcándose como **campo de acción:** El proceso de *comunicación entre los sistemas no interconectados XABAL IDBIOACCESS y SPDI.*

Se tiene como **idea a defender:** *El desarrollo del módulo permitiría la comunicación de los sistemas no interconectados XABAL IDBIOACCESS y SPDI para la emisión de documentos de identificación.*

A partir del objetivo general se desglosan los siguientes **objetivos específicos:**

- Establecer los fundamentos teóricos y metodológicos relacionados con la comunicación entre sistemas no interconectados.
- Realizar análisis y diseño del Módulo para la comunicación de los sistemas no interconectados XABAL IDBIOACCESS y SPDI.
- Implementar el Módulo para la comunicación de los sistemas no interconectados XABAL IDBIOACCESS y SPDI.
- Validar la solución implementada.

Para llevar a cabo los objetivos antes mencionados se trazaron las siguientes **tareas a realizar:**

- Definición de los principales conceptos asociados a la problemática planteada. (Haileén Alicia Romero Sanabria).
- Caracterización de sistemas homólogos. (Haileén Alicia Romero Sanabria).

- Caracterización de mecanismos de seguridad que pueden ser aplicados a ficheros de datos. (Haileén Alicia Romero Sanabria y Ernesto Carmona Escalona).
- Definición de los mecanismos de seguridad que serán utilizados en el Módulo para la comunicación de los sistemas no interconectados XABAL IDBIOACCESS y SPDI. (Haileén Alicia Romero Sanabria y Ernesto Carmona Escalona).
- Definición de las herramientas informáticas, tecnologías y metodología a usar en el desarrollo del Módulo para la comunicación de los sistemas no interconectados XABAL IDBIOACCESS y SPDI. (Haileén Alicia Romero Sanabria y Ernesto Carmona Escalona).
- Definición de modelos de dominio del módulo. (Haileén Alicia Romero Sanabria).
- Definición de los requisitos funcionales y no funcionales. (Ernesto Carmona Escalona).
- Diseño de prototipos no funcionales de interfaz de usuario. (Haileén Alicia Romero Sanabria).
- Descripción de los requisitos funcionales del sistema. (Ernesto Carmona Escalona).
- Realización del modelo de diseño. (Haileén Alicia Romero Sanabria y Ernesto Carmona Escalona).
- Implementación de la solución. (Haileén Alicia Romero Sanabria y Ernesto Carmona Escalona).
- Caracterización de los tipos de pruebas de software. (Haileén Alicia Romero Sanabria).
- Definición de las pruebas a realizar. (Haileén Alicia Romero Sanabria y Ernesto Carmona Escalona).
- Aplicación de las pruebas definidas. (Haileén Alicia Romero Sanabria y Ernesto Carmona Escalona).
- Presentación de los resultados de las pruebas aplicadas. (Haileén Alicia Romero Sanabria y Ernesto Carmona Escalona).

Se emplearon diversos **métodos de investigación**, tales como los **Métodos Teóricos y Empíricos**.

Dentro de los **Métodos Teóricos**, los siguientes:

- **Analítico – Sintético:** Se utilizó este método para descomponer el objeto de estudio y estudiar sus partes con facilidad y así descubrir las características fundamentales de las mismas y los nexos existentes entre ellas.
- **Inductivo – Deductivo:** Este método fue empleado para estudiar las particularidades de la comunicación de sistemas que no se encuentran interconectados y de esta manera llegar a un conocimiento general para desarrollar la solución.
- **Modelación:** Se escogió la Modelación para confeccionar diagramas y modelos con el objetivo de entender y representar los elementos claves relacionados con la solución.

De los **Métodos Empíricos** se aplicó el que a continuación se expone:

- **Observación:** Se empleó para apreciar el funcionamiento de los sistemas XABAL IDBIOACCESS y SPDI en entorno real.

Justificación de la Investigación: Con el desarrollo del "Módulo para la comunicación de los sistemas no interconectados XABAL IDBIOACCESS y SPDI", las empresas del país podrán emitir sus credenciales a través del sistema XABAL IDBIOACCESS que la UCI les ofrece sin tener que adquirir el SPDI y la universidad podrá prestar un servicio para la personalización de documentos de identificación, representando esto una entrada monetaria para la misma y a su vez un ahorro para las instituciones del país.

El presente trabajo de diploma está constituido por tres capítulos, los cuales quedarán descritos a continuación:

Capítulo 1: Fundamentos Teóricos y Metodológicos: En este capítulo se realiza un estudio acerca de los sistemas para la emisión de documentos de identificación y de otros sistemas que se comunican sin conexión directa, reconociendo a la vez los conceptos fundamentales relacionados con el objeto de estudio. Además, se caracterizan las herramientas, metodología y tecnologías a utilizar para llevar a cabo la solución y se definen los mecanismos de seguridad que serán aplicados para proteger la información.

Capítulo 2: Propuesta de Solución: En esta etapa de la investigación se presenta la propuesta de solución del problema planteado y se concreta una lista de requerimientos para guiar a los desarrolladores durante la implementación de la solución. Además, se confeccionan diagramas y modelos que rigen el flujo de las actividades.

Capítulo 3: Implementación y Prueba: En este capítulo se establecen estándares de codificación para crear un ambiente de desarrollo intuitivo para los programadores. Además, se modelan dos tipos de diagramas, uno para conocer cómo está compuesto el módulo y otro para reflejar el entorno donde se desplegará la solución. Llevado a cabo todo esto, se implementan cada una de las funcionalidades descritas y se le aplican las pruebas necesarias para comprobar su funcionamiento.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS Y METODOLÓGICOS

1.1 Introducción.

Las empresas cubanas necesitan tener el control de sus trabajadores, por lo que se hace preciso que cada uno porte una credencial para acceder a la institución que pertenezca. Para la confección de las credenciales estos organismos deberán adquirir un sistema para la emisión de documentos de identificación y la UCI les ofrece el sistema XABAL IDBIOACCESS. Para que esta aplicación sea adaptable a cualquier entorno será necesario desarrollar un módulo que permita comunicar este sistema con el SPDI sin que exista conexión directa entre ellos, por lo que para desarrollar dicho módulo es ineludible realizar un estudio de sistemas que emiten documentos de identidad y de sistemas no interconectados, además se deberán seleccionar tecnologías, metodología, herramientas y mecanismos de seguridad para llevarlo a cabo.

1.2 Software para la emisión de documentos de identificación.

Hace ya varios siglos surgió la necesidad de identificar a las personas; todo data del viejo continente (Eurasia) cuando se intentó segregar a la población mediante un documento de identificación; clasificando así los sectores más bajos en ‘desviados’, prostituidos, locos, delincuentes y nómades.

Un **documento de identificación** es un documento que contiene datos personales de un individuo, el mismo es emitido por un empleado público con autoridad competente para permitir la identificación personal. En algunos países se conoce como Carné de Identidad (CI), Tarjeta de Identidad (TI), Cédula de Ciudadanía (CC), Registro Civil (RC), Cédula de Extranjería (CE), Cédula de Identidad (CI), Documento Nacional de Identidad (DNI), identificación oficial o simplemente identificación, todos los residentes de un país deben poseer uno de los antes mencionados, pero existen los que son opcionales como licencias de conducción, pasaportes, credenciales, carné de la organización a la que pertenezca, entre otros. Los documentos de identificación se renuevan en el período de tiempo que estime conveniente la administración pública, ya sean 5, 10 años o más. Los documentos de identificación son de vital importancia para las personas, tanto para identificarlas como para permitirles ejecutar acciones legales en la sociedad.

Hasta hace algunos años la confección de documentos de identificación era manual, lo cual era un proceso muy lento, engorroso y propenso a equivocaciones. Hoy, las NTIC han dado un vuelco total a este proceso, pues se han desarrollado software que lo realizan de manera rápida y libre de errores.

A nivel mundial existen diversos software para la emisión de documentos de identificación, en el transcurso de la investigación se indagó acerca de algunos que se muestran a continuación.

1.2.1 Datacard ID Works.

El software **Datacard ID Works** proporciona a los usuarios poderosas opciones para la emisión de tarjetas de identificación; primero, el módulo *Designer*, que permite establecer conectividad con bases de datos y el diseño de la tarjeta, permitiendo realizar la personalización mediante plantillas, gráficas y textos.

El segundo módulo de *Id Worksproduction* ofrece la captura de imágenes, la producción de tarjetas y reportes. Estas opciones están disponibles en venta por separado o en kit completo.

Datacard ID Works viene disponible en versión básica, estándar y empresarial, es compatible con todas las impresoras Datacard y con el sistema operativo Windows, opcionalmente tiene soporte para la emisión de tarjetas con *chip smart card*[1], banda magnética, captura de firma mediante dispositivo compatible o de huella digital mediante lectores biométricos.

El grupo Datacard además de comercializar software para credenciales, vende paquetes que incluyen todo lo necesario para la creación de las mismas, impresora de tarjetas, cámara fotográfica digital, cintas monocromáticas, tarjetas pvc y otros. Este grupo español es el impulsor de la gran mayoría de los programas más importantes de emisión de tarjetas a nivel mundial. Las soluciones para la identificación segura y personalización de tarjetas se utilizan a diario para emitir millones de tarjetas financieras y documentos de identidad. Es una empresa con más de 1.400 empleados en todo el mundo, con una red de servicios y apoyo que abarca más de 120 países. Comercializa sus líneas de impresoras en diversas regiones que incluyen centros de desarrollo en el Reino Unido, Alemania, Francia, India, Japón, Malasia y Estados Unidos. [1]

1.2.2 SAIME.

El Servicio Administrativo de Identificación, Migración y Extranjería (**SAIME**) es un software desarrollado especialmente para una entidad que lleva su mismo nombre radicada en la República Bolivariana de Venezuela, la misma se dedica a la emisión de cédulas de identidad, pasaportes, entre otros tipos de documentos.

Esta aplicación informática permite a Venezuela mejorar la seguridad ciudadana ofreciendo a sus residentes un pasaporte que cumpla con los estándares internacionales.

[1] Chip smart card: es un circuito integrado para tarjetas inteligentes.

Este software para imprimir los documentos utiliza el sistema de Personalización del Chip de Bundesdruckerei[2] (BCP, por sus siglas en inglés) y en este proceso se tienen en cuenta los siguientes elementos:

- Grabación en láser de los pasaportes.
- Personalización electrónica en conformidad con los estándares de la OACI[3].
- Verificador óptico en tiempo real.
- Garantía de calidad.
- Generación de información de estado que puede solicitar el Sistema de Identidad de Venezuela.
- Eliminación de los datos personales del sistema.

BCP es una solución flexible para la implementación de flujos de trabajo para gestionar la personalización de pasaportes electrónicos. Una de las características principales es la codificación de los datos de personalización en el chip en diferentes escenarios de personalización. Además, puede adaptarse a diferentes aplicaciones, proveedores de chips y fabricantes de sistemas operativos para chips; esto significa que pueden implementarse distintos documentos de identidad tales como pasaportes, los permisos de conducir o la cédula de identidad con registro de firma, utilizando para ello disímiles características biométricas.[2]

El **SAIME** proporciona al módulo de BCP los datos correspondientes de los pasaportes a personalizar en formato XML y éste le informa sobre todas las actividades y acontecimientos que han tenido lugar. BCP es responsable de la preparación de datos conforme a los estándares, de la personalización óptica y de la personalización eléctrica de pasaportes y el SAIME de la inscripción de ciudadanos, de la gestión de orden, de la gestión de existencias y del despacho de documentos.

Todas las interfaces entre BCP y el **SAIME** se implementan utilizando un servicio web y la comunicación es segura y cifrada empleando el protocolo HTTPs. Los servicios web ofrecidos por BCP se despliegan en un servidor de comunicación.

1.2.3 Sistema de emisión de documentos de identidad 3M™.

El **Sistema de emisión de documentos de identidad 3M™** permite la generación de licencias de conducción, credenciales y otros tipos de documentos. Este software puede configurarse para reflejar las políticas del cliente y su flujo de trabajo, cuenta además con intuitivas interfaces de usuario, con amplia capacitación y asistencia técnica.

[2] Bundesdruckerei: Compañía alemana proveedora de soluciones informáticas para personalización de documentos de identificación.
[3] OACI: Organización de la Aviación Civil Internacional.

.:CAPÍTULO 1: FUNDAMENTOS TEÓRICOS Y METODOLÓGICOS:.

El **Sistema de emisión de documentos de identidad 3M™** pertenece a 3M Aquires Cogent, Inc., una compañía de innovación a nivel mundial que se dedica a la producción de miles de productos y son líderes en mercados tan diversos como: Salud, Seguridad, Industria, Consumo, Energía y Electrónicos. Esta compañía tiene su oficina central en St. Paul Minnesota, Estados Unidos y sucursales en más de 100 países.

Características del **Sistema de emisión de documentos de identidad 3M™:**

- Se verifican los solicitantes, según sus criterios, para garantizar que estén autorizados a recibir el documento.
- Las consultas pueden realizarse en texto, imágenes o información biométrica en bases de datos internas o externas.
- La arquitectura completamente escalable del sistema se adapta a las necesidades de los emisores de documentos pequeños, centrales y grandes.
- Disponible para licencias de conducción.
- Los materiales son duraderos y resistentes a la falsificación, la alteración y el uso de productos químicos.
- El diseño de la tarjeta puede personalizarse para adaptarse a los requisitos locales.
- Disponible para identificaciones de navegantes.
- Antes de la entrega de la tarjeta a los solicitantes, se verifica el código de barras para garantizar que su codificación sea correcta.
- El control de inventario realiza un seguimiento de los documentos en una ubicación determinada y permite a los operadores autorizados anular los documentos y deshacerse de ellos cuando sea necesario. [3]

Este software ofrece una amplia variedad de opciones de impresión, producción y personalización para ayudar al cliente a enfrentar desafíos específicos y lograr el nivel de seguridad deseado para pasaportes y tarjetas de identificación.

1.2.4 EMIPAS.

El sistema **EMIPAS** es un software para la emisión de pasaporte, desarrollado por la empresa cubana DATYS[4], que cumple con estándares de seguridad y calidad internacionales en un ambiente amigable y flexible, posee una interfaz sencilla e intuitiva, garantizando a las oficinas tramitadoras y a las emisoras, eficacia y eficiencia en la tramitación del documento.

[4] DATYS: Empresa cubana de Tecnologías y Sistemas.

.:CAPÍTULO 1: FUNDAMENTOS TEÓRICOS Y METODOLÓGICOS:.

DATYS es una institución encargada de producir bienes y servicios informáticos, desarrollando el empleo integral de las tecnologías de la información, de las comunicaciones y de seguridad técnica, con alta calidad y eficiencia para ser reconocida por ayudar a las organizaciones a mejorar sustancialmente su gestión y satisfacer las necesidades y expectativas de sus usuarios. [4] Dentro de sus clientes en el mercado exterior se encuentran México, Francia, Venezuela y otros.

El sistema **EMIPAS** permite:

- Personalización de Pasaportes de Lectura Mecánica.
- Inclusión de código de barras bidimensional para datos biométricos y alfanuméricos.
- Control de Calidad y Supervisión del pasaporte, que garantizan fiabilidad y exactitud de la información contenida.
- Control de los inventarios de los pasaportes y el estado de cada documento personalizado.
- Permite la captación de la información en vivo o a través de documentos que la contengan.
- Posibilidad de verificación de cada persona que solicita pasaporte contra Listas de Impedimentos[5].
- Verificación dactilar en el momento de la entrega del pasaporte.
- Transmisión de datos cifrados entre las oficinas.
- Registro de las máquinas que tienen acceso al sistema.

Características de **EMIPAS**:

- No depende de equipamientos específicos. No obstante, el sistema controla los medios que están registrados dentro de cada configuración para garantizar la confiabilidad.
- Dispone de herramientas para garantizar la unicidad de las personas en la Base de Datos.
- Es un sistema de impresión centralizada y/o descentralizada según el interés del cliente.
- Puede trabajar de manera independiente o integrada a un sistema de identidad.
- Dispone de mecanismos de seguridad y auditoría de la información.
- Se soporta en el sistema operativo Windows. [5]

1.2.5 EMIMAR.

EMIMAR es un sistema para la emisión del carné de identificación del marino, también desarrollado por la empresa DATYS, cumple además con las normas internacionales de la OACI para documentos de lectura mecánica en un ambiente agradable y flexible.

EMIMAR permite:

[5] Listas de Impedimentos: Listado de elementos que los viajeros no pueden poseer; por ejemplo enfermedades infecto-contagiosas, antecedentes penales, etc.

- La captura y almacenamiento de información biométrica (huellas, firmas, foto, etc.).
- Verificación de calidad de la imagen según las normas internacionales de la OACI.
- Control de la calidad de las huellas capturadas.
- Posibilidad de verificación contra una Lista de Negra[6].
- Verificación de la producción de los documentos.

Características que **EMIMAR** posee:

- Recoge datos exclusivos de marineros y personal del mar.
- Inclusión de código de barras bidimensional para datos biométricos y alfanuméricos.
- Mecanismos de seguridad y auditoría de la información.
- Herramientas de detección y eliminación de duplicados.
- Autenticación de los usuarios por contraseña y huella. [6]

1.2.6 Análisis de los sistemas de identificación estudiados.

A partir del estudio realizado acerca de los sistemas de emisión de documentos de identificación se pudo concluir que las empresas fabricantes de este tipo de sistemas ofrecen soluciones completas para emitir los documentos, en ningún momento se encuentra la personalización separada del resto del sistema, es decir, sin conexión directa. Debido a esto se hizo necesario investigar acerca de otros sistemas que, aunque no sean de identificación, brinden características que aporten elementos importantes en el desarrollo del Módulo para la comunicación de los sistemas no interconectados XABAL IDBIOACCESS y SPDI.

1.3 Sistemas no interconectados.

Los **sistemas no interconectados** son aquellos que se comunican asincrónicamente.

La **comunicación asincrónica** se establece entre dos o más personas/sistemas de manera diferida en el tiempo cuando no existe coincidencia temporal. Además, intervienen algunos elementos como el **emisor**, que es el que envía la información sabiendo que no obtendrá respuesta inmediata, el **receptor** es el que sabrá de la llegada del mensaje cuando acceda al canal especificado entre ambos y por último el **canal**, que es el medio físico por el que se transmite la información.

A continuación se aborda acerca de algunos sistemas no interconectados.

[6] Lista Negra: lista de personas que son buscadas por la justicia.

1.3.1 AFIS Civil de Venezuela.

AFIS (*Automated Fingerprint Identification System*) es un sistema de identificación de huella dactilar que permite verificar la concordancia de una huella con cualquiera de las que el propio sistema posee en su base de datos. Todo ello lo hace de manera automática, efectiva y rápida. **AFIS** es una terminología general, por lo que cualquier sistema con estas características se denomina así.

El **AFIS Civil de Venezuela** recibe, procesa y responde a solicitudes de identificación y autenticación provenientes de sistemas del Ministerio del Interior y Justicia (MIJ) o de otros sistemas a través de una interfaz única.

El **AFIS Civil** se integra en estos sistema como un bloque "motor de identificación" que responde a solicitudes de identificación, autenticación o recuperación de datos, por el intermedio de datos al formato ANSI/NIST[7] enviados por el protocolo SMTP[8].

El Servidor SMTP es el punto de entrada y salida del AFIS, materializa la comunicación entre el mundo externo y el AFIS.

Recibe las diversas peticiones (autenticación, identificación, etc.) bajo la forma de correos electrónicos que contienen un archivo ANSI/NIST adjunto. Asimismo, responde a dichas peticiones enviando correos electrónicos que contienen un archivo ANSI/NIST adjunto.

El Servidor SMTP se comunica únicamente y en forma directa con un servidor de correo electrónico del MIJ designado especialmente para ese fin. [7]

1.3.2 SmartCoP.

SmartCoP es una plataforma en construcción para el desarrollo de servicios en línea utilizando tarjetas inteligentes, es funcional para diferentes tipos de navegadores web y sistemas operativos, aprovechando al máximo todas las ventajas de éstas a través de Internet. La plataforma en sí va dirigida a los desarrolladores de sitios web que deseen sustentar su seguridad y/o funcionalidad en el uso de tarjetas inteligentes.

Esta plataforma está siendo construida por los miembros del departamento de Tarjetas Inteligentes, perteneciente al CISED. Se tuvo una reunión con uno de sus miembros donde explicó algunas características del sistema y se tomaron algunas notas de interés que se exponen a continuación.

[7] ANSI/NIST: Formato de datos para el intercambio de huellas dactilares, facial.
[8] SMTP: Es el protocolo simple de transferencia de correo, protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto.

Cuando un desarrollador termine de programar un *plugin,* lo envía al administrador de **SmartCoP** por correo electrónico o se lo hace llegar en algún dispositivo de almacenamiento. Una vez que el administrador tiene en sus manos el *plugin* lo copia en un directorio específico que la plataforma tiene configurado para que lo cargue de allí automáticamente.

1.3.3 Control de Hoteles en Oficinas y Control de Hoteles en Establecimientos.

En el marco del convenio Cuba-Venezuela se desarrolló en el CISED los sistemas **Control de Hoteles en Oficinas** y **Control de Hoteles en Establecimientos**, este último en su primera versión constituyó el trabajo de diploma “Sistema para la informatización del proceso del Control de Estancias en Hoteles para la Dirección de Migración y Fronteras de la República Bolivariana de Venezuela” de la autora Yadira Lázara Rodríguez Peláez en el año 2009. El sistema de Control de Hoteles en Establecimientos se encarga fundamentalmente de registrar los huéspedes extranjeros que se hospedan en los diferentes hoteles, pensiones y posadas de la República Bolivariana de Venezuela. El registro de los huéspedes extranjeros debe entregarse en las oficinas del SAIME (como entidad) cada 7 días según establece la Ley de Migración y Extranjería. El sistema de Control de Hoteles en Establecimientos genera un fichero XML encriptado con el registro de sus huéspedes. Para lograr que el registro de huéspedes de los diferentes establecimientos de hospedaje llegara a la oficina correspondiente en el tiempo establecido se desarrolló una integración entre ambos sistemas con las siguientes variantes:

- Llevar directamente el fichero en un dispositivo de almacenamiento masivo.
- Subir el fichero a través de un servidor FTP.
- Enviar el fichero por correo electrónico.

Una vez que se recibe el fichero con el registro de los huéspedes en las oficinas, el mismo es descifrado y se procesa su información.

1.3.4 Resultados arrojados del estudio de los sistemas no interconectados.

Con el estudio realizado se pudo identificar diferentes vías útiles para la transportación de la información de un sistema a otro, las cuales son: transportar la información en un dispositivo de almacenamiento, enviarla mediante por correo electrónico o guardarla en un servidor FTP. Se conoció sobre algunas variantes para el tipo de archivo que se puede emplear para guardar la información a transportar y los elementos de seguridad que pueden ser aplicados, dando esto la posibilidad de ponerlas en práctica en el módulo a desarrollar.

1.4 Mecanismos de seguridad.

La información, sea confidencial o no, puede ser enviada por Internet o por otra vía, pero corre el riesgo de que agentes no deseados se apoderen de ella. Evitar que la información sea interceptada es imposible, pero sí se puede impedir que sea leída y modificada por intrusos, para esto se han creado mecanismos de cifrado.

Un **mecanismo de cifrado,** también conocido como **mecanismo de encriptación**, en informática, es un algoritmo matemático que se le aplica a la información, donde el texto plano pasa a ser un conjunto de caracteres no entendibles a la vista humana, todo esto para mantener la información segura, y así garantizar su confidencialidad e integridad.

Algunos de los grupos en los que se pueden clasificar los algoritmos criptográficos son los siguientes:

- Criptografía simétrica o de clave secreta.
- Criptografía asimétrica o de clave pública.

La **criptografía simétrica** es un método criptográfico en el cual se usa una misma clave para cifrar y descifrar mensajes. Las dos partes que se comunican han de ponerse de acuerdo con anterioridad sobre la clave a usar. Una vez que ambas partes tienen acceso a esta clave, el remitente cifra un mensaje usando la clave, lo envía al destinatario, y éste lo descifra con la misma clave.

La **criptografía asimétrica** es otro método criptográfico en el que se usan un par de claves para el envío de mensajes. Las dos claves pertenecen a una misma persona, una clave es pública y se puede entregar a cualquier persona, la otra clave es privada y el propietario debe guardarla de tal manera que nadie tenga acceso a ella. Además, los métodos criptográficos garantizan que esa pareja de claves sólo se puede generar una vez, de modo que se puede asumir que no es posible que dos personas hayan obtenido casualmente la misma pareja de claves. Si el remitente usa la clave pública del destinatario para cifrar el mensaje, una vez cifrado, sólo la clave privada del destinatario podrá descifrar este mensaje, ya que es el único que la conoce. Por tanto se logra la confidencialidad del envío del mensaje, nadie salvo el destinatario puede descifrarlo. [8]

1.4.1 DES.

DES (*Data Encryption Standard*) es un algoritmo de cifrado simétrico, escogido como un estándar FIPS[9] en los Estados Unidos en 1976.

Algunas características del DES son las que a continuación se presentan:

[9] FIPS: Estándares Federales de Procesamiento de la Información.

- Algoritmo simétrico más extendido mundialmente.
- Codifica bloques de 64 bits empleando claves de 56 bits.
- Consta de 16 rondas, más dos permutaciones, una que se aplica al principio y otra al final, tal así que la última es la inversa de la primera.
- Para descifrar basta con usar el mismo algoritmo empleando el orden inverso.

Hoy en día, DES se considera inseguro para muchas aplicaciones. Esto se debe principalmente a que el tamaño de clave de 56 bits es corto; las mismas se han roto en menos de 24 horas. Existen también resultados analíticos que demuestran debilidades teóricas en su cifrado, aunque son inviables en la práctica. [9]

1.4.2 AES.

AES (*Advanced Encryption Standard*, también conocido como *Rijndael*) es un esquema simétrico de cifrado por bloques adoptado como un estándar por el gobierno de los Estados Unidos, fue desarrollado por dos belgas, Joan Daemen y Vincent Rijmen, ambos estudiantes de la *Katholieke Universiteit Leuven*. Este algoritmo fue anunciado por el Instituto Nacional de Estándares y Tecnología (NIST, por sus siglas en inglés) el 26 de noviembre de 2001 después de un proceso de estandarización que duró 5 años.

Como peculiaridad tiene que todo el proceso de selección, revisión y estudio tanto de este algoritmo se efectuó de forma pública y abierta, por lo que, toda la comunidad criptográfica mundial ha participado en su análisis, lo cual convierte a *Rijndael* en un algoritmo perfectamente digno de la confianza de todos.

Características de AES:

- Diseñado para manejar longitudes de clave de 128 a 256 bits.
- Sus longitudes de bloque son variables comprendida entre 128 y 256 bits.
- Es un algoritmo resistente al criptoanálisis tanto lineal como diferencial.
- Es uno de los algoritmos más seguros en la actualidad.
- Hasta hoy no se ha podido romper. [9]

1.4.3 Triple DES.

Triple DES es el algoritmo simétrico que hace triple cifrado del DES, conocido también como TDES o 3DES y fue desarrollado por IBM[10] en 1998.

Cuando se descubrió que una clave de 56 bits no era suficiente para evitar un ataque de fuerza bruta, TDES fue elegido como forma de agrandar el largo de la clave sin necesidad de cambiar de algoritmo de

[10] IBM: International Business Machines (IBM) es una empresa multinacional estadounidense de tecnología y consultoría.

cifrado. Este método de cifrado es inmune al ataque por encuentro a medio camino, doblando la longitud efectiva de la clave (112 bits), pero en cambio es preciso triplicar el número de operaciones de cifrado, haciendo este método de cifrado muchísimo más seguro que el DES. Por tanto, la longitud de la clave usada será de 192 bits, aunque su eficacia sólo es de 112 bits.

El Triple DES está desapareciendo lentamente, siendo reemplazado por el algoritmo AES. Sin embargo, la mayoría de las tarjetas de crédito y otros medios de pago electrónicos tienen como estándar el algoritmo Triple DES (anteriormente usaban el DES). Por su diseño, el DES y por lo tanto el TDES son algoritmos lentos. El algoritmo AES puede llegar a ser hasta 6 veces más rápido. [9]

1.4.4 RC5.

RC5 es un algoritmo simétrico, una unidad de cifrado por bloques, notable por su simplicidad y diseñada por Ronald Rivest[11] en 1994, sus siglas se deben al inglés de "Cifrado de Rivest".

Algunas características de este algoritmo son las siguientes:

- RC5 tiene tamaño variable de bloques (32, 64 o 128 bits).
- Tamaño de clave entre 0 y 2040 bits.
- Número de vueltas entre 0 y 255.
- La combinación sugerida originalmente era de bloques de 64 bits, claves de 128 bits y 12 vueltas.
- Las rutinas de cifrado y descifrado pueden ser especificadas en pocas líneas de código.
- La programación de claves es complicada.

Mediante la programación distribuida este algoritmo se ha podido romper para claves de 54 y 56 bits; ya en febrero de 2010 se estaba trabajando en romper cifrados con claves de 72 bits. [9]

1.4.5 IDEA.

IDEA proviene del inglés *International Data Encryption Algorithm*, que significa algoritmo internacional de cifrado de datos. IDEA es un algoritmo simétrico de cifrado por bloques diseñado en la Escuela Politécnica Federal de Zúrich, Suiza, descrito por primera vez en 1991.

Características de IDEA:

- Opera con bloques de 64 bits usando una clave de 128 bits.
- Consiste de ocho transformaciones idénticas (cada una llamada ronda) y una transformación de salida (llamada media ronda).

[11] Ronald Rivest: Criptógrafo y profesor de ciencias de la computación en el departamento de ingeniería eléctrica y ciencias de la computación del Instituto Tecnológico de Massachusetts.

- El proceso para cifrar y descifrar es similar.
- El ataque por fuerza bruta resulta impracticable, ya que sería necesario probar 1038 claves.

Los diseñadores analizaron a IDEA para medir su fortaleza frente al criptoanálisis diferencial y concluyeron que es inmune bajo ciertos supuestos. No se han informado debilidades frente al criptoanálisis lineal o algebraico. Se han encontrado algunas claves débiles, las cuales en la práctica son poco usadas siendo necesario evitarlas explícitamente. Es considerado por muchos como uno de los cifrados en bloque más seguros que existen. [9]

1.4.6 Diffie – Hellman.

Diffie – Hellman es un protocolo de establecimiento de claves entre partes que no ha tenido contacto previo utilizando un canal inseguro y de manera no autentificada. Este protocolo criptográfico asimétrico fue diseñado por Whitfield Diffie[12] y Martin Hellman[13], a ellos debe su nombre.

Su seguridad radica en la extrema dificultad (conjeturada, no demostrada) de calcular logaritmos discretos en un cuerpo finito.

El protocolo es sensible a ataques activos del tipo *Man-in-the-middle*. Si la comunicación es interceptada por un tercero, éste se puede hacer pasar por el emisor de cara al destinatario y viceversa, ya que no se dispone de ningún mecanismo para validar la identidad de los participantes en la comunicación. [9]

1.4.7 RSA.

RSA es un sistema criptográfico de clave pública (asimétrico). Fue desarrollado en 1997 por Ronald Rivest, Adi Shamir[14]y Leonard Adleman[15], el nombre proviene de la primera letra del apellido de cada uno de sus inventores.

Las primeras versiones de PGP[16] lo incorporaban como método de cifrado y firma digital.

Características de RSA:

- La seguridad de RSA se basa en la dificultad para realizar la factorización de grandes números. Si se descubriera un método sencillo de factorización, RSA sería roto.
- Se le tiene como uno de los algoritmos asimétricos más seguros. [9]

[12] Whitfield Diffie: es un estadounidense pionero en la criptografía asimétrica.
[13] Martin Hellman: famoso por ser el inventor junto a Diffie de la criptografía de clave pública.
[14] Adi Shamir: matemático e informático, es uno de los descubridores del criptoanálisis diferencial, un método general para atacar cifrado por bloques.
[15] Leonard Adleman: profesor en ciencias de la computación y biología molecular de la Universidad del Sur de California.
[16] PGP: es el acrónimo de Pretty Good Privacy (Privacidad Bastante Buena), un programa que sirve para cifrar contenido y acceder a él mediante una clave pública y firmar documentos digitalmente para autentificarlos.

1.4.8 ElGamal.

ElGamal es un algoritmo de cifrado basado en problemas matemáticos de logaritmos discretos, diseñado en la idea de Diffie – Hellman, funciona de una forma parecida a este algoritmo discreto. Este algoritmo asimétrico puede ser utilizado tanto para generar firmas digitales como para cifrar o descifrar. Fue descrito por Taher Elgamal[17] en 1984.

La seguridad del algoritmo se basa en la suposición que la función utilizada es de un sólo sentido y la dificultad de calcular un logaritmo discreto.

Hasta el momento el algoritmo ElGamal de cifrado/descifrado puede ser considerado un algoritmo efectivo. Sin embargo existe un caso en que este algoritmo se vuelve maleable. Esto significa que bajo un ataque específico la seguridad de ElGamal se puede quebrar. Este ataque usa el hecho de tener el texto cifrado del texto claro (ambos conocidos). Sabiendo esto se puede llegar a que el texto cifrado corresponde al texto plano. Si la persona que cifró el mensaje anterior genera otro texto cifrado utilizando el mismo que cifró anteriormente) el adversario podría ser capaz de llegar al texto plano correspondiente. [9]

1.4.9 Firma Digital.

Una **firma digital**, pertenece a la criptografía de clave asimétrica, es una secuencia de caracteres que se añade a una pieza de información cualquiera, y que permite garantizar su autenticidad de forma independiente del proceso de transmisión tantas veces como se desee.

Este mecanismo implica el cifrado, por medio de la clave secreta del emisor, de una cadena comprimida de datos que se va a transferir. La firma digital se envía al receptor junto con los datos ordinarios. Este mensaje se procesa en el receptor para verificar su integridad. [10]

Propiedades de la firma digital:

- Sólo el usuario legítimo puede firmar su documento.
- Nadie puede falsificar una firma.
- Cualquiera puede verificar una firma digital.
- Una copia de una firma digital es igual a la original (Esto no es cierto para la firma escrita convencional).
- No se puede reutilizar una firma.
- No se puede modificar una firma.
- No se puede negar haber firmado un documento.

[17] Taher Elgamal: es un reconocido criptógrafo egipcio.

- No se puede alterar un documento después de haberlo firmado. [11]

Con la firma digital se logra:

- **Autenticidad:** La firma garantiza que las personas que intervienen son quienes dicen ser. Es decir, el contenido del mensaje se encuentra resguardado por medio de algoritmos matemáticos, salvaguardando la autenticidad del mensaje inicial.
- **Integridad:** Se logra verificar que el mensaje no fue modificado en el proceso de comunicación electrónica, garantizándose que el mensaje transmitido no se ha manipulado. Tratándose de la firma digital y/o electrónica, ésta se halla directamente relacionada con el documento por lo que cualquier cambio en el texto inhabilita la firma.
- **Confidencialidad:** El mensaje resulta secreto o confidencial para las partes de la comunicación electrónica, impidiendo que terceros, ajenos al mensaje, puedan conocerlo.
- **No Repudio:** La doctrina ha desarrollado la función del no repudio consistente en que el suscriptor no pueda negar que ha firmado digitalmente o electrónicamente el respectivo mensaje. [12]

1.4.10 Mecanismos de seguridad más adecuados.

Después de un análisis sobre los algoritmos criptográficos existentes, se arribó a la conclusión de que **AES** es el más indicado, debido a que DES es muy fácil de romper y Triple DES es mucho más lento; del algoritmo RC5 se han roto algunas de sus claves, por lo que no es del todo seguro, del IDEA se han encontrado claves débiles, y del AES se puede afirmar que es uno de los algoritmos más seguros que existen actualmente y aún no se ha podido romper, por eso se ha elegido éste como el algoritmo indicado para cifrar la información a transportar del sistema XABAL IDBIOACCESS al SPDI y viceversa.

Además, es de vital importancia proteger la clave que generará el algoritmo simétrico seleccionado por lo que se propuso utilizar un algoritmo asimétrico y de los examinados el más apropiado resultó ser **RSA**, porque Diffie - Hellman es sensible a ataques activos y ElGamal puede llegar a convertirse en un algoritmo de cifrado maleable, debido a esto su seguridad puede quebrar; es por ello que RSA es escogido y además, se reconoce como uno de los algoritmos asimétricos más seguro.

Debido a que la información se transportará por canales inseguros y se desconocerá si realmente su emisor fue quien la envió, es necesario firmarla digitalmente para garantizar la integridad y el no repudio.

1.5 Metodología, Herramientas y Tecnologías a utilizar.

Para el desarrollo de un software se deben tener en cuenta ciertos elementos importantes, como la metodología que regirá el ciclo de vida, las tecnologías y herramientas necesarias para su construcción. A

continuación se presenta el entorno de desarrollo definido por el proyecto Plataforma Modular de Identificación y Control de Acceso (PMICA), proyecto en el que se ha desarrollado el sistema XABAL IDBIOACCESS. Para la construcción de la solución se utilizará esta propuesta tecnológica y a continuación se podrá apreciar una síntesis de la misma realizada durante la investigación.

1.5.1 Metodología de desarrollo de software.

Las metodologías de desarrollo de software son un marco de trabajo empleado para estructurar, planificar y controlar el proceso de desarrollo del mismo. Éstas proporcionan un conjunto de técnicas, procedimientos, herramientas y soporte documental a la hora de desarrollar un producto.

Desarrollo Manejado por Rasgos (FDD).

La Metodología **Desarrollo Manejado por Rasgos** (FDD por sus siglas en inglés *Feature Driven Development*) es un enfoque ágil para el desarrollo de sistemas. Sin embargo, fue diseñado 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. [13]

Para el desarrollo del software, **FDD se divide en cinco fases**, las cuales son:

1. **Desarrollo de un modelo global:** Los expertos del dominio están al tanto de la visión, del contexto y de los requerimientos del sistema a construir. Éstos presentan un ensayo para mostrar a los miembros del equipo y al arquitecto jefe una descripción a alto nivel del sistema.
2. **Construcción de una lista de funcionalidades:** El equipo de trabajo identifica las funcionalidades, las agrupa y las prioriza. La lista de funcionalidades es revisada por los usuarios y patrocinadores para asegurar su validez.
3. **Planeación por funcionalidad:** Se crea un plan a alto nivel, en el que los conjuntos de funcionalidades se ponen en secuencia teniendo en cuenta su dependencia, prioridad y se asignan a los programadores jefes.
4. **Diseño por funcionalidad:** Se toma la funcionalidad a desarrollar, se identifican las clases involucradas, se hace una descripción de las mismas y sus métodos; el propietario de clases selecciona los correspondientes equipos ubicados por funcionalidad.
5. **Construcción por funcionalidad:** Cada propietario de clases construye los métodos para las funcionalidades y le realizan pruebas de unidad a cada una de las clases. Además, se inspecciona el diseño y el código, luego el propietario chequea el repositorio. Posteriormente se realiza una construcción principal donde la funcionalidad implementada se integra. [14]

La aplicación de FDD permite disminuir el riesgo de los proyectos, pues gracias a sus entregas tangibles y el constante monitoreo de su calidad asegura el firme avance de los mismos.

1.5.2 Lenguaje Unificado de Modelado (UML).

UML (también conocido como Lenguaje Unificado para la Construcción de Modelos) se define como un lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software. Es un sistema de notaciones destinado a los sistemas de modelado que utilizan conceptos orientados a objetos.

UML es un lenguaje para construir modelos, pero no guía al desarrollador en la forma de realizar el análisis y diseño orientados a objetos ni le indica cuál proceso de desarrollo adoptar. UML satisface necesidades importantes en software y desarrollo de aplicaciones, especialmente el modelado, que es una manera fácil para entender un sistema, ya que permite al desarrollador concentrarse en un cuadro grande. Este lenguaje ayuda a ver y a resolver los problemas más importantes, por lo que impide distraerse con muchos de detalles que serán usados más tarde. Diagramas y modelos debidamente construidos son las técnicas de comunicación eficaces que no sufren la ambigüedad del idioma hablado, y no molestan al espectador con datos abrumadores. [15]

El Lenguaje de Modelado Unificado proporciona a los arquitectos del sistema un fácil trabajo en el análisis y diseño de objetos con un lenguaje consistente para especificar, visualizar, construir y documentar los artefactos de sistemas de software, así como para el modelado de negocios. [16]

Con este lenguaje de modelado se construyeron los esbozos necesarios para llevar a cabo la construcción del módulo.

1.5.3 Visual Paradigm 8.0.

Visual Paradigm 8.0 es una herramienta UML, está diseñada para una amplia gama de usuarios, incluidos los ingenieros de software, analistas de sistemas, analistas de negocios y arquitectos de sistemas, o para cualquier persona que esté interesada en la construcción de sistemas de software fiables a gran escala con un enfoque orientado a objetos. Visual Paradigm 8.0 soporta los últimos estándares de la notación UML.

Visual Paradigm se dedica al desarrollo continuo y la entrega de software, servicios y alianzas para ayudar a los clientes a transformar con precisión sus necesidades de sistema en soluciones de software de alta calidad, todo ello con el mínimo riesgo y el máximo retorno de la inversión. Todos los productos de Visual Paradigm están diseñados y desarrollados para eliminar la complejidad, mejorar la productividad y comprimir el tiempo de desarrollo de software a los plazos fijados con los clientes. [17]

En esta herramienta de modelado se esbozaron los diagramas de clases, modelos de dominio, modelo de datos y diagramas de secuencia para mejor comprensión de la solución. Además, es ideal a la hora de mostrarles ideas a clientes o para poner en orden las ideas a la hora de comenzar una aplicación, evitando tener que diseñar todo el modelo desde cero en *Photoshop* o *Illustrator*, pues con esta versión del Visual Paradigm se modelaron los prototipos de interfaz de usuario correspondientes.

1.5.4 Lenguaje de Programación.

C# (leído en inglés "*C Sharp*" y en español "C Almohadilla") es un lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo, por lo que se convierte en sencillo e intuitivo.

A continuación se mencionan algunas de sus características:

- **Sencillez:** C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET.
- **Orientación a objetos:** Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos. Soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo.
- **Orientación a componentes:** La sintaxis de C# permite que se puedan definir propiedades, eventos y atributos; características que tienen que ser simuladas en otros lenguajes.
- **Seguridad de tipos:** Incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que permite evitar que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura.
- **Extensibilidad de operadores:** Permite redefinir el significado de la mayoría de los operadores incluidos los de conversión, tanto para conversiones implícitas como explícitas cuando se apliquen a diferentes tipos de objetos.
- **Eficiente:** En C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros.
- **Compatible:** Para facilitar la migración de programadores, permite incluir directamente en código escrito en C# fragmentos de código escrito en C, C++ o Java. [18]

Este lenguaje fue empleado para implementar todas las funcionalidades requeridas para la solución.

1.5.5 Entorno Integrado de Desarrollo (IDE).

Un **Entorno Integrado de Desarrollo** es un programa informático que puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.

Dentro de sus componentes están:

- Un editor de texto.
- Un compilador.
- Un intérprete.
- Un depurador.
- Un cliente.

Visual Studio 2010 (VS).

Visual Studio 2010 es una suite de lenguajes y herramientas que permite desarrollar software para la plataforma .NET. [19] Es un conjunto de herramientas en una sola aplicación que ayuda a los usuarios a escribir los programas. [20] Es el entorno de desarrollo más conocido para el diseño de aplicaciones en los sistemas operativos de Microsoft. [21] VS es un conjunto completo de herramientas para la creación tanto de aplicaciones de escritorio como de aplicaciones web empresariales para trabajo en equipo. [22]

Visual Studio 2010 integra las principales herramientas de desarrollo de la plataforma .NET: *Visual Basic, Visual C#, Visual C++* y *Visual Web Developer.* Además, simplifica las tareas básicas de la creación, depuración e implementación de aplicaciones y para su instalación requiere en principio contar con el Framework .NET 4.0.

Visual Studio 2010 posee diversas características que a continuación se describen:

- **Código generado automáticamente:** VS incluye un conjunto de tipos de proyectos que se pueden elegir una vez que se inicia un nuevo proyecto. Además, genera automáticamente el código esqueleto que se puede compilar y ejecutar inmediatamente.
- **Experiencia de Codificación:** El editor VS optimiza su experiencia de codificación, pues gran parte de su código es coloreada, se pueden apreciar los consejos que aparecen a medida que se escribe y atajos de teclado para realizar una multitud de tareas.
- **Personalización y extensibilidad:** Se puede personalizar muchas partes del entorno de VS, incluyendo colores, opciones de edición y diseño. [20]

Otras de las características que VS ofrece:

- Permite programación en paralelo.
- Posee un explorador de arquitectura.

- Da soporte para varios lenguajes. [21]

1.5.6 .Net framework 4.0.

.Net framework 4.0 es el modelo de programación completo y coherente de Microsoft para crear aplicaciones que tengan una comunicación segura y la capacidad de modelar una gama de procesos de negocio. [23] Es una moderna y completa plataforma de desarrollo para todo tipo de programas y aplicaciones, ya sean software de escritorio, servidores, teléfonos móviles o consolas.

Algunas de las características que presenta .Net Framework 4.0 son las siguientes:

- Desarrolla y ejecuta aplicaciones para Windows.
- Permite depurar y reutilizar código.
- Diversas mejoras en el modelado y en el acceso a los datos.
- Añadidas innovaciones en los lenguajes de programación Visual Basic y C#.
- Es perfectamente compatible con los sistemas operativos Windows.
- La aplicación requiere como mínimo Windows XP SP3. [24]

1.5.7 Sistema Gestor de Base de Datos.

PostgreSQL 9.1 es un sistema de gestión de bases de datos objeto-relacional, desarrollado en la Universidad de California y su código fuente está libremente disponible. Es el sistema de gestión de bases de datos de código abierto más potente del mercado.[25] Es compatible con una gran parte del estándar SQL.

Ofrece muchas características modernas que a continuación se exponen:

- Consultas complejas.
- Claves externas.
- Disparadores.
- Vistas.
- Integridad transaccional.
- Control de concurrencia multiversión. [26]

Con esta versión del SGBD PostgreSQL se gestionó la base de datos que interviene la solución.

1.5.8 NHibernate 3.1.0.

NHibernate 3.1.0 es un marco que permite la comunicación con una base de datos relacional de una manera orientada a objetos. Se pueden almacenar (o como también se suele decir, "persistir") los objetos de una base de datos y cargar más adelante. NHibernate genera las sentencias SQL necesarias para

insertar, actualizar, eliminar y cargar datos. No es el único marco ORM[18] para .NET, pero es probablemente el más maduro y rico de todos.

NHibernate posee muchas características y a continuación se presentan las más importantes:

- NHibernate 3.1.0 contiene en su núcleo un proveedor LINQ[19] totalmente reescrito, que soporta gran parte del espectro completo de LINQ, y supera muchas de las limitaciones del proveedor LINQ anterior. El proveedor de LINQ anterior fue parte de las contribuciones de NHibernate y se basa en la API[20] de criterios. Sólo se admite un conjunto limitado de consultas LINQ.
- Se pueden definir las asignaciones en el código y agregarlos a la configuración de NHibernate, esto se denomina ConfORM. En esta API, se pueden definir reglas y excepciones para instruir a NHibernate en cómo crear nuestras asignaciones. El proceso de mapeo de conjunto es basado totalmente en convenciones. ConfORM es altamente extensible y fácilmente se pueden definir nuestras propias reglas y excepciones. [27]

Además de estas características, ha habido un conjunto de correcciones de errores y mejoras de estabilidad, por lo que con esta tecnología se mapearán los datos en el desarrollo de la solución.

1.5.9 Windows Presentation Foundation (WPF).

Windows Presentation Foundation es un conjunto de librerías para implementar aplicaciones interactivas.

WPF presenta muchos aspectos interesantes como son:

- Separación de apariencia y lógica.
- Soporte del patrón “orden” (*command*).
- Fácil conexión a fuentes de datos vía ligaduras (*bindings*).
- Simplificación de trabajo con objetos observables mediante propiedades de dependencia.
- Herencia de valores para propiedades por relación jerárquica entre componentes.
- Acceso directo a hardware gráfico, animaciones, personalización completa de componentes mediante plantillas, etcétera.

WPF propone separar apariencia de lógica y lo lleva al extremo de ofrecer una herramienta para diseñadores gráficos que se integra en el proceso de desarrollo. Cuando el programador crea una interfaz gráfica se concentra en los elementos desde el punto de vista lógico, en cómo se comunican entre sí y

[18] ORM es mapeo objeto-relacional (más conocido por su nombre en inglés, *Object-Relational Mapping*).
[19] LINQ (por sus siglas en inglés *Language-Integrated Query*) es un conjunto de características presentado en Visual Studio 2008 que agrega capacidades de consulta eficaces a la sintaxis de los lenguajes C# y Visual Basic.
[20] API, sus siglas provienen de *Application Programming Interface*, que significan Interfaz de Programación de Aplicaciones.

con los datos de la aplicación. Los ficheros generados son directamente accesibles con *Microsoft Expression Blend*, allí el diseñador encuentra una aplicación en la que es sencillo cambiar el aspecto visual de los elementos, aplicar efectos y diseñar animaciones. *Blend* es parte de la suite *Microsoft Expression*, que incluye más herramientas orientadas a diseñadores gráficos. [28]

Con este marco de trabajo para interfaz de usuario se crearon las interfaces gráficas del módulo.

1.5.10 XML.

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto, juega un papel fundamental en el intercambio de una gran variedad de datos. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información. [29]

Las tecnologías basadas en XML han supuesto un importante cambio en la forma de pensar de los arquitectos de sistemas. Gracias a las capacidades de interoperabilidad, distribución, integración e independencia de las plataformas y tecnologías subyacentes a las aplicaciones, XML se ha consolidado como el estándar para el intercambio de datos.

A continuación se mostrarán especificaciones desarrolladas para dotar los propios documentos XML de mayor seguridad.

XML Signature es un estándar desarrollado por el W3C[21]que nos permiten autenticar al remitente, asegurar la integridad de partes de los documentos XML transportados.

XML Encryption es un estándar también desarrollado por el W3C que describe el modo de utilizar XML para representar recursos de forma digital y codificada. La especificación está pensada para ser usada junto con la especificación de firmas digitales *XML Signature* y así poder firmar y cifrar el contenido de los documentos XML.

XML Encryption soporta diferentes algoritmos de cifrado. Para el cifrado de bloques se citan los siguientes:

- 3DES.
- AES128.
- AES256.
- AES192.

[21] W3C: El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web.

Gracias a *XML Encryption* se puede agregar confidencialidad a los documentos XML y con *XML Signature* se garantiza la integridad de la información enviada y se evita el repudio del envío de mensajes. [30]

1.6 Conclusiones parciales.

Con el estudio de sistemas no interconectados se identificaron diferentes vías para transportar la información entre sistemas sin comunicación directa.

Con los mecanismos de seguridad escogidos se protegerá la información a transportar y se garantizarán dos aspectos de la seguridad informática: **confidencialidad** e **integridad**.

Con el empleo de las herramientas, metodología y tecnologías seleccionadas se podrá llevar a cabo con éxito la realización del módulo.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.1 Introducción.

Para desarrollar un módulo que permita la comunicación de los sistemas no interconectados XABAL IDBIOACCESS y SPDI, lleva como primer paso presentar la propuesta de solución y la confección de un modelo de dominio para establecer un lenguaje común entre los interesados, seguidamente se conforma la arquitectura para apreciar de forma clara la relación existente entre las capas que intervienen en el desarrollo del módulo, se obtienen los requisitos, se describen para su posterior implementación y se presentan los diagramas de clases, diagramas de secuencia y modelo de datos de la solución propuesta.

2.2 Propuesta de solución.

Acorde a la investigación realizada y para dar solución al problema planteado, se propone realizar un módulo que permita la comunicación de los sistemas XABAL IDBIOACCESS y SPDI sin que estén conectados entre sí, donde la UCI podrá ofrecer un servicio de personalización de documentos de identificación a cualquier empresa que lo solicite.

El módulo tendrá dos componentes, uno ubicado en el XABAL IDBIOACCESS denominado “Componente para enviar a personalización” que está compuesto de tres opciones: “Enviar Solicitudes”, donde se listarán los trámites en estado “listo para impresión” y se podrán seleccionar los que se desean personalizar. Además, brindará la posibilidad de escoger por cuál vía se transportará la información: por correo electrónico, FTP o mediante un dispositivo de almacenamiento, este último lo trasladará un emisario hasta su destino. Otra de las opciones es “Configuración General”, donde se configurará el correo electrónico, el FTP y se podrá elegir el tipo de información que se desea imprimir de cada trámite en la credencial. La tercera y última opción es “Actualizar”, donde se actualizará la base de datos del XABAL IDBIOACCESS con la información proveniente del SPDI luego de estar impresos todos los trámites enviados.

El otro componente del módulo, nombrado “Componente para comunicación con el SPDI”, que se encargará de recibir y enviar información de respuesta para el XABAL IDBIOACCESS. Cuando recibe la información la prepara para que el sistema de personalización pueda utilizarla para imprimir las credenciales y una vez que el SPDI termina el proceso de personalización, genera la información de respuesta para el XABAL IDBIOACCESS, donde este componente se encargará también de prepararla y de enviarla.

La información en juego será guardada en ficheros XML y para evitar que sea alterada será protegida por los mecanismos de seguridad definidos en el capítulo anterior, los cuales son el algoritmo simétrico AES

para cifrar la misma, pero antes se firma digitalmente el fichero XML para garantizar el no repudio, y el algoritmo asimétrico RSA para cifrar la llave generada de AES.

2.2.1 ¿Cómo funcionará?

Una vez que el usuario seleccione los trámites que desea enviar a personalizar, seleccione alguna de las opciones de envío y oprima el botón “Aceptar”, el “Componente para enviar a personalización” creará un archivo XML con los trámites seleccionados, al final de la información se plasmará una firma digital para garantizar su autenticidad y el no repudio por parte de la entidad que lo envía. Posteriormente se cifrará el fichero con el algoritmo simétrico AES y la llave generada se cifrará con RSA. El sistema generará un inventario para llevar el control de lo enviado, el cual contendrá con el número del solapín, el id del solapín y el id de cada persona envuelta en los trámites. Además generará un reporte con la cantidad de trámites en juego, los nombres, apellidos y el id de las personas involucradas. Después de terminar estas operaciones, el sistema comprime la información generada y la envía según la opción señalada.

Si la información está en el buzón de correo destino, el usuario responsable lo descargará y lo guardará en un directorio específico, el mismo para poder cargarla abrirá desde el “Componente para comunicación con el SPDI” una ventana y seleccionará el archivo, el sistema lo cargará, lo descomprimirá, lo descifrará, comprobará su autenticidad a través de la firma, confeccionará las órdenes de impresión y las enviará al SPDI. El mismo proceso pasa para el FTP y el dispositivo de almacenamiento.

Una vez que el SPDI termina la personalización devolverá un información de respuesta, la misma contiene el id del trámite y el código de impresión del mismo, esto para validar que la credencial está lista. El “Componente para comunicación con el SPDI” crea un archivo XML con esa información resultante, al final de la misma firma digitalmente el XML, cifra el fichero con el algoritmo AES y la llave resultante con el algoritmo RSA, luego lo comprime y lo envía al XABAL IDBIOACCESS por la vía que estime conveniente el usuario. Cuando la información esté en el buzón de correo o en el servidor FTP o en el dispositivo de almacenamiento se aplica lo mismo que se menciona en el párrafo anterior, sólo que esta vez es el “Componente para enviar a personalización” y el usuario oprimirá el botón “Actualizar” y el componente actualizará la base de datos del sistema XABAL IDBIOACCESS con la información recibida.

2.3 Modelo de dominio.

Para un mejor entendimiento de la propuesta de solución se confeccionaron dos modelos de dominio uno para cada componente, los cuales permiten de manera visual mostrar a los usuarios los principales conceptos que intervienen en el negocio y establecer un vocabulario común entre desarrolladores, clientes e interesados.

A continuación se presenta el **"Modelo de dominio del Componente para enviar a personalización"** y la explicación de cada una de sus clases y en el **ANEXO 1** se podrá apreciar el **"Modelo de dominio del Componente para comunicación con el SPDI"**.

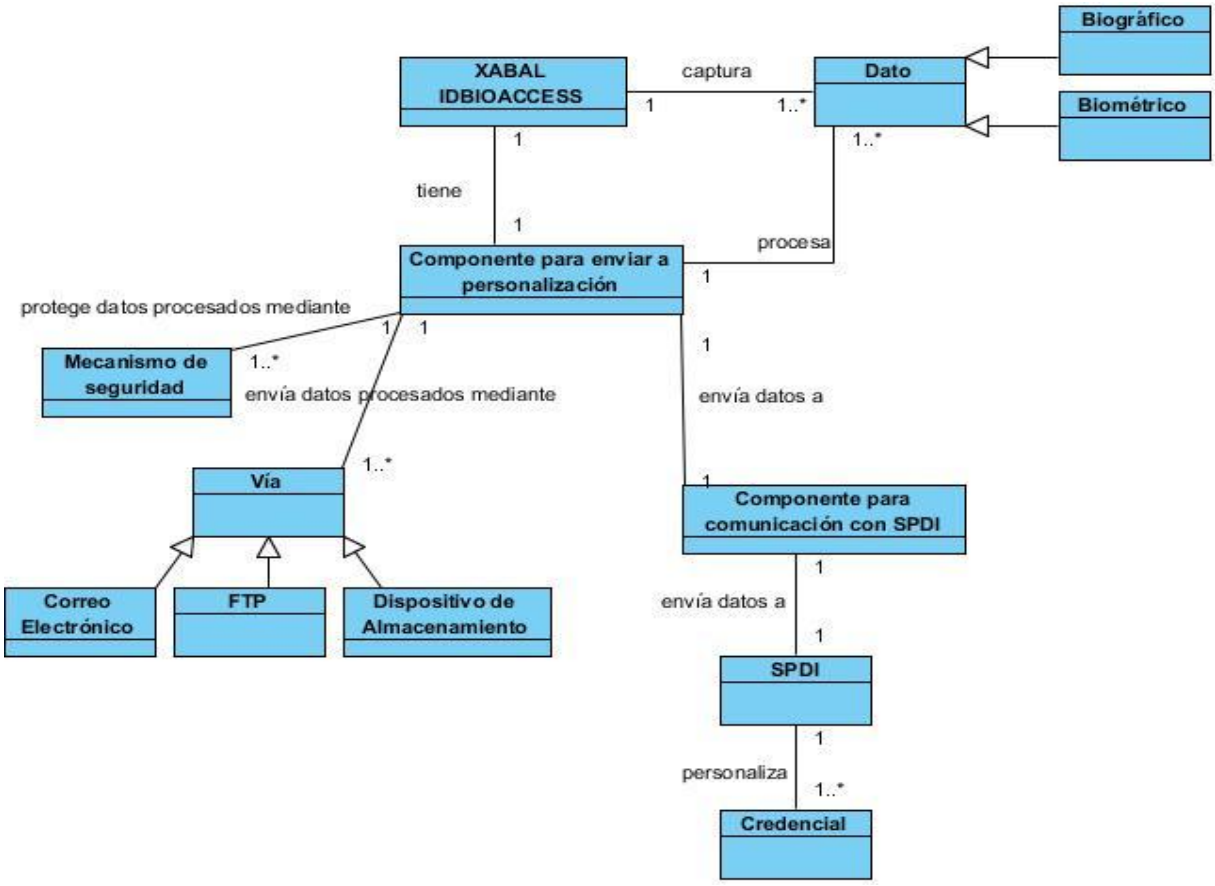


Figura 1: Modelo de dominio del Componente para enviar a personalización.

Componente para enviar a personalización: Es el componente que estará acoplado al sistema XABAL IDBIOACCESS, que permite seleccionar y preparar la información para ser enviada al Componente para comunicación con SPDI.

Componente para comunicación con el SPDI: Es el componente que se encarga de recibir y procesar la información para enviársela al SPDI, el cual realiza la impresión de la misma.

Correo Electrónico: Es un servicio que permite el intercambio de mensajes a través de sistemas de comunicación electrónicos. [31]

Credencial: Documento no muy grande de algún material duradero (plástico o cartón plastificado) que identifica a su poseedor y lo autoriza a acceder a cierto lugar.

Dato (biométrico o biográfico): Es la información que se le capta a las personas, que puede ser biográfica (nombre, apellidos, número de identidad, etc.) y biométrica (huella digital, imagen facial, firma, etc.).

Dispositivo de Almacenamiento: Unidad donde se puede guardar información (memoria flash, cd, disco duro y otros).

FTP: Abreviatura de *File Transfer Protocol*, protocolo para intercambiar archivos en Internet. El FTP utiliza los protocolos de Internet TCP/IP para permitir la transferencia de datos. Además se emplea principalmente para descargar un archivo de un servidor o para subir un archivo a un servidor a través de Internet. [32]

Mecanismo de seguridad: Técnica que se aplica a la información para mantener su integridad y confidencialidad.

SPDI: Sistema de Personalización de Documentos de Identificación, encargado de imprimir credenciales.

Vía: Medio por el cual se envía la información para que llegue a su destino.

XABAL IDBIOACCESS: Es el sistema encargado de realizar la captación de la información para la emisión de credenciales y es quien lleva el control de la entrega al titular correspondiente.

2.4 Arquitectura de la Solución.

El diseño de la arquitectura de un sistema es el proceso mediante el cual se define una solución para los requisitos técnicos y operacionales del mismo. Este proceso define qué componentes forman el sistema y cómo se relacionan entre ellos. El objetivo final de la arquitectura es identificar los requisitos que producen un impacto en la estructura del sistema y reducir los riesgos asociados con la construcción del mismo. La arquitectura debe soportar los cambios futuros del software, del hardware y de las funcionalidades demandadas por los clientes. [33]

El módulo se desarrolló aplicando la arquitectura por capas, debido a que la misma brinda flexibilidad y mejoras en las posibilidades de mantenimiento, es decir, como cada capa es independiente de la otra se pueden realizar cambios sin afectar la aplicación como un todo. El módulo se añade como una capa nueva para no afectar la arquitectura del sistema XABAL IDBIOACCESS y para que la solución pueda ser adaptable a otros sistemas. A continuación se evidencia una breve descripción de la misma:

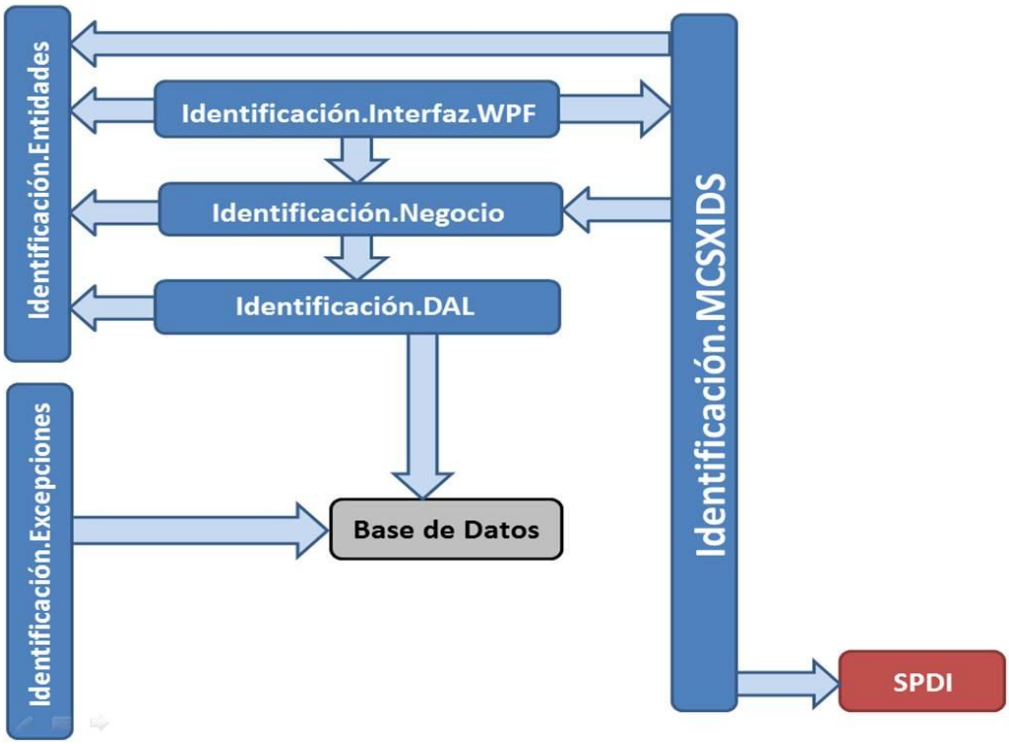


Figura 2: Arquitectura del Módulo para la comunicación de los sistemas no interconectados XABAL IDBIOACCESS Y SPDI.

Identificación.Interfaz.WPF: En esta capa se encuentran todas las interfaces que interactúan con el usuario. Es la encargada de presentar el sistema al usuario, comunicándole y capturando información en un mínimo de proceso. Esta capa se comunica con la capa inmediata inferior Identificación.Negocio.

Identificación.Negocio: Es la capa que encapsula la lógica del sistema, compuesta por clases que manejan las entidades y que valiéndose de la capa de acceso a datos da respuesta a las funcionalidades requeridas por el usuario, para mostrarle correctamente las interfaces a través de la capa de presentación.

Identificación.DAL: Es la capa de acceso a datos, es una porción de código que justamente realiza el acceso a los datos. De esta manera cuando es necesario cambiar el motor de base de datos, solamente se tiene que corregir esa capa.

Identificación.Entidades: En esta capa se encuentra la declaración de las entidades de la aplicación, de manera que se puede referenciar desde otras capas sin entrar en ciclos recursivos de compilación. Este esquema facilita la incorporación, en la capa de acceso a datos, de componentes tipo ORM que permiten "mapear" (representar) objetos en un esquema relacional. Esto funciona bien dado que las bases de datos más utilizadas son las bases de datos relacionales.

Identificación.Excepciones: En esta capa se encuentran todo el tratamiento de excepciones, cada vez que se lanza una excepción, éstas son almacenadas en la Base de Datos.

Identificación.MCSXIDS: Es donde se encuentra la solución de la presente investigación. Es la capa que permite la comunicación del sistema XABAL IDBIOACCESS y SPDI sin que se encuentren interconectados. Se integra al sistema XABAL IDBIOACESS mediante la capa Identificación.Negocio y hace uso de los servicios de comunicación del SPDI.

Base de Datos: Está constituida por todo el conjunto de tablas y procedimientos que permiten el almacenamiento de la información recolectada y procesada.

SPDI: Es el Sistema de Personalización de Documentos de Identificación, el mismo cuenta con una arquitectura específica que no es de interés para el desarrollo del módulo. La comunicación se realiza mediante dos servicios web que publica el SPDI, uno para el intercambio de información y otro para consultar el estado de los trámites.

2.5 Especificación de los requerimientos del sistema.

Para desarrollar un buen producto de software hay que tener en cuenta el paso más importante del proceso, la obtención de los requisitos o las exigencias por parte del cliente como también se les conoce. Los clientes suelen tener una noción de lo que quieren, una idea del resultado final; pero no entienden de las funciones que deben llevarse a cabo para que su software sea como lo pensaron. Detallar los requerimientos ayuda a los ingenieros a comprender el problema y así desarrollar una solución que satisfaga las necesidades de los interesados. Los requisitos se pueden clasificar en dos grupos: requisitos funcionales y requisitos no funcionales como se evidencia a continuación.

2.5.1 Requisitos Funcionales.

Los **requisitos funcionales (RF)** definen las funciones que el software debe realizar, estos requerimientos establecen el comportamiento del sistema teniendo en cuenta las entradas y las salidas. Para llevar a cabo la solución se identificaron los siguientes requisitos funcionales:

RF1. Configurar los datos a imprimir de la persona.

RF1.1 Mostrar los datos a imprimir de la persona.

RF1.2. Seleccionar los datos a imprimir de la persona.

RF2. Configurar el correo electrónico.

RF2.1 Insertar los datos para la configuración de correo electrónico:

- Nombre del usuario de correo.

- Dirección de correo del usuario.
- Servidor de entrada.
- Servidor de salida.
- Usuario.
- Contraseña.
- Dirección de correo destino.
- Asunto.

RF2.2 Modificar los datos para la configuración de correo electrónico.

RF3. Configurar FTP para envío de datos.

RF3.1 Insertar datos:

- Usuario.
- Contraseña.
- Dominio.
- URL.

RF3.2 Modificar datos.

RF4. Listar los trámites en estado “Listo para impresión”.

RF5. Seleccionar trámites en estado “Listo para impresión”.

RF6. Guardar la información en el FTP destino.

RF7. Guardar la información en un dispositivo de almacenamiento.

RF8. Enviar la información por correo electrónico.

RF9. Preparar información para enviar al SPDI.

RF10. Procesar información recibida del XABAL IDBIOACCESS.

RF11. Preparar información para enviar al XABAL IDBIOACCESS.

RF12. Procesar información recibida del SPDI.

2.5.2 Descripción de las funcionalidades.

Teniendo en cuenta lo que plantea la metodología FDD, se identifican los requisitos y se agrupan en funcionalidades para luego ser descritos de forma detallada, esclareciendo las acciones que deben seguir los desarrolladores para llegar a la mejor solución posible.

La Tabla 1 muestra la descripción detallada de la funcionalidad “Enviar a personalización”, el resto de las descripciones se pueden apreciar en el **ANEXO 2**.

Tabla 1: Descripción de la funcionalidad “Enviar Solicitudes”.

Precondiciones	El usuario debe estar autenticado en el sistema y debe poseer el permiso de “Enviar Solicitudes”.
Funcionalidades tratadas	RF4, RF5, RF6, RF7, RF8.
Conceptos tratados	Correo electrónico, dispositivo de almacenamiento, FTP, información.
Descripción básica	1. El sistema muestra un menú que contiene múltiples opciones, dentro de ellas las siguientes: • Configuración General. • Enviar Solicitudes. • Actualizar. 2 Si el usuario selecciona la opción “Enviar Solicitudes” el sistema muestra la interfaz “Enviar Solicitudes” (ver Figura 3), donde estarán publicados: • Las opciones de envío de información. • Lista de trámites en estado “Listo para impresión”. 2.1 El sistema mostrará el listado de los trámites en estado “Listo para impresión”, donde el usuario podrá seleccionar los trámites que desea imprimir. 2.2 En la sección de “Opción de envío” se encontrarán: • Correo electrónico. • FTP. • Dispositivo de Almacenamiento. 2.2.1 Si el usuario escoge la opción “Correo electrónico” y oprime el botón “Aceptar”: • El sistema comprueba que haya conexión con el servidor de correo. • El sistema prepara los trámites seleccionados para enviarlos al SPDI (ver **ANEXO 2**, Tabla 9) y les

	cambia el estado, de “listo para impresión” a “enviado a personalización”. • El sistema envía la información a su destino. • El sistema muestra el mensaje: “Enviado”. 2.2.2 Si el usuario escoge la opción de FTP y oprime el botón “Aceptar”: • El sistema comprueba que haya conexión con el servidor FTP. • El sistema prepara los trámites seleccionados para enviarlos al SPDI (ver **ANEXO 2**, Tabla 9) y les cambia el estado, de “listo para impresión” a “enviado a personalización”. • El sistema guarda la información en el FTP destino. • El sistema envía una notificación al correo destino indicando que la información fue guardada en el FTP. • El sistema muestra el mensaje: “Guardado en FTP”. 2.2.3 Si el usuario selecciona la opción de “Dispositivo de almacenamiento” y oprime el botón “Aceptar”: • El sistema prepara los trámites seleccionados para enviarlos al SPDI (ver **ANEXO 2**, Tabla 9) y les cambia el estado, de “listo para impresión” a “enviado a personalización”. • El sistema guarda la información en un destino predeterminado para su posterior traslado. • El sistema muestra el mensaje: “Guardado”.
Prototipo	

Figura 3: Interfaz "Enviar Solicitudes".

2.5.3 Requisitos No Funcionales.

Los **requisitos no funcionales (RNF)** son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. [34]

- **Usabilidad.**

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

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

RNF2: El módulo será distribuido en idioma español siguiendo las características del sistema XABAL IDBIOACCESS.

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

- **Restricciones de diseño e implementación.**

RNF4: Plataforma de desarrollo.NET 4.0 utilizando Visual Studio 2010.

RNF5: Para el acceso a datos se utilizará el ORM NHibernate 3.1.0.

RNF6: El módulo será implementado usando el lenguaje C#.

RNF7: Como sistema gestor de bases de datos se empleará PostgreSQL 9.1.

RNF8: Como marco de trabajo para interfaz de usuario se empleó Windows Presentation Foundation.

- **Seguridad.**

RNF9: Para que el módulo sea robusto y seguro se garantizará el tratamiento de excepciones.

RNF10: La información será protegida mediante algoritmos criptográficos.

- **Software.**

RNF11: El módulo acorde al sistema XABAL IDBIOACCESS deberá ser instalado en computadoras con sistema operativo Windows XP o superior.

- **Interfaz.**

RNF12: El módulo dispondrá de un diseño ameno e intuitivo.

- **Eficiencia**

RNF13: El módulo no debe demorar más de 2 segundos para cifrar un fichero XML con un total de 50 trámites.

2.6 Diagrama de Clases del Diseño.

Un **diagrama de clases** es empleado para representar y documentar el diseño. Además, sirve para visualizar las clases existentes en el sistema y las relaciones entre ellas. Un diagrama de clases está compuesto por los siguientes elementos: Clase (atributos, métodos y visibilidad) y Relaciones (Herencia, Composición, Agregación, Asociación y Uso).

Para llevar a cabo la solución se diseñaron dos diagramas de clases, uno del **Componte para enviar a personalización,** que se puede apreciar a continuación con una breve descripción de las clases que intervienen y otro del **Componente para comunicación con el SPDI**, que se puede observar en el **ANEXO 3**.

Las clases que intervienen en el **Diagrama de Clases del Componente para enviar a personalización** son las siguientes:

- **Correo:** Se emplea para enviar y recibir correos electrónicos.
- **Configuraciones:** Para cargar y guardar las configuraciones del sistema.
- **FTP:** Es para subir archivos al FTP.
- **Inventario:** Crea un archivo XML que contiene el identificador de la persona, el número del solapín y el identificador del solapín.
- **Principal:** Es la manejadora del flujo principal del módulo.
- **ProcesamientoImagen:** Se encarga de convertir la imagen a un arreglo de byte y viceversa.
- **Reporte:** Crea un documento en formato PDF con la cantidad de trámites enviados a personalización, los nombres, apellidos y el identificador de las personas involucradas en dichos trámites.
- **Seguridad:** Donde se implementa la lógica para utilizar los algoritmos criptográficos.
- **XML:** Clase que se encarga de crear y leer los archivos XML.
- **Zip:** La que permite comprimir y descomprimir los archivos generados.
- **Zlm:** Clase donde se genera el número de solapín, el código de barra y el código QR.

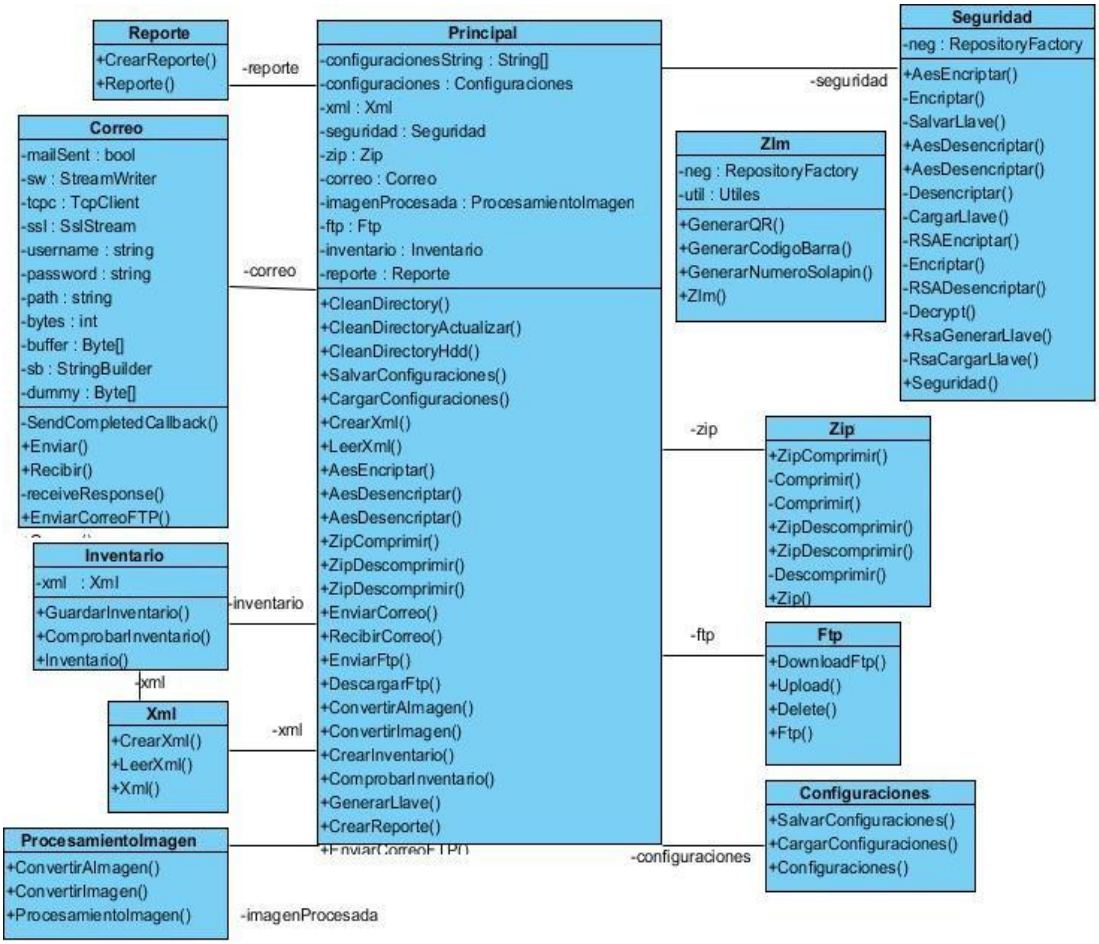


Figura 4: Diagrama de clases del Componente para enviar a personalización.

2.7 Diseño de Base de Datos.

Una **base de datos** es un fondo común de información almacenada en una computadora relativa a un mismo contexto, organizada de forma sistemática para servir a un propósito específico. La información debe estar disponible para cualquier persona o programa autorizado para acceder a ella. [35]

2.7.1 Modelo de Datos.

Para desarrollar el módulo se utilizó la base de datos del sistema XABAL IDBIOACCESS y se le agregaron las tablas **enrolamiento.ntipodatoopcional, enrolamiento.datributodatoopcional, enrolamiento.dvalordatoopcional** y **public.drsallave** para completar la solución. A continuación se expone un fragmento del modelo de datos y se describen brevemente las tablas empleadas (Ver Figura 5). En el **ANEXO 4** se encuentra el modelo de datos completo para un mejor entendimiento.

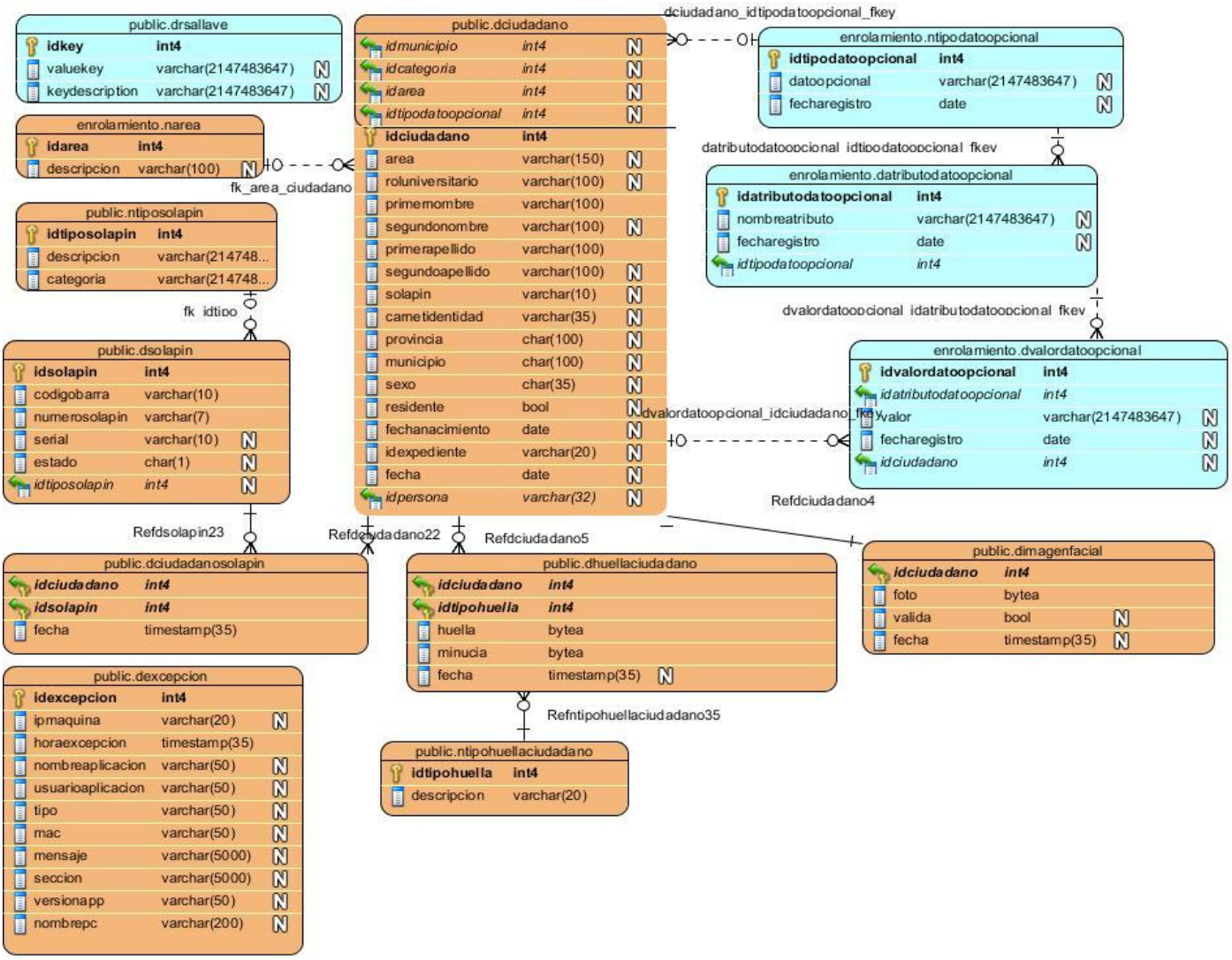


Figura 5: Fragmento del Modelo de Datos.

Las **tablas** empleadas son las siguientes:

- **public.dciudadano:** En esta tabla de la base de datos se registran todos los datos referentes a una persona de X entidad.
- **public.dhuellaciudadano, public.ntipohuellaciudadano:** En estas tablas se almacenan las huellas digitales de un ciudadano con todas sus particularidades.
- **public.dimagenfacial:** Está contenida en esta tabla la imagen facial que se le toma al ciudadano para su posterior impresión en la identificación.
- **public.dciudadanosolapin, public.dsolapin, public.ntiposolapin:** Contenido en estas tres tablas está el solapín del individuo, desde su número hasta su código.
- **public.dexcepcion:** Cuando el sistema da algún error, lanza también una excepción, la misma se almacena en esta tabla y en ella se guardan además el IP, la MAC y el nombre de la estación de trabajo donde se origina la excepción, la hora, el nombre de la aplicación, así como el usuario del sistema. También se registra el tipo de excepción y los detalles de la misma.
- **enrolamiento.narea:** Se guarda una descripción del área de trabajo a la que pertenece un ciudadano.
- **enrolamiento.ntipodatoopcional:** Almacena el tipo de dato del nuevo dato que se va a agregar, esto permite añadir nuevos datos que se quieran imprimir en la credencial.
- **enrolamiento.datributodatoopcional:** Guarda el nombre del tipo de dato que se ha almacenado en Enrolamiento.ntipodatoopcional.
- **enrolamiento.dvalordatoopcional:** Guarda el valor perteneciente al tipo de dato cuyo nombre fue almacenado en la tabla Enrolamiento.datributodatoopcional.
- **public.drsallave:** En la presente tabla se guardan las llaves para el cifrado, es decir, se almacenan la llave pública del “Componente para comunicación con el SPDI”, la pública y la privada del “Componente para enviar a personalización”.

2.8 Diagrama de Secuencia.

Se realizan **diagramas de secuencia** para detallar las acciones que se realizan en la aplicación. En un diagrama de secuencia se indican los elementos que forman parte del programa y las llamadas que se hacen en cada uno de ellos para realizar una tarea determinada.

A continuación se muestra el diagrama de secuencia diseñado para la funcionalidad “Configuración General” (Ver Figura 6). El resto de los diagramas se encuentran en el **ANEXO 5**.

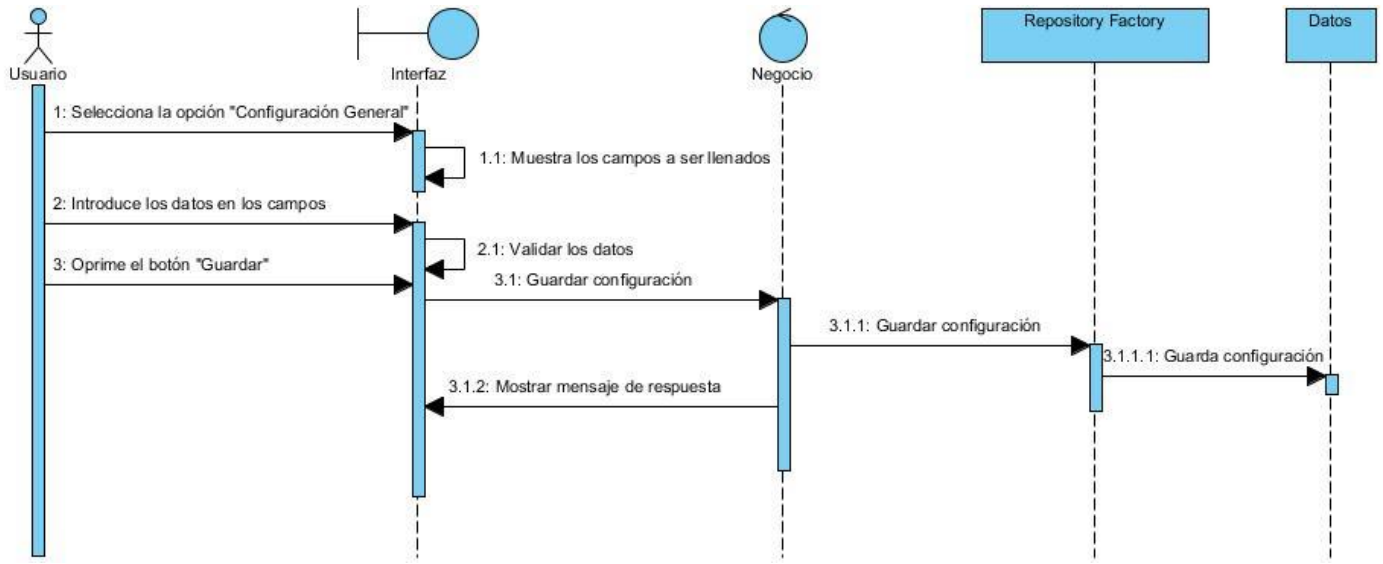


Figura 6: Diagrama de secuencia de la funcionalidad "Configuración general".

2.9 Patrones de Diseño.

El **patrón** es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas. [36]

Los patrones de diseño utilizados en la solución son los **GRASP**[22] y los **GOF**[23]. Los **patrones GRASP** describen principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Los **patrones GOF** se clasifican en 3 categorías: creacionales, estructurales y comportamiento. Los **patrones creacionales** están relacionados con los procesos de creación de objetos, los **estructurales** definen estructuras típicas entre clases y objetos, mientras que los **patrones de comportamiento** caracterizan la forma en que los objetos se distribuyen responsabilidades.

A continuación se describen algunos de los patrones antes mencionados y puestos en práctica en la solución:

- **Experto:** Es un patrón GRASP y su deber es asignar una responsabilidad al experto en información, es decir, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Ejemplo la clase **Configuración,** ella posee toda la información referente a las configuraciones tanto de Correo Electrónico como de FTP. (Ver Figura 7).

[22] GRASP: acrónimo que significa General Responsibility Asignment Software Patterns (patrones generales de software para asignar responsabilidades).
[23] GOF: acrónimo que significa Gang of Four, también conocidos como Pandilla de los Cuatro.

```
class Configuraciones
{
    public Configuraciones() { }

    public void SalvarConfiguraciones(List<string> configuraciones, string directorio)...

    public string[] CargarConfiguraciones(string directorio)...
}
```

Figura 7: Patrón Experto.

- **Creador:** Patrón GRASP que se encarga de asignarle a la clase B la responsabilidad de crear una instancia de clase A, como es el caso de la clase **Correo** que contiene un objeto de la clase **MailAddress**. (Ver Figura 8).

```
MailAddress to = new MailAddress(configuraciones[6]);
```

Figura 8: Patrón Creador.

- **Builder (constructor virtual):** Es un patrón creacional, que abstrae el proceso de creación de un objeto complejo, centralizando dicho proceso en un único punto. En la Figura 9 se presenta el constructor de la clase **Principal**.

```
public Principal()
{
    configuracionesClase = new Configuraciones();
    xml = new Xml();
    seguridad = new Seguridad();
    zip = new Zip();
    correo = new Correo();
    ftp = new AsynchronousFtpUploader();
    imagenProcesada = new ProcesamientoImagen();
    ftpAuxiliar = new Ftp();
    inventario = new Inventario();
    reporte = new Reporte();
    configSeguridad = new ConfigSeguridad();
}
```

Figura 9: Patrón Builder.

- **Facade (Fachada):** Es la clase definida que ofrece una interfaz común con un conjunto heterogéneo de interfaces. Un ejemplo de ello es la clase **Principal** que puede apreciarse en la Figura 10.

```
public class Principal
{
    string[] configuracionesString;
    Configuraciones configuracionesClase;
    Xml xml;
    Seguridad seguridad;
    Zip zip;
    Correo correo;
    AsynchronousFtpUpLoader ftp;

    public Principal()...

    public void SalvarConfiguraciones(List<string> configuracionesLista, string directorio)...

    public void CargarConfiguraciones(string directorio)...

    public void CrearXML(List<List<string>> datosAImprimir, string cantidadAImprimir)...

    public void AESEncriptar()...

    public void AESDesencriptar()...

    public void ZipComprimir()...

    public void ZipDescomprimir()...

    public void EnviarCorreo() ...

    public void RecibirCorreo() ...

    public void EnviarFtp() ...
}
```

Figura 10: Patrón Facade.

2.10 Conclusiones parciales.

La creación de los modelos de dominio permitió visualizar los principales conceptos que intervienen en el negocio para un mejor entendimiento de usuarios y desarrolladores.

Con la arquitectura especificada se identificaron los elementos que producen un impacto en la estructura del sistema y así reducir los riesgos asociados con la construcción del mismo.

La definición y descripción de los requisitos funcionales ayudaron al desarrollador a programar cada una de las funcionalidades pertinentes para que respondan a las necesidades del cliente.

Los patrones de diseño sirvieron de apoyo para desarrollar el módulo, utilizando soluciones probadas para problemas comunes.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

3.1 Introducción.

La implementación es un momento importantísimo en el desarrollo de un software, pues es donde se materializa el análisis y diseño. Para llevar a cabo este proyecto se hará necesario definir estándares de codificación para establecer un lenguaje común entre las personas que en él trabajan. Además, es preciso construir un diagrama de componentes para una mejor comprensión de los elementos que intervienen en la implementación. La confección del diagrama de despliegue ayuda a conocer el entorno donde se utilizará dicho módulo. Una vez que la implementación esté lista se realiza un conjunto de pruebas para demostrar que la solución es conforme con su especificación, que da respuesta a los requisitos del cliente, y así corroborar que se ha obtenido un resultado satisfactorio.

3.2 Diagrama de Componentes.

En los **diagramas de componentes** se muestran los elementos de diseño de un sistema de software. 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. [37]

Se diseñaron dos diagramas, uno para el componente que se encontrará desplegado junto al SPDI en la UCI, el mismo puede apreciarse en el **ANEXO 6** y el otro para el componente integrado al sistema XABAL IDBIOACCESS que se describe a continuación.

En la **Presentación** están las librerías **Identificación.Interfaz.WPF.dll** donde se encuentran las interfaces con las que interactúa el usuario y con **PresentationCore.dll** y **PresentationFramework.dll** se diseñaron las mismas. En el paquete **MCSXIDS,** en **Identificación.MCSXIDS.dll** es donde se encuentra la parte lógica de la solución desarrollada y se utilizó **ICSharpCode.SharpZipLib.dll** para la compresión de los archivos que contienen la información procesada por el módulo. **Identificación.MCSXIDS.dll** utiliza algunas de las entidades existentes en el sistema XABAL IDBIOACCESS a través de **Identificación.Entidades.dll** ubicada en el paquete **Entidades**. Utiliza además, **IdentificaciónNegocio.dll** que le permite contactar con el **Acceso a Datos**, especialmente con la librería **Identificación.DAL.dll** que a la vez emplea **NHibernate.dll**, **FluentNHibernate.dll** y **NHibernate.ByteCode.Castle.dll** para el mapeo de los datos. (Ver Figura 11).

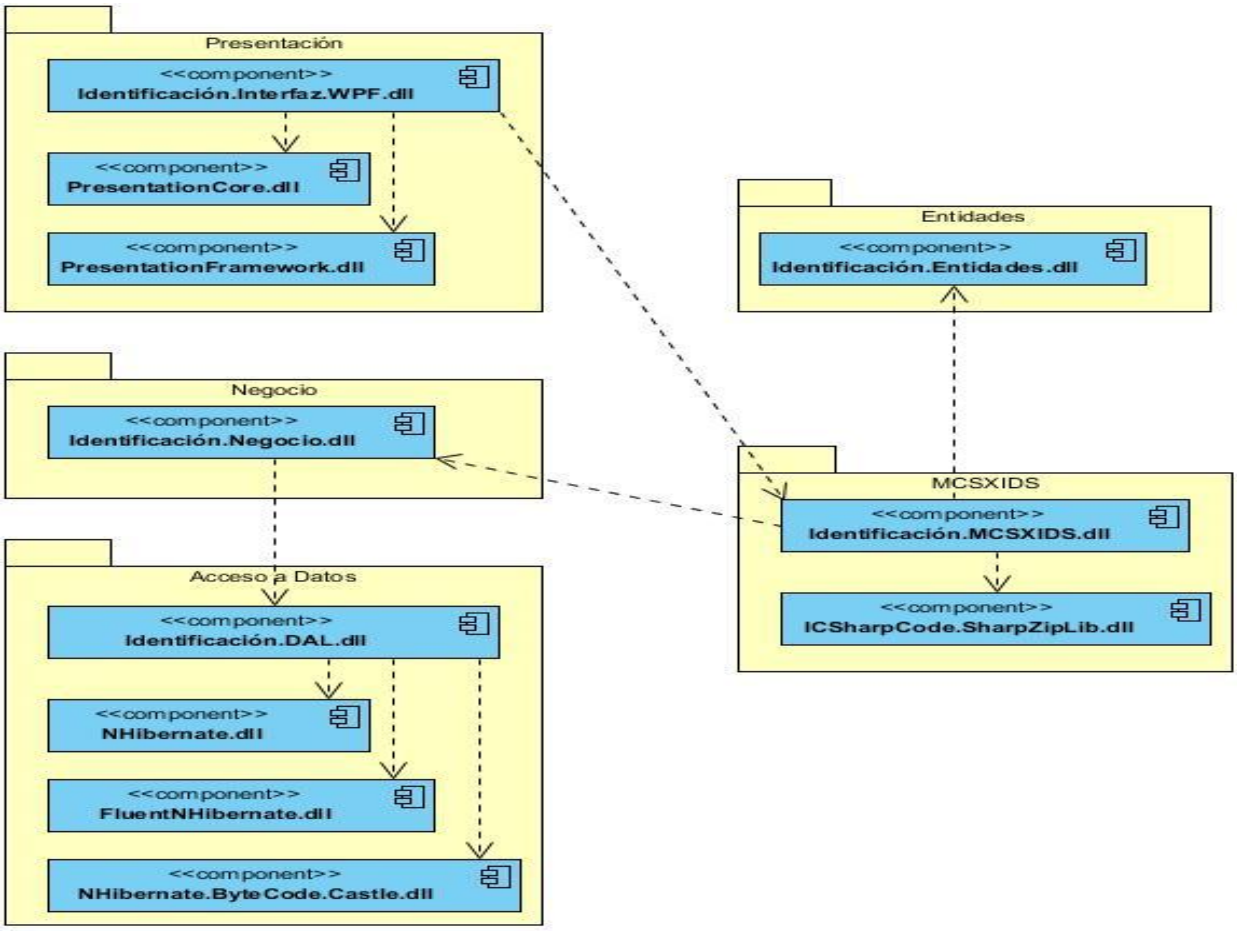


Figura 11: Diagrama de componentes del Componente para enviar a personalización.

3.3 Diagrama de Despliegue.

Un **diagrama de despliegue** es un diagrama estructurado que muestra la arquitectura del sistema desde el punto de vista de la distribución de los **artefactos** del software en los **destinos de despliegue**.

Los **artefactos** representan elementos concretos en el mundo físico que son el resultado de un proceso de desarrollo. Ejemplos de artefactos son archivos ejecutables, bibliotecas, archivos, esquemas de bases de datos, archivos de configuración, etc.

El **destino de despliegue** está generalmente representado por un nodo que es o bien de los dispositivos de hardware o bien algún entorno de ejecución de software. Los nodos pueden ser conectados a través de vías de comunicación para crear sistemas en red de complejidad arbitraria. [38]

El **diagrama de despliegue** perteneciente al módulo se puede apreciar en la Figura 12.

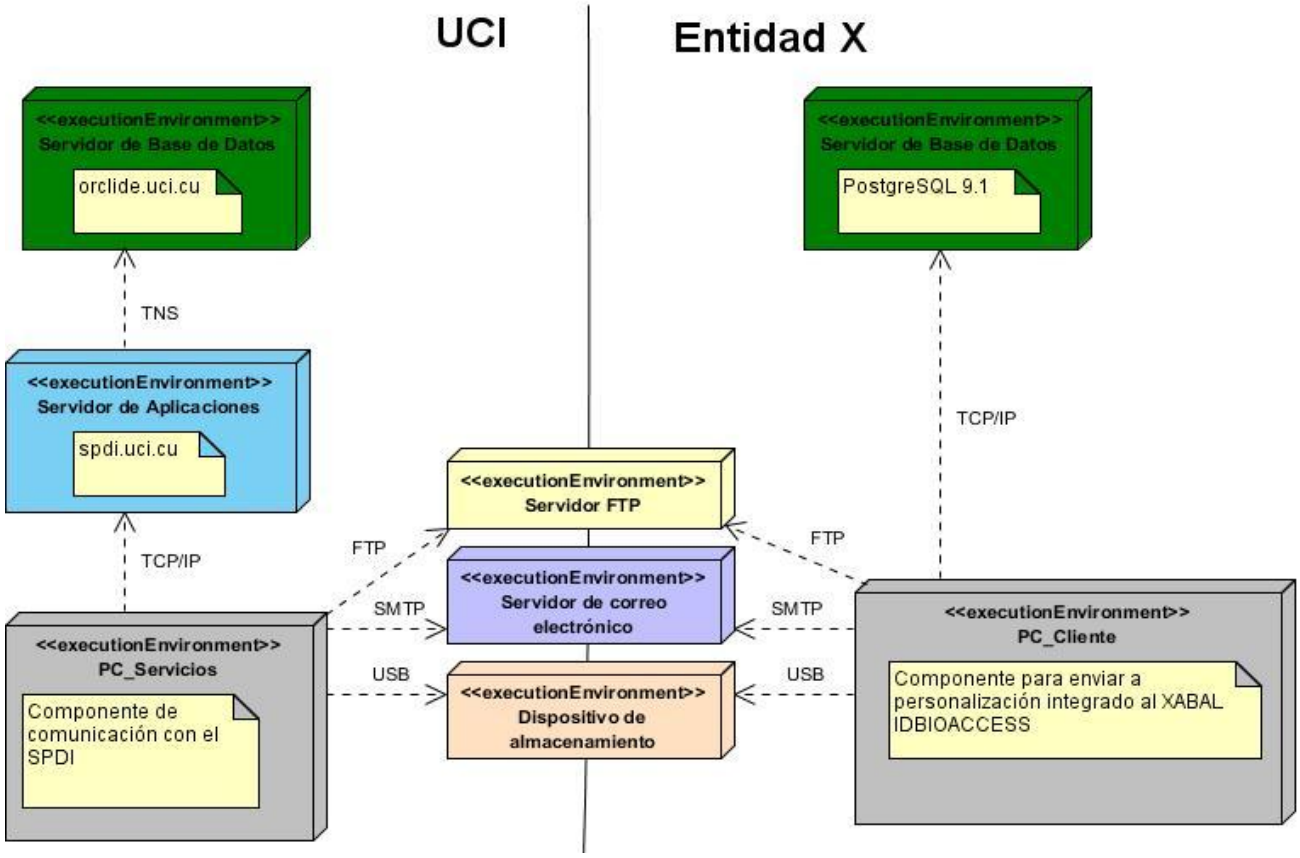


Figura 12: Diagrama de Despliegue.

En el diagrama se puede apreciar cómo quedará desplegado el módulo una vez que esté listo para su utilización. La Entidad X representa a las empresas interesadas en la adquisición del software, en ellas estará instalado en una estación de trabajo el sistema XABAL IDBIOACCESS e integrado a éste el **Componente para enviar a personalización**, que se conectará a un servidor de base de datos que emplea dicho sistema con PostgreSQL en su versión 9.1. En la sección UCI se encontrará el SPDI en el servidor de aplicaciones y conectado a éste una PC donde se estará instalado el **Componente de comunicación con el SPDI**. La comunicación entre la Entidad X y la UCI se establecerá mediante un servidor de correo electrónico, un servidor FTP o un dispositivo de almacenamiento.

3.4 Estándares de Codificación.

Cuando se lleva a cabo un proyecto es necesario definir estándares de codificación para establecer un lenguaje común entre las personas que en él trabajan y de esta manera el producto final tenga la calidad que requiere.

Un **estándar de codificación** comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo

práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, se debe establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada. Cuando el proyecto de software incorpore código fuente previo, o bien cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación debería establecer cómo operar con la base de código existente. [39]

El proyecto PMICA tiene sus propios estándares para que el código quede claro y entendible para todos sus miembros. A continuación se mostrarán algunos ejemplos de la aplicación de éstos y una síntesis de las convenciones que fueron aplicadas en la implementación del módulo. Para mayor información consultar el documento **"0120_51 CISED_PMICA_ID_UCI_Estándares de codificación para C# 1.0"** ubicado en el expediente del proyecto PMICA.

Ficheros.

El nombre de los ficheros debe coincidir con el nombre de las clases. Por ejemplo el fichero que contiene la clase Configuraciones, su nombre es **configuraciones.cs**, teniendo presente que sólo debe existir una clase por fichero (Ver Figura 13).

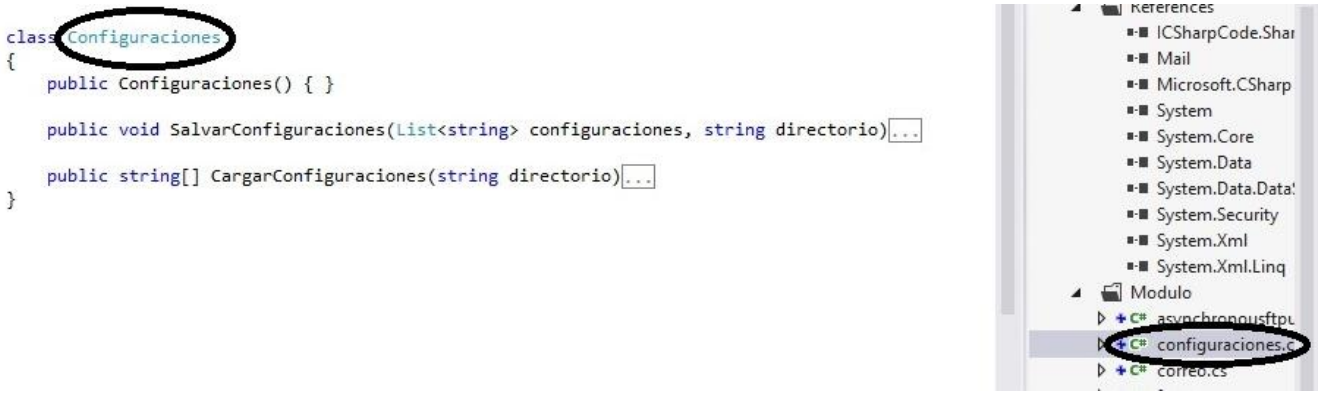


Figura 13: Estándar para Ficheros.

Llaves.

Las llaves deben ser utilizadas sobre líneas separadas y no sobre la misma línea como en **if, for**, etc. (Ver Figura 14).

Figura 14: Estándar para Llaves.

Comentarios.

Usar ***//*** o ***///*** para los comentarios. Evitar usar ***/*…*/***. (Ver Figura 15).

```
// Summary:
//     Este método es para guardar las configuraciones en un directorio específico
public void SalvarConfiguraciones(List<string> configuraciones, string directorio)...
```

Figura 15: Estándar para Comentarios.

Las convenciones utilizadas son **Pascal** y **Camel**, las cuales se explican a continuación.

- **Convención Pascal:** el primer caracter de cada palabra es en mayúscula y el resto en minúscula. Ejemplo: BackColor.
- **Convención Camel:** el primer caracter de cada palabra es en mayúscula (excepto la primera palabra) y el resto en minúscula. Ejemplo: backColor.

Tabla 2: Usos y ejemplos de las convenciones Camel y Pascal.

Usos	Convención Camel	Convención Pascal	Ejemplos

Nombre de las clases.		X	**class FtpState**
			{
			...
			}

Nombre de los métodos		X	**public void** Enviar (**string**[] configuraciones)
			{
			...
			}

Nombre de variables.	X		**private ManualResetEvent** wait;

3.5 Tratamiento de Errores.

El **tratamiento de errores** permite al programador controlar problemas que puedan ocurrir en la ejecución de un programa utilizando **excepciones**.

Las **excepciones** son un mecanismo de recuperación directa de fallos, llevan el control de errores en tiempo de ejecución y hacen que la aplicación continúe si se produce un error.

Para la gestión de las excepciones en el proyecto PMICA se desarrolló un módulo que publica las excepciones lanzadas, un publicador para base de datos y otro para registros que serán publicados en el

Windows Event Logs de la estación de trabajo donde está instalada la aplicación. En una tabla de la base de datos se guardan los datos de la excepción como el IP, la MAC y el nombre de la estación de trabajo donde se origina la excepción, la hora, el nombre de la aplicación, así como el usuario del sistema. Se registra también el tipo de excepción, la excepción o los detalles de la misma así como la versión de la aplicación en cuestión. Todos estos datos ayudan al equipo de desarrollo del sistema a resolver los errores pues tienen registrados los detalles del mismo, mientras que al usuario de la aplicación lo que se le muestra es un mensaje personalizado de la excepción y algunos detalles de la misma.

Para garantizar robustez en el módulo se hizo necesario tratar los errores de la siguiente manera:

- El tratamiento de las excepciones se realiza a través de las sintaxis *try {…} catch (Exception ex) {…}.*
- Los mensajes de error que emitirá el sistema se mostrarán en un lenguaje de fácil compresión para los usuarios.

3.6 Validación de la Propuesta de Solución.

A todo software después de terminado es necesario aplicarle un conjunto de pruebas para verificar si responde a los requisitos y validar si posee la calidad requerida.

El único instrumento adecuado para determinar el *status* de la calidad de un producto software es el **proceso de pruebas**. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. [40]

El **proceso de prueba** es una actividad en la cual se ejecuta un sistema o uno de sus componentes en circunstancias previamente especificadas, los resultados se observan, se registran y se realiza una evaluación de algún aspecto.

El **objetivo de las pruebas** no es asegurar la ausencia de defectos en un software, únicamente pueden demostrar que existen defectos en el software, por eso es necesario diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo posibles.

Una buena prueba debe centrarse en **dos objetivos**:

1. Probar si el software no hace lo que debe hacer.
2. Probar si el software hace lo que no debe hacer. [41]

Después de implementado el módulo se le aplicaron varias pruebas, los resultados de las mismas se podrán apreciar posteriormente.

3.6.1 Definición y descripción de las pruebas.

Pruebas Unitarias.

Las **pruebas unitarias** examinan cada subconjunto de la aplicación para garantizar que se implementaron de acuerdo con las especificaciones. Consiste además, en ejecutar un código fuente llamando directamente a los métodos de una clase, pasándole a estos los parámetros apropiados. Con las pruebas unitarias se consigue el aislamiento de las partes del código y la demostración de que estas partes no contienen errores. Es por ello que se consideran a las pruebas unitarias como uno de los tipos de pruebas más importantes que se le aplican a los software, logrando como resultado que disminuya en un gran porciento el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad. [42]

Pruebas de Caja Negra.

Las **pruebas de caja negra** son pruebas funcionales sin acceso al código fuente de las aplicaciones, se trabaja con entradas y salidas. Éstas se llevan a cabo sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa, para ello no es necesario conocer la lógica de la aplicación, únicamente la funcionalidad que debe realizar.

El funcionamiento de estos tipos de pruebas consiste en ver al componente como una “Caja Negra”, cuyo comportamiento sólo puede ser determinado estudiando sus entradas y las salidas obtenidas a partir de los casos de prueba. Para seleccionar el conjunto de entradas y salidas sobre las que trabajar, hay que tener en cuenta que en todo programa existe un conjunto de entradas que causan un comportamiento erróneo en nuestro sistema, y como consecuencia producen una serie de salidas que revelan la presencia de defectos. Entonces, dado que la prueba exhaustiva es imposible, el objetivo final es pues, encontrar una serie de datos de entrada cuya probabilidad de pertenecer al conjunto de entradas que causan dicho comportamiento erróneo sea lo más alto posible.

Las **pruebas de caja negra** pretenden encontrar estos tipos de errores:

- Funciones incorrectas o ausentes.
- Errores en la interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación. [42]

Un **caso de prueba** es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular. Los casos de prueba especifican una forma de probar el sistema, incluyendo las entradas con las que se ha de probar, los resultados esperados y las condiciones bajo las que ha de probarse. Se realizan con el fin de asegurar que el producto es operativo.

Con los **casos de prueba** de caja negra se pretenden demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma correcta.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene. [40]

Prueba de Integración.

La **prueba de integración** es una técnica sistemática para construir la estructura del programa mientras al mismo tiempo, se lleva a cabo pruebas para detectar errores asociados con la interacción. El objetivo es tomar los módulos probados en unidad y estructurar un programa que esté de acuerdo con el que dicta el diseño. La integración puede ser descendente si se integran los módulos desde el control o programa principal, o bien ascendente, si la verificación del diseño empieza desde los módulos más bajos y de allí al principal. La selección de una estrategia de integración depende de las características del software, y a veces del plan del proyecto; en algunos de los casos se puede combinar ambas estrategias. [43]

3.6.2 Aplicación de las Pruebas.

Pruebas unitarias.

Una vez implementado el módulo se procedió a la aplicación de pruebas unitarias para comprobar el correcto funcionamiento de las acciones del sistema o los posibles errores que pudieran existir.

En las Figuras 16 y 17 se muestran un ejemplo de la aplicación de las pruebas unitarias y el resultado arrojado:

```
namespace Identificacion.MCSXIDS.Test
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void CargarConfiguracionesTest()
        {
            Principal principal = new Principal();
            string directorio = @"..\..\config\configCount.txt";
            principal.CargarConfiguraciones(directorio);

            Assert.IsTrue(true);
        }
    }
}
```

Figura 16: Prueba Unitaria "CargarConfiguracionesTest".

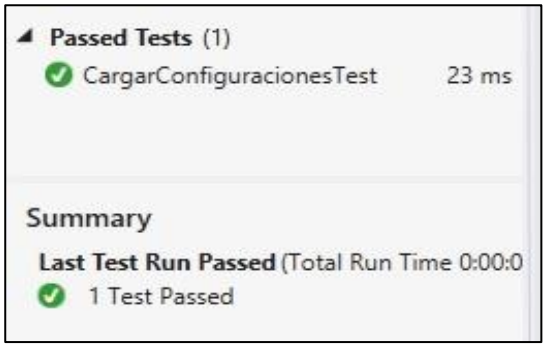


Figura 17: Resultado de la Prueba Unitaria "CargarConfiguracionesTest".

Pruebas de Caja Negra.

Para validar que el módulo funcione correctamente se definió un conjunto de casos de pruebas que reúnen la mayor cantidad de escenarios posibles. En la Tabla 4 se presenta el caso de prueba de la funcionalidad "Enviar a personalización". Los restantes casos de pruebas se encuentran en el **ANEXO 7**:

Tabla 3: Descripción de las variables del caso de prueba "Enviar Solicitudes".

No.	Nombre del Campo	Clasificación	Valor Nulo	Descripción
1	Lista de trámites	Lista	Sí	En caso de que no se haya seleccionado algún trámite, y se oprima el botón Aceptar, el sistema lanzará un mensaje de error.
2	Correo Electrónico	*RadioButton*	Sí	Puede ser nulo siempre y cuando se haya seleccionado la opción FTP o Dispositivo de Almacenamiento.

3	FTP	*RadioButton*	Sí	Puede ser nulo siempre y cuando se haya seleccionado la opción Correo Electrónico o Dispositivo de Almacenamiento.
4	Dispositivo de Almacenamiento	*RadioButton*	Sí	Puede ser nulo siempre y cuando se haya seleccionado la opción FTP o Correo Electrónico.

Tabla 4: Caso de prueba de la funcionalidad "Enviar Solicitudes".

Escenario	**Descripción**	**Lista de trámites**	**Correo electrónico**	**FTP**	**Dispositivo de Almacenamiento**	**Respuesta del Sistema**	**Flujo Central**
EC 1.1. Listado y selección de trámites en estado "Listo para impresión" para enviar por Correo electrónico.	El sistema deberá listar los trámites, y una vez seleccionados los mismos deberá enviarlos por correo electrónico.	**V**	**V**	N/A	N/A	El sistema devolverá una notificación de envío exitoso.	Opción "Enviar Solicitudes". Opción correo electrónico. Aceptar.
		V	**I**	N/A	N/A	El sistema devolverá mensaje de error.	
		I	**V**	N/A	N/A	El sistema devolverá mensaje de error.	
		I	**I**	N/A	N/A	El sistema devolverá mensaje de error.	
EC 1.2. Listado y selección de trámites	El sistema deberá listar los trámites, y una vez	**V**	N/A	**V**	N/A	El sistema devolverá una notificación de envío exitoso.	Opción "Enviar Solicitudes". Opción FTP

en estado "Listo para impresión" para enviar por FTP.	seleccionados los mismos deberá guardarlos en el FTP.	**V**	N/A	**I**	N/A	El sistema devolverá mensaje de error.	Aceptar.
		I	N/A	**V**	N/A	El sistema devolverá mensaje de error.	
		I	N/A	**I**	N/A	El sistema devolverá mensaje de error.	
EC 1.3. Listado y selección de trámites en estado "Listo para impresión" para ser transportada en un dispositivo de almacenamiento.	El sistema deberá listar los trámites, y una vez seleccionados los mismos deberá guardarlos en un directorio específico.	**V**	N/A	N/A	**V**	El sistema devolverá una notificación de envío exitoso.	Opción "Enviar Solicitudes". Opción Dispositivo de Almacenamiento. Aceptar.
		V	N/A	N/A	**I**	El sistema devolverá mensaje de error.	
		I	N/A	N/A	**V**	El sistema devolverá mensaje de error.	
		I	N/A	N/A	**I**	El sistema devolverá mensaje de error.	

Pruebas de Integración.

Aun cuando los módulos de un programa funcionen bien por separado es necesario probarlos conjuntamente, ya que un módulo puede tener un efecto adverso sobre otro.

Se eligió el enfoque ascendente para la realización de las pruebas de integración de la propuesta de solución. A continuación se muestra el Caso de Prueba de Integración "Enviar información al SPDI", el resto de los casos de pruebas pueden apreciarse en el **ANEXO 8**.

Tabla 5: Caso de Prueba de Integración "Enviar información al SPDI".

Caso de prueba: Enviar información al SPDI.
Sistema al que se integra: XABAL IDBIOACCESS.

Condiciones de ejecución: Que se haya hecho la captura de datos, los mismos se encuentren guardados en la base de dato del sistema y exista conexión con la misma.
Descripción de la prueba: Comprobar que el módulo es capaz guardar la información en un archivo XML, protegido con algoritmos criptográficos y además comprimido y que sea capaz de enviarlo por correo electrónico, de guardarlo en un FTP o en un directorio específico.
Entradas/Pasos de ejecución: Se listan los trámites, se seleccionan los que se desean enviar a personalización y la opción por la cual desean transportar la información (correo electrónico, FTP, dispositivo de almacenamiento) y por último se oprime el botón “Enviar”. El sistema genera un archivo XML con los trámites seleccionados, se firma digitalmente el XML, se cifra, se comprime y se envía por alguna de las vías destinadas.
Resultado esperado: El sistema es capaz de enviar la información por correo electrónico, de guardarla en un FTP o en un directorio específico.
Evaluación: Prueba satisfactoria.

3.6.3 Resultados de las Pruebas.

Pruebas unitarias.

Se realizaron tres iteraciones durante la ejecución de las pruebas unitarias y en la Tabla 6 se pueden apreciar los resultados arrojados.

Tabla 6: Resultados de las Pruebas Unitarias.

Iteraciones	Funcionalidades con errores	Funcionalidades correctas
1	5	7
2	2	10
3	0	12

En la Figura 18 se muestran gráficamente los resultados anteriores.

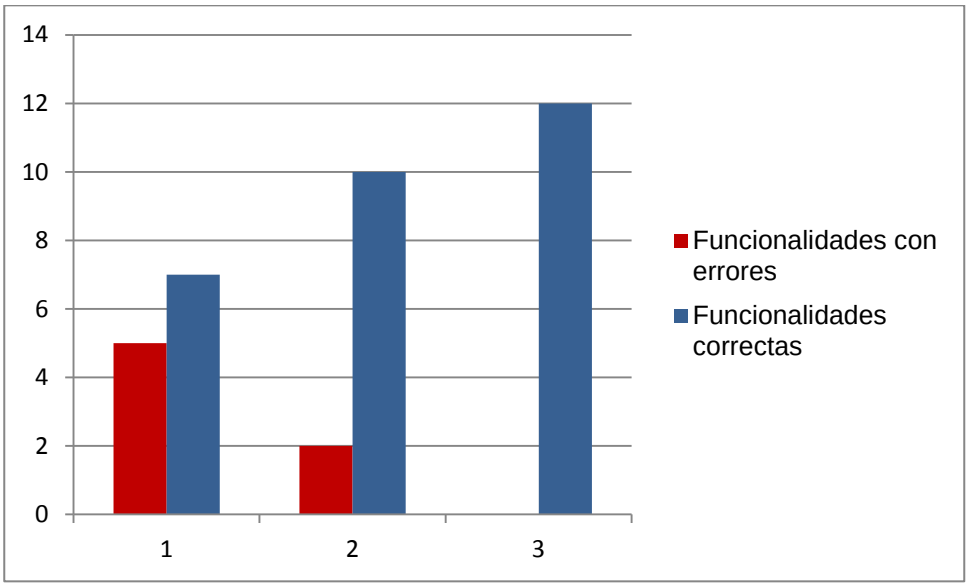


Figura 18: Gráfica de los resultados de las Pruebas Unitarias

Las pruebas unitarias permitieron detectar errores en la implementación de los métodos y gracias a las mismas se les pudo dar solución.

Pruebas Caja Negra.

Durante la realización de las pruebas de caja negra se identificaron varias no conformidades, las cuales fueron disminuyendo considerablemente con cada iteración realizada:

Tabla 7: Resultados de las Pruebas de Caja Negra.

Iteraciones	No Conformidades
1	**5**
2	**2**
3	**0**

En la Figura 19 se podrá apreciar una gráfica con los resultados anteriores.

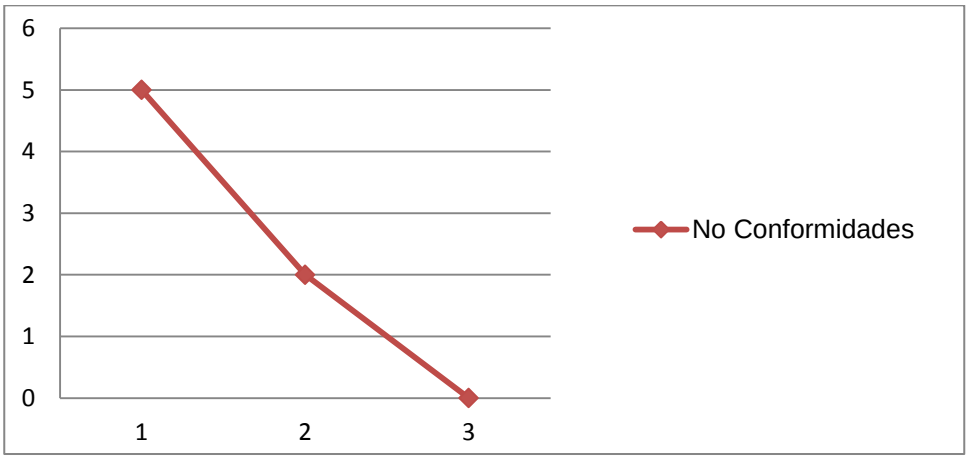


Figura 19: Gráfica de los resultados de las Pruebas de Caja Negra.

Al concluir la tercera iteración, las pruebas de caja negra permitieron comprobar que el módulo cumple con los requerimientos del cliente sin errores funcionales.

Pruebas de Integración.

Se realizaron 4 iteraciones para integrar los componentes del módulo, donde en la primera iteración se integraron 9 funcionalidades y al terminar la última iteración ya estaba integrado el módulo completamente. A continuación se muestran los resultados obtenidos.

Tabla 8: Resultados de las Pruebas de Integración.

Iteraciones	Cantidad de funcionalidades integradas
1	**9**
2	**10**
3	**11**
4	**12**

A continuación, en la Figura 20, se podrá apreciar una gráfica con los resultados anteriores.

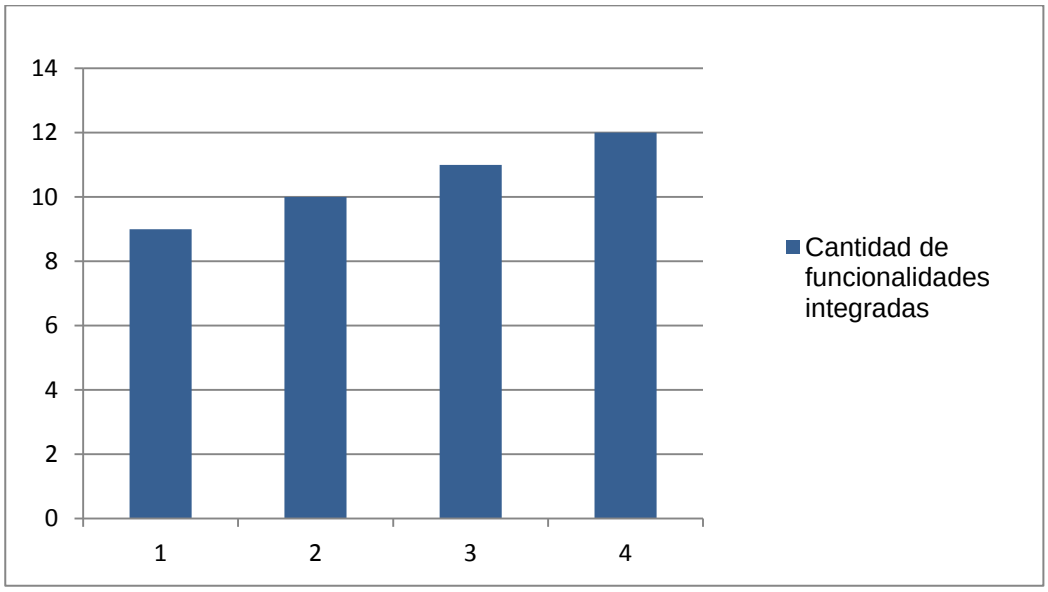


Figura 20: Gráfica de los resultados de las Pruebas de Integración.

Con la prueba de integración se pudo comprobar que los componentes desarrollados funcionan satisfactoriamente, acoplados a los sistemas XABAL IDBIOACCESS y SPDI.

Resultados generales de las pruebas.

Con la aplicación de las diferentes pruebas se demostró que el módulo cumple con los requisitos funcionales especificados y que permite la comunicación segura de los sistemas no interconectados XABAL IDBIOACCESS y SPDI. Además, la solución obtenida permitirá que las empresas del país puedan emitir sus credenciales a través del sistema XABAL IDBIOACCESS que la UCI les ofrece sin tener que adquirir el SPDI.

3.7 Conclusiones parciales.

A través de la implementación se pudo materializar una solución que responde a las especificaciones del cliente.

La definición de los estándares de codificación estableció una armonía de trabajo en el equipo de desarrollo, creando un lenguaje común entre sus miembros.

La aplicación de diferentes pruebas corroboró que la solución responde a los requisitos planteados por el cliente y validó que la misma funciona satisfactoriamente.

CONCLUSIONES GENERALES

Como resultado del trabajo realizado se arribó a las siguientes conclusiones:

- Con el estudio de sistemas no interconectados se detectaron diferentes vías para transportar la información entre sistemas sin comunicación directa.
- Con la aplicación de los mecanismos de seguridad seleccionados se protegió la información en su traslado y se garantizaron dos aspectos de la seguridad informática: confidencialidad e integridad.
- La confección de los modelos de dominio permitió una mejor visión del funcionamiento del negocio, lográndose un entendimiento común entre los implicados.
- La definición y descripción de los requisitos detallaron las particularidades que el sistema debe cumplir para satisfacer las necesidades del cliente.
- A través de la implementación se materializaron los requisitos establecidos para llevar a cabo el software.
- La aplicación de las pruebas posibilitó evaluar el producto, comprobándose el correcto funcionamiento del módulo y la ausencia de errores.
- El módulo desarrollado permite una comunicación segura entre los sistemas no interconectados XABAL IDBIOACCESS y SPDI, brindando la posibilidad a las empresas cubanas de utilizar el servicio de personalización que la UCI ofrece.

RECOMENDACIONES

Se recomienda:

- Implementar una funcionalidad que permita el fraccionado de ficheros según la capacidad del correo electrónico por el cual se vaya a enviar la información.
- Para la protección de la llave privada del “Componente para comunicación con el SPDI” emplear un dispositivo externo de almacenamiento seguro.

BIBLIOGRAFÍA REFERENCIADA

1. MBCEStore. *Datacard ID Works*. 2006 [cited 10 de diciembre 2013]; Available from: http://www.mbcestore.com.mx/datacard/id-works.htm.

2. Bundesdruckerei, *Especificación Técnica de la Solución para la Personalización de Pasaportes Electrónicos*. p. 8.

3. Inc., M.A.C. *Sistemas de emisión de credenciales 3M*. 2006 [cited 17 de diciembre 2014]; Available from: http://solutions.3m.com.co/wps/portal/3M/es_CO/SSD_LA/Security_Systems/Product/One/.

4. DATYS. *Sobre Nosotros*. 2012 [cited 21 de mayo 2014]; Available from: http://www.datys.cu/wpsobrenosotros.aspx.

5. DATYS. *Emisión de Pasaporte*. 2012 [cited 18 de diciembre 2013]; Available from: http://www.datys.cu/wpinfproducto.aspx?5.

6. DATYS. *Emisión del Carné del Marino*. 2012 [cited 18 de diciembre 2013]; Available from: http://www.datys.cu/wpinfproducto.aspx?72.

7. Sécurité, S.D., *Especificaciones de la interfase NIST*, 2005. p. 46.

8. Martí, J.R.E. *Seguridad en la red. Criptografía*. 2006 [cited 21 de enero 2014]; Available from: http://www.seguridadenlared.org/es/index25esp.html.

9. Lobo, B.M.G., *Algoritmos criptográficos*, in *Departamento de Redes*2010, Universitat de Valencia: Valencia. p. 36.

10. Jordi Forné, J.L.M.y.M.S. *Criptografía y Seguridad en Comunicaciones*, in *Departamento Matemática Aplicada y Telemática* 2013, Universitat Politècnica de Catalunya.

11. Cámara, E.M., *Introducción a la Criptografía*, 2009, Universidad de Zaragoza. p. 112.

12. SELA, S.P.d., *Fundamentos de la firma digital y su estado del arte en América Latina y el Caribe.* 2012: p. 25.

13. Calabria, L., *Metodología FDD*, in *Cátedra de Ingeniería de Software* 2003, Universidad ORT Uruguay. p. 19.

.:BIBLIOGRAFÍA REFERENCIADA:.

14. Martínez, J.L.C. *Práctica 3 - Otras metodologías Ágiles*. Scribd. [cited 18 de noviembre 2013]; Available from: http://es.scribd.com/doc/84488342/Practica-3-Otras-metodologias-Agiles.

15. Chonoles, M.J., *UML 2 for Dummies*. 2003. 412.

16. Group, O.M., *OMG Unified Modeling Language Specification*. Vol. 1.3. 1999. 808.

17. Paradigm, V. *Visual Paradigm for UML 8.0 Released*. 2010 [cited 26 de noviembre 2013]; Available from: http://www.visual-paradigm.com/aboutus/newsreleases/vpuml80.jsp.

18. Seco, J.A.G., *El lenguaje de programación C#*. 307.

19. Luna, F., *Herramientas enriquecedoras para HTML5 y CCS3. La web a través de Visual Basic 2010.* Pixels&Code, 2011. 3: p. 44.

20. Mayo, J., *Microsoft Visual Studio2010. A Beginner's Guide*. 2010. 449.

21. Ferri-Benedetti, F. *Microsoft Visual Studio. El entorno de desarrollo más potente para Windows*. 2013 [cited 26 de noviembre 2013; Available from: http://microsoft-visual-studio.softonic.com/.

22. Network, M.D. *Visual Studio*. 2007 [cited 26 de noviembre 2013]; Available from: http://msdn.microsoft.com/es-es/library/52f3sw5c%28v=vs.90%29.aspx.

23. Microsoft. *.NET Framework 4*. 2011 [cited 25 de noviembre 2013]; Available from: http://descargar.cnet.com/Microsoft-NET-Framework-4/3000-10250_4-75450154.html.

24. Hernández, H. *.NET Framework 4 Español*. 2012 [cited 25 de noviembre 2013]; Available from: http://net-framework-4.malavida.com/.

25. PostgreSQL-es. *Sobre PostgreSQL*. 2010 [cited 21 de mayo 2014]; Available from: http://www.postgresql.org.es/sobre_postgresql.

26. SQL, P. *What is PostgreSQL?* 2009 [cited 27 de noviembre 2013]; Available from: http://www.postgresql.org/docs/9.1/static/intro-whatis.html.

27. Dentler, J., *NHibernate 3.0 Cookbook*. 2010, BIRMINGHAM - MUMBAI. 328.

28. Marzal, A., *Desarrollo de aplicaciones con .NET y WPF*, in *Departamento de Lenguajes y Sistemas Informáticos*2010, Universitat Jaume I: Decharlas. p. 29.

29. Española, O. *Guía Breve de Tecnologías XML*. 2010 [cited 7 de febrero 2014]; Available from: http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML.

30. Colsa, L.E.C.d., *Seguridad en XML*, 2005. p. 12.

31. De, D. *Correo Electrónico*. 2008 [cited 17 de enero 2014]; Available from: http://definicion.de/correo-electronico/.

32. Masadelante.com. *Definición de FTP*. 2013 [cited 21 de enero 2014]; Available from: http://www.masadelante.com/faqs/ftp.

33. Llorente, C.d.l.T., *Guía de Arquitectura N-Capas orientada al Dominio con .Net 4.0*. Kraiss Press ed. 2010. 433.

34. González, A.H., *REQUISITOS A PARTIR DEL MODELO DEL NEGOCIO*, 2005. p. 5.

35. Peralta, J.A.y.D., *Bases de Datos*, Universidad de Granada. p. 76.

36. Larman, C., *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. 1999. 536.

37. Microsoft. *Diagrama de componentes de UML*. [cited 14 de marzo de 2014]; Available from: http://msdn.microsoft.com/es-es/library/dd409390.aspx.

38. Sarmiento, J. *UML: Diagrama de despliegue*. 2013 [cited 14 de marzo 2014]; Available from: http://umldiagramadespliegue.blogspot.com/.

39. Microsoft. *Revisiones de código y estándares de codificación*. [cited 14 de marzo de 2014]; Available from: http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx.

40. Software, P.d. *Pruebas de software.Gestión de Calidad y Pruebas de Software*. 2005 [cited 18 de marzo 2014; Available from: http://www.pruebasdesoftware.com/laspruebasdesoftware.htm.

41. LSI, *Técnicas de Evaluación Dinámica.* 2010: p. 14.

42. UAL, *Técnicas de prueba*, 2012, Universidad de Almería. p. 10.

43. Conocimiento, C.D.d. *Pruebas de integración*. [cited 18 de marzo de 2014]; Available from: http://www.academica.mx/blogs/las-pruebas-integraci%C3%B3n-software.

44. QR, C. 2013 [cited 28 de mayo de 2014; Available from: http://www.codigos-qr.com/.

BIBLIOGRAFÍA CONSULTADA

ALEGSA. Definición de Encriptación. 2012 [cited 16 de diciembre 2013]; Available from: http://www.alegsa.com.ar/Dic/encriptacion.php.

Azcárate, E.Q. PostgreSQL. Cómo funciona una Base de Datos por dentro.

Barchini, G.E., Métodos "I + D" de la Informática. Revista de Informática Educativa y Medios Audiovisuales, 2005. 2: p. 9.

Barrientos, M.J.L., CRIPTOGRAFÍA. ALGORITMOS DE CRIPTOGRAFÍA CLÁSICA, in Facultad de Ingeniería 2007, Universidad Nacional Autónoma de México. p. 13.

Boch, J.R. y.I.J.y.G., El Lenguaje Unificado de Modelado. Manual de Referencia., ed. A. Wesley. 1998, California 528.

Boggs, W.y.M., Mastering UML with Rational Rose 2002. 2002. 714.

Bustos, G., Guía de Uso de la Herramienta CASE. Visual Paradigm Standard Edition Versión 8.0, 2010, Escuela de Ingeniería Industrial. p. 44.

Carrillo, N.L.S. Tipos de Documentos. 2010 [cited 10 de diciembre 2013]; Available from: http://www.monografias.com/trabajos82/tipos-de-documentos/tipos-de-documentos.shtml.

Cohen, E. Tutorial C# .NET Journal of Computing and Information Technology, 1999. 7.

Delgado, O.G. Requisitos y características de diferentes SGBD. 2012 [cited 26 de noviembre 2013]; Available from: http://garcia-delgado.blogspot.com/p/requisitos-y-caracteristicas-de.html.

Escalona, R.L.C., Estándares de codificación para C# para Plataforma Modular de Identificación y Control de Acceso, 2012, Universidad de las Ciencias Informáticas. p. 22.

Fowler, M., UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition. 3 ed. 2003: Addison Wesley. 208.

Garzás, J. La metodología ágil FDD. Además de Scrum hay otras metodologías ágiles. 2012 [cited 18 de noviembre 2013]; Available from: http://www.javiergarzas.com/2012/09/metodologia-agil-fdd-2.html.

Gómez, J. Sistema de gestión de bases de datos SQL. 2009. 13 de junio de 2012 [cited 27 de noviembre 2013]; Available from: http://postgresql.softonic.com/.

.:BIBLIOGRAFÍA CONSULTDA:.

González, R.A.H.L.y.S.C., El Proceso de Investigación Científica. 2011, Cuidad de la Habana: Editorial Universitaria. 110.

Gretel, A.y.A. Encriptación de Datos. 2007 [cited 16 de diciembre 2013]; Available from: http://encripdedatos.blogspot.com/.

Group, O.M., OMG Unified Modeling Language Specification. Vol. 1.5. 2003. 736.

Group, O.M., Unified Modeling Language: Superstructure. Vol. 2.0. 2005. 710.

Gutiérrez, D., Patrones de Diseño, 2010, Universidad de los Andes. p. 35.

IMADOC. Fabricación de tarjetas PVC. [cited 11 de diciembre 2013]; Available from: http://www.tarjetaspvc.com/tarjetas-pvc/.

Kreimer, R. Historia del Documento de Identidad. 2009 [cited 10 de diciembre 2013; Available from: http://www.oocities.org/filosofialiteratura/HistoriaDocumentoIdentidad.htm.

Martínez, J.L.C. Práctica 3 - Otras metodologías ágiles. Scribd. [cited 18 de noviembre 2013]; Available from: http://es.scribd.com/doc/84488342/Practica-3-Otras-metodologias-Agiles.

Mestres, J.P., Patrones de diseño orientados a objetos, in Dpto. Ingeniería del Software e Inteligencia Artificial, Universidad Complutense de Madrid. p. 24.

Malenkovich, S. ¿Por qué debemos cifrar nuestros datos? KASPERSKY 2013 [cited 13 de diciembre 2013]; Available from: http://blog.kaspersky.es/por-que-debemos-cifrar-nuestros-datos/.

Marañón, G.Á. Mecanismos de seguridad. 2000 [cited 16 de diciembre 2013]; Available from: http://www.iec.csic.es/criptonomicon/seguridad/mecanism.html.

Martínez, R. Sobre PostgreSQL. PostgreSQL-es. Portal en español sobre Postgre. 2010 [cited 27 de noviembre 2013]; Available from: http://www.postgresql.org.es/sobre_postgresql.

Muro, C., Las formas básicas del pensamiento. 2010.

Network, M.D. Visual Studio. 2007 [cited 26 de noviembre 2013]; Available from: http://msdn.microsoft.com/es-es/library/52f3sw5c%28v=vs.90%29.aspx.

OACI, Documentos de viaje de lectura mecánica, 2008. p. 124.

.:BIBLIOGRAFÍA CONSULTDA:.

Obando, I.P.A., Material de apoyo: Curso básico de administración del SGBD PostgreSQL, 2010, Universidad Nacional: Programa de las Naciones Unidas Para el Desarrollo. p. 65.

Pantaleo, G., Pruebas de Software, in Departamento de Capacitación 2011, it- Mentor. p. 21.

Peinado, F., LPS: Introducción a los Patrones de Diseño de Software, in Dpto. de Ingeniería del Software e Inteligencia Artificial, Universidad Complutense de Madrid. p. 23.

Pozo, G.M., Tema 6: Excepciones, in Dpto. de Ingeniería de Software e Inteligencia Artificial 2007, Universidad Complutense de Madrid. p. 10.

Schenone, M.H., Diseño de una Metodología Ágil de Desarrollo de Software, in Facultad de Ingeniería 2004, Universidad de Buenos Aires: Buenos Aires. p. 200.

Sécurité, S.D., Especificaciones del sistema AFIS Civil, 2005. p. 55.

Schenker, A.C.y.D.G.N., NHibernate 3. Beginner's Guide. 2 ed. 2011, BIRMINGHAM - MUMBAI. 368.

Sampieri, P.B.L.y.C.F.-C.y.R.H., Metodología de la Investigación. 4 ed. 2006, Iztapalapa, México D.F. 882.

Tello, J.C., Diagramas de Secuencia, in Dpto. Ciencias de la Computación, Universidad de Alcalá. p. 2.

UPM, Patrones del "Gang of Four", in Unidad Docente de Ingeniería del Software, Universidad Politécnica de Madrid. p. 57.

UNAM. Definición de Estándar. 2008 [cited 1 de abril 2014]; Available from: http://redyseguridad.fi-p.unam.mx/proyectos/biometria/estandares/estandar.html.

UAIP. Información confidencial. 2013 [cited 1 de abril 2013]; Available from: http://transparencia.guanajuato.gob.mx/faq_detalle.php?id_clasificacion=5.

Vilalta, J., UML: Guía Visual. 2001. 47.

Valdor, E.d.I.d.L.y.L.J.A.P. Dudas sobre el idioma español. 2013 [cited 18 de diciembre 2013]; Available from: http://www.juventudrebelde.cu/dudas-idioma/?tag=plastificar.

Zayas, D.C.Á.d., Metodología de la Investigación Científica, in Centro de Estudios de Educación Superior "Manuel F. Gran"1995, Universidad de Oriente: Santiago de Cuba. p. 80.

GLOSARIO DE TÉRMINOS

A

ANSI/NIST: Formato de datos para el intercambio de huellas dactilares, facial, etc.

C

Cifrado: Consiste en transformar un texto en claro mediante un proceso de cifrado en un texto cifrado, gracias a una información secreta o clave de cifrado.

CISED: Centro de Identificación y Seguridad Digital radicado en la UCI.

Código QR: Código de respuesta rápida (*quick response code*). Es un tipo de código de barras bidimensionales. A diferencia de un código de barras convencional, la información está codificada dentro de un cuadrado, permitiendo almacenar gran cantidad de información alfanumérica. [44]

Correo Electrónico: Es un servicio que permite el intercambio de mensajes a través de sistemas de comunicación electrónicos.

Credencial: Es un documento no muy grande de algún material duradero (plástico o cartón plastificado) que autoriza a su dueño acceder a cierto lugar.

D

Dispositivo de Almacenamiento: unidad en el que se puede guardar información (memoria flash, cd, disco duro y otros).

Documento de identificación: es un documento que contiene datos personales de un individuo, el mismo es emitido por un empleado público con autoridad competente para permitir la identificación personal.

E

Emisario: Persona encargada de trasportar el dispositivo de almacenamiento con la información a su destino.

Encriptación: Proceso para volver ilegible información considerada importante. La información una vez cifrada sólo puede leerse aplicándole una clave. Se trata de una medida de seguridad que es usada para almacenar o transferir información delicada que no debería ser accesible a terceros. Pueden ser contraseñas, números de tarjetas de crédito, conversaciones privadas, etc. El término encriptación es traducción literal del inglés y no existe en el idioma español. La forma más correcta de utilizar este término sería cifrado.

Excepciones: Errores que ocurren en procesos internos de la aplicación que deben ser tratados para no caer en estados inestables y brindar seguridad y confianza al usuario.

F

FTP: Abreviatura de File Transfer Protocol, el protocolo para intercambiar archivos en Internet. El FTP utiliza los protocolos de Internet TCP/IP para permitir la transferencia de datos. Además se emplea principalmente para descargar un archivo de un servidor o para subir un archivo a un servidor a través de Internet.

I

Información: Es un conjunto de datos procesados con un propósito específico.

Información confidencial: Es aquella información relativa al contenido esencial del derecho a la privacidad, del derecho a la intimidad. La que ponga en riesgo la vida, la integridad, el patrimonio, la seguridad o la salud de cualquier persona; o afecte directamente el ámbito de la vida privada de las personas; la que por mandato expreso de una Ley sea considerada confidencial o secreta.

M

Mecanismo de seguridad: Técnica que se aplica a la información para mantener su integridad y confidencialidad.

N

NTIC: son las Nuevas Tecnologías de la Información y las Comunicaciones, se utiliza este término para ubicar al lector de que se tratan de las últimas tecnologías existentes (telefonía móvil, robótica, Internet, etc.).

P

Plastificar, plasticar: El uso de ambas formas: plasticar y plastificar, es correcto. La voz plasticar constituye un cubanismo y aparece recogido en el Diccionario del español de Cuba en el 2000. Recubrir un documento, por ejemplo un carné, con una lámina de material plástico transparente, para evitar que se deteriore: ej. Si no plasticas el carné del gimnasio dentro de poco va a estar hecho un asco. El sinónimo de esta voz en el español general es plastificar según registra el Diccionario de la lengua española en el 2001: plastificar. 1. tr. Recubrir papeles, documentos, telas, gráficos, etc., con una lámina de material plástico.

PMICA: Plataforma Modular de Identificación y Control de Acceso.

PVC o Policloruro de Vinilo: es un moderno, importante y conocido miembro de la familia de los termoplásticos. Es un polímero obtenido de dos materias primas naturales cloruro de sodio o sal común (ClNa) (57%) y petróleo o gas natural (43%), siendo por lo tanto menos dependiente de recursos no renovables que otros plásticos.

S

SMTP: Es el protocolo simple de transferencia de correo, protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto

.:GLOSARIO DE TÉRMINOS:.

Software Robusto: Programa informático que puede continuar el proceso a pesar de problemas inesperados.

SPDI: Sistema de Personalización de Documentos de Identificación, encargado de imprimir credenciales y de enviar información de respuesta a XABAL IDBIOACCESS.

U

UCI: Universidad de las Ciencias Informáticas, ubicada en La Habana, Cuba.

V

Vía de transporte: Medio por el cual se envía la información para que llegue a su destino.

X

XABAL IDBIOACCESS: Es el sistema encargado de realizar la captación de los datos para la emisión de credenciales y es quien lleva el control de la entrega al titular correspondiente.

ANEXOS

ANEXO 1: Modelo de dominio del Componente para comunicación con el SPDI.

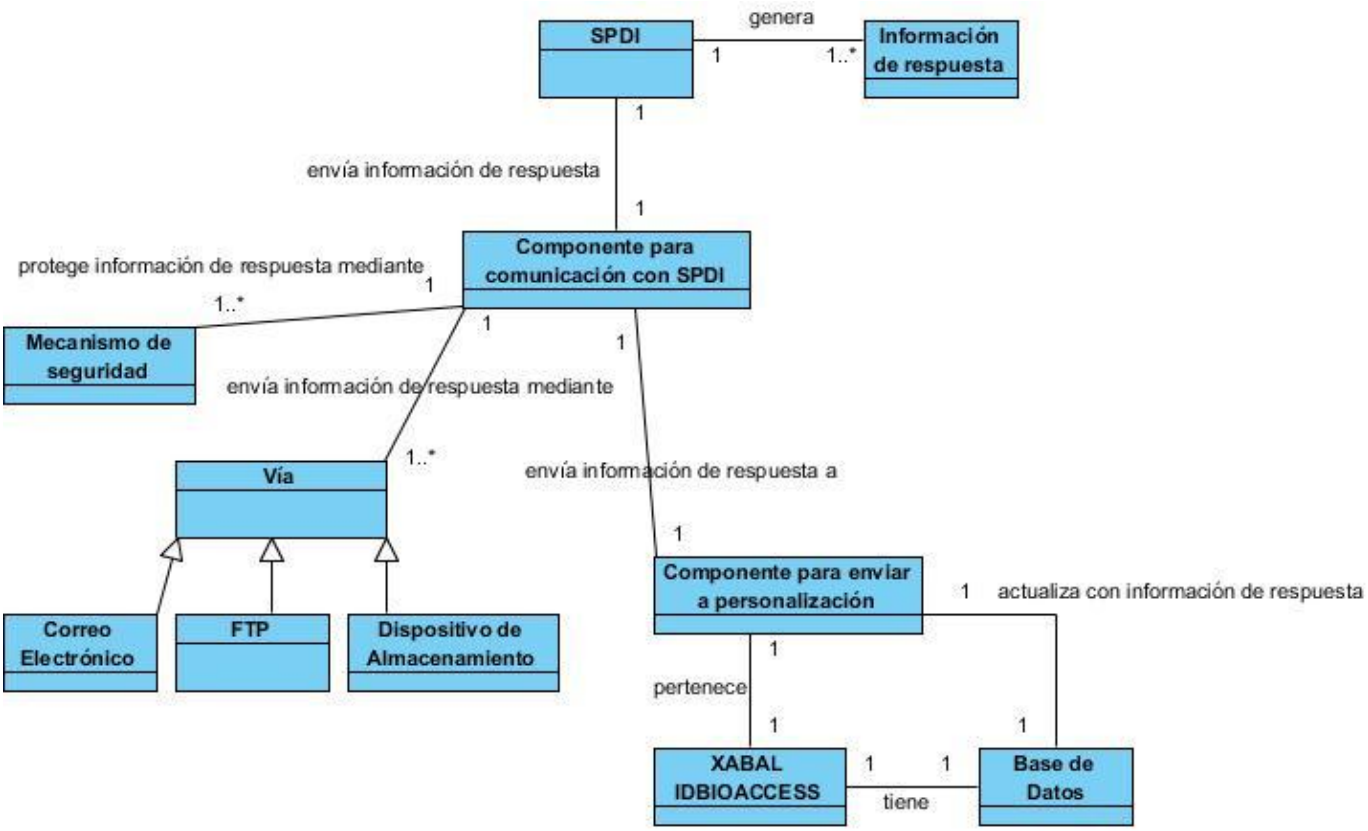


Figura 21: Modelo de dominio del Componente para comunicación con el SPDI.

ANEXO 2: Descripción de los requisitos funcionales del sistema.

Tabla 9: Descripción de la funcionalidad "Preparar información para enviar al SPDI".

Precondiciones.	El usuario debe oprimir el botón "Enviar".
Funcionalidades tratadas.	RF9.
Conceptos tratados.	Componente para enviar a personalización
Descripción básica.	Una vez que el usuario oprime el botón "Enviar": • El **Componente para enviar a personalización** genera un archivo XML con los trámites seleccionados y lo firma digitalmente. • El componente cifra el archivo XML utilizando el algoritmo simétrico AES. • El componente cifra la llave generada con el algoritmo AES utilizando la llave pública procedente del **Componente de comunicación con SPDI**. • El componente genera un inventario en un XML con el id de la persona, el

	número del solapín y el id del solapín.
	• El componente genera un reporte con la cantidad de trámites enviados a personalización, los nombres, apellidos y el identificador de las personas involucradas en dichos trámites.
	• El componente comprime:
	  • XML cifrado.
	  • Llave simétrica cifrada.
	  • Su llave pública.
	  • Inventario.
	  • Reporte.

Tabla 10: Descripción de la funcionalidad "Procesar información recibida del XABAL IDBIOACCESS".

Precondiciones.	El **Componente para comunicación con SPDI** debe haber recibido la información procedente del XABAL IDBIOACCESS.
Funcionalidades tratadas.	RF10.
Conceptos tratados.	Componente para comunicación con SPDI, Componente para enviar a personalización.

Descripción básica.	Una vez que el "Componente para comunicación con SPDI" recibe la información recibida del XABAL IDBIOACCES:
	• El componente descomprime el archivo, por lo que queda:
	  • XML cifrado.
	  • Llave simétrica cifrada.
	  • Su llave pública.
	  • Inventario.
	  • Reporte.
	• El componente descifra con su llave privada la llave simétrica recibida.
	• El componente descifra con la llave simétrica el archivo XML.
	• El componente comprueba la autenticidad del archivo XML con la firma digital.
	• El componente confecciona la orden de impresión.
	• El componente envía la orden de impresión al SPDI mediante el servicio web OrdenImport.

Tabla 11: Descripción de la funcionalidad "Preparar información para enviar al XABAL IDBIOACCESS".

Precondiciones.	Después que el SPDI personaliza las credenciales devuelve una información de respuesta al **Componente para comunicación con SPDI**.
Funcionalidades tratadas.	RF11.

Conceptos tratados.	correo electrónico, dispositivo de almacenamiento, FTP, XABAL IDBIOACCESS

Descripción básica.	• El componente recibe mediante el servicio web OrdenStatus la información de respuesta procedente del SPDI
	• El componente genera un archivo XML con la información de respuesta (el identificador de cada trámite y el número del solapín correspondiente) y lo firma digitalmente para garantizar el no repudio.
	• El componente cifra el archivo XML utilizando el algoritmo simétrico AES.
	• El componente cifra la llave generada con el algoritmo AES utilizando la llave pública procedente del **Componente para enviar a personalización**.
	• El componente comprime:
	  • XML cifrado.
	  • Llave simétrica cifrada.
	• El componente envía el archivo comprimido al XABAL IDBIOACCESS mediante alguna vía de transporte especificada (correo electrónico, FTP o dispositivo de almacenamiento).

Tabla 12: Descripción de la funcionalidad "Configuración general".

Precondiciones.	El usuario debe estar autenticado en el sistema y debe poseer el permiso de "Configuración general".
Funcionalidades tratadas.	RF1, RF1.1, RF1.2, RF2, RF2.1, RF2.2, RF3, RF3.1, RF3.2.
Conceptos tratados.	Correo electrónico, FTP.
Descripción Básica.	1. El sistema muestra un menú en el que se encuentran las siguientes opciones: • Configuración general. • Enviar a personalización. • Actualizar datos de personalización. 2. Si el usuario oprime la opción "Configuración general" el sistema muestra la interfaz "Configuración general", donde estarán publicados por pestañas: • Configuración de correo. • Configuración de FTP. • Datos a imprimir. • Generar Llaves. 2.1 En la pestaña "Configuración de correo" el usuario podrá introducir y modificar los siguientes datos (ver Figura 22): • Nombre del usuario de correo. • Dirección de correo del usuario. • Servidor de entrada.

- Servidor de salida.
- Usuario.
- Contraseña.
- Dirección de correo destino.

2.2 En la pestaña “Configuración de FTP” el usuario podrá introducir y modificar los siguientes datos (ver Figura 23):

- Usuario.
- Contraseña.
- URL.
- Dominio.

2.3 En la pestaña “Datos a imprimir” el usuario podrá seleccionar los datos que desea imprimir de las personas (ver Figura 24):

2.4 El sistema muestra el botón “Generar llave”, una vez que se oprime se crean las llaves pública y privada del componente, éstas se almacenan en la base de datos y se desactiva el botón.

2.5 El sistema guardará la información en archivos de configuración y las contraseñas de correo y de FTP son cifradas para su protección, luego el sistema lanzará el mensaje “Guardado”.

Prototipos

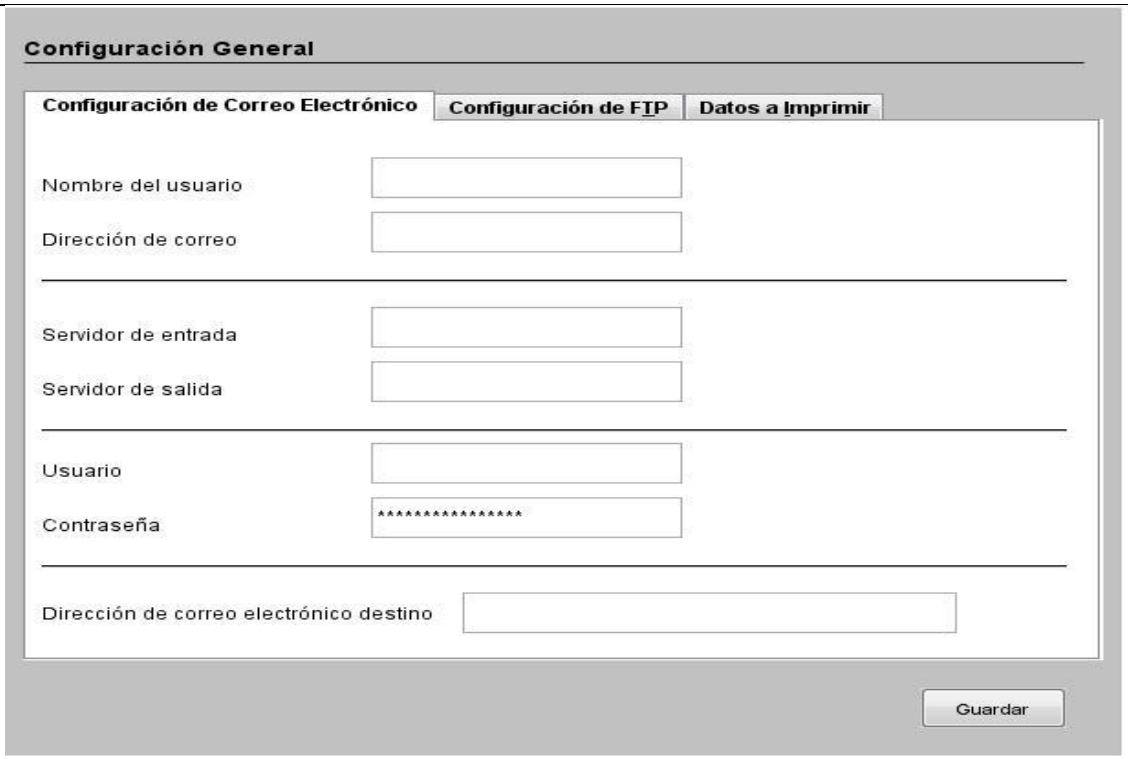


Figura 22: Interfaz “Configuración de Correo Electrónico”.

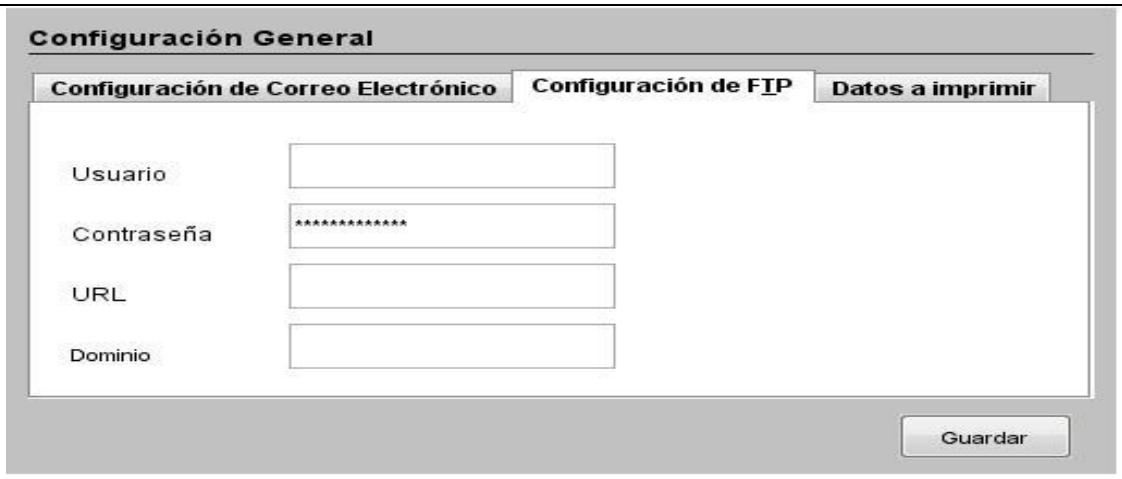


Figura 23: Interfaz "Configuración de FTP".

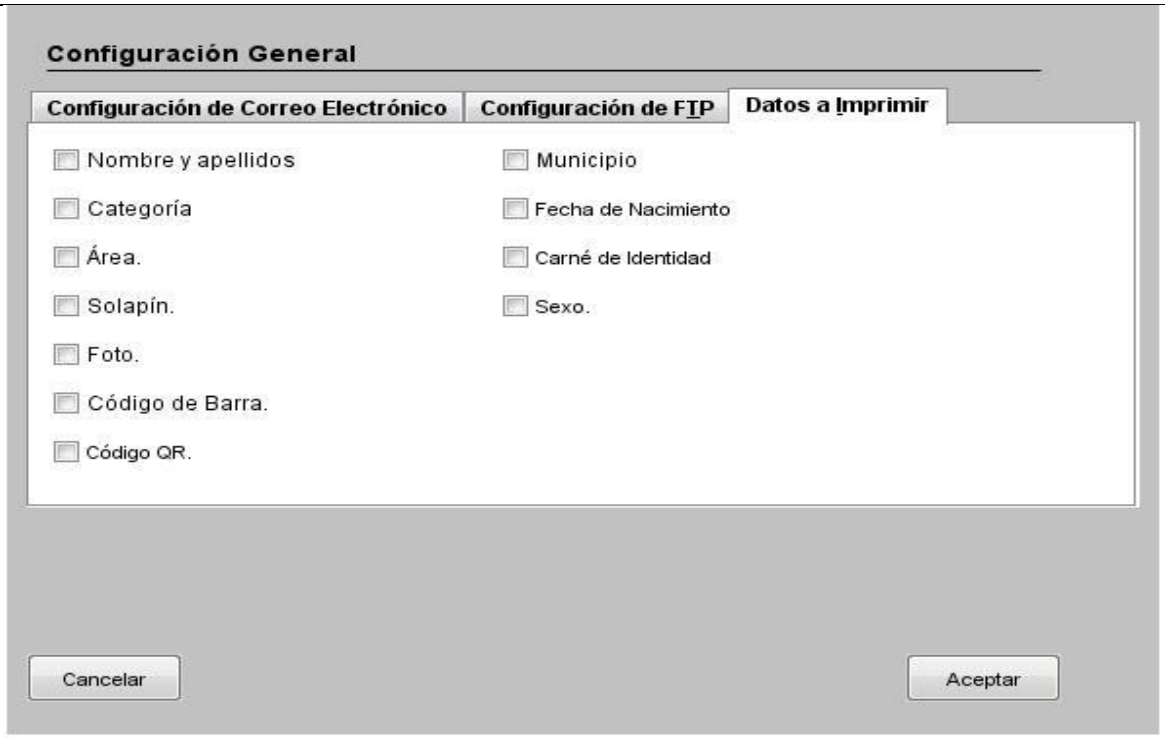


Figura 24: Interfaz "Datos a imprimir".

Tabla 13: Descripción de la funcionalidad "Procesar información recibida del SPDI.".

Precondiciones.	El usuario debe estar autenticado en el sistema y debe poseer el permiso de "Actualizar".
Funcionalidades tratadas.	RF12.
Conceptos tratados.	Base de datos.
	1. El sistema muestra un menú en el que se encuentran las siguientes opciones: • Configuración general.

	• Enviar Solicitudes. • Actualizar. 2. Si el usuario oprime la opción “Actualizar” el componente muestra en pantalla una ventana donde puede seleccionar la ubicación para cargar el archivo. 3. El usuario selecciona el archivo con la información y presiona el botón “Aceptar”. 4. El componente carga el archivo de la información. 5. El componente descomprime el archivo, por lo que queda: • XML cifrado. • Llave simétrica cifrada. 6. El componente descifra con su llave privada la llave simétrica recibida. 7. El componente descifra con la llave simétrica el archivo XML. 8. El componente comprueba la autenticidad del archivo XML con la firma digital. 9. El sistema actualiza la base de datos.

ANEXO 3: Diagrama de clases del Componente para comunicación con el SPDI.

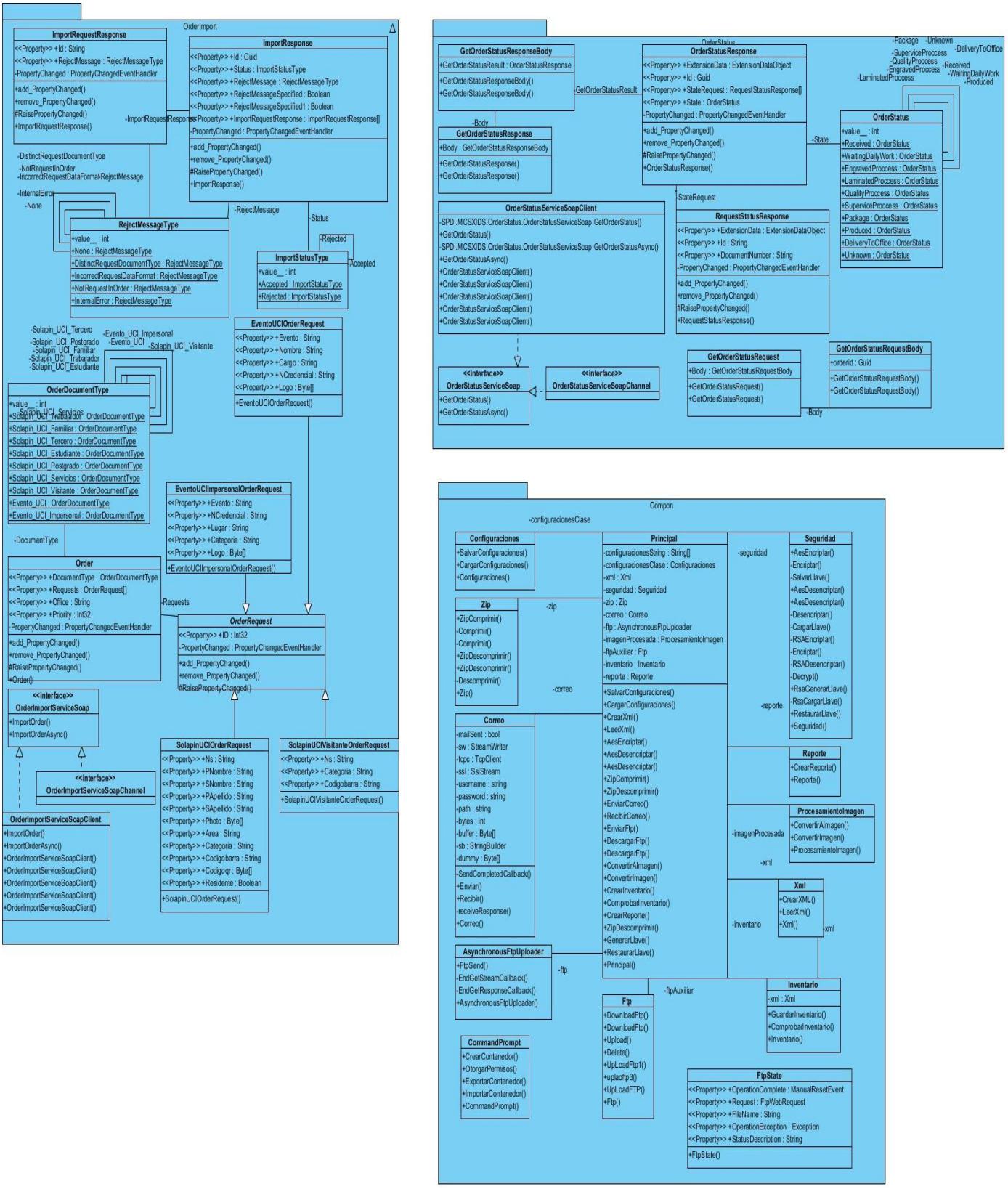


Figura 25: Diagrama de clases del Componente para comunicación con el SPDI.

ANEXO 4: Modelo de datos del sistema XABAL IDBIOACCESS con las nuevas tablas de la solución.

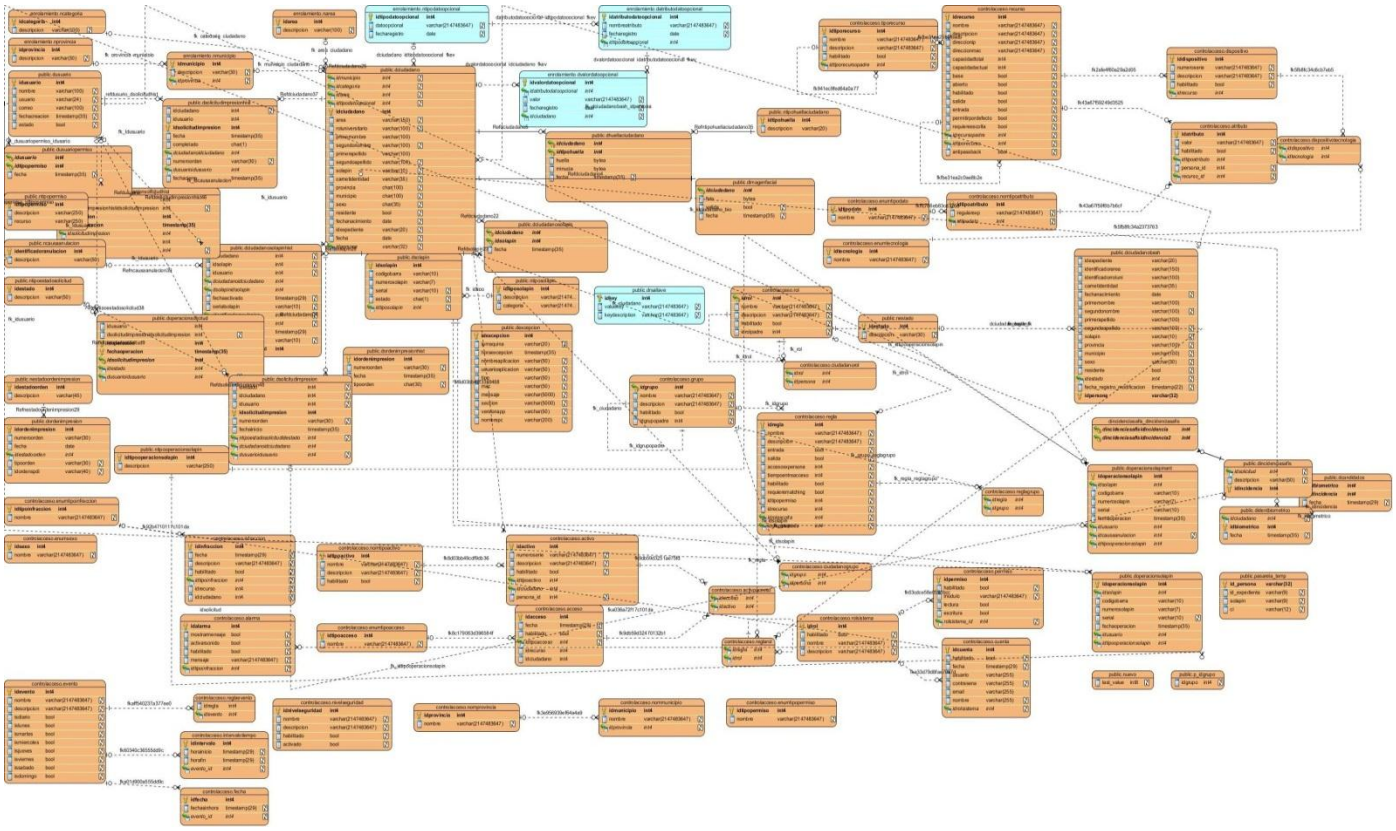


Figura 26: Modelo de datos del sistema XABAL IDBIOACCESS con las nuevas tablas de la solución.

ANEXO 5: Diagramas de secuencia.

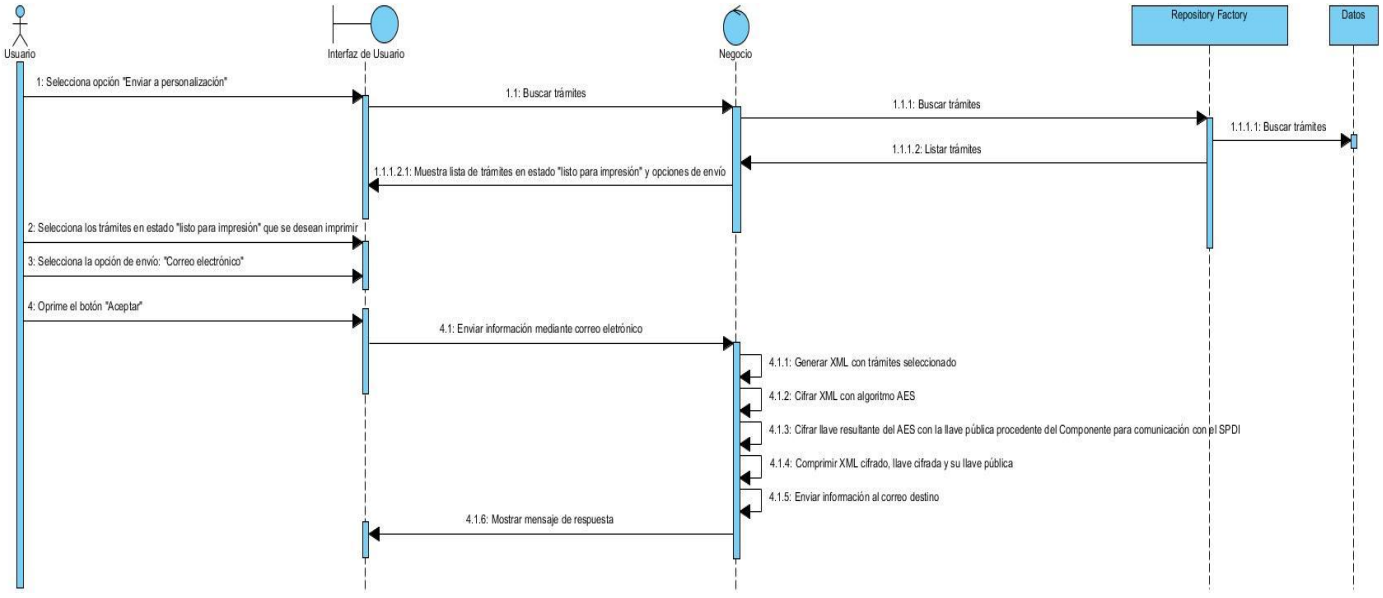


Figura 27: Diagrama de secuencia "Enviar a personalización". "Envío mediante Correo Electrónico".

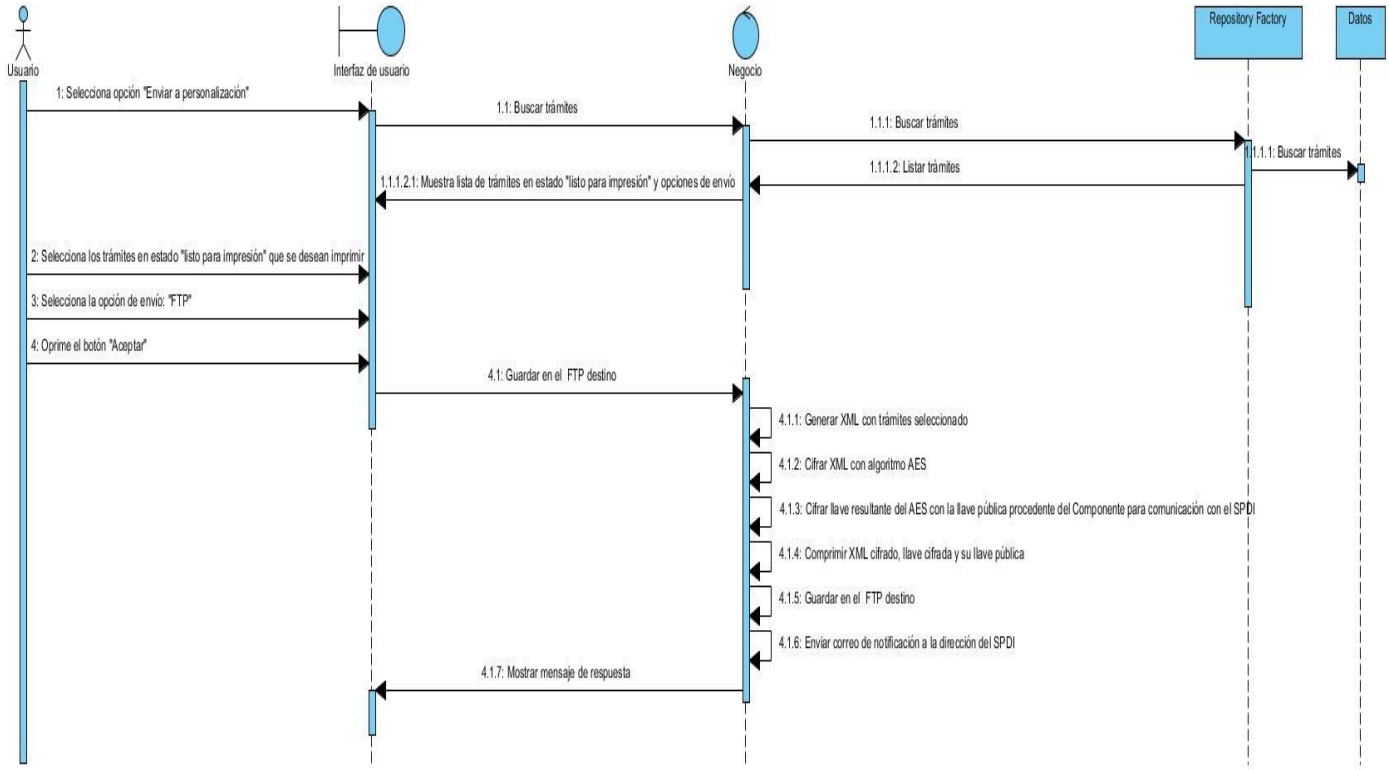


Figura 28: Diagrama de secuencia "Enviar a personalización". "Guardado en FTP".

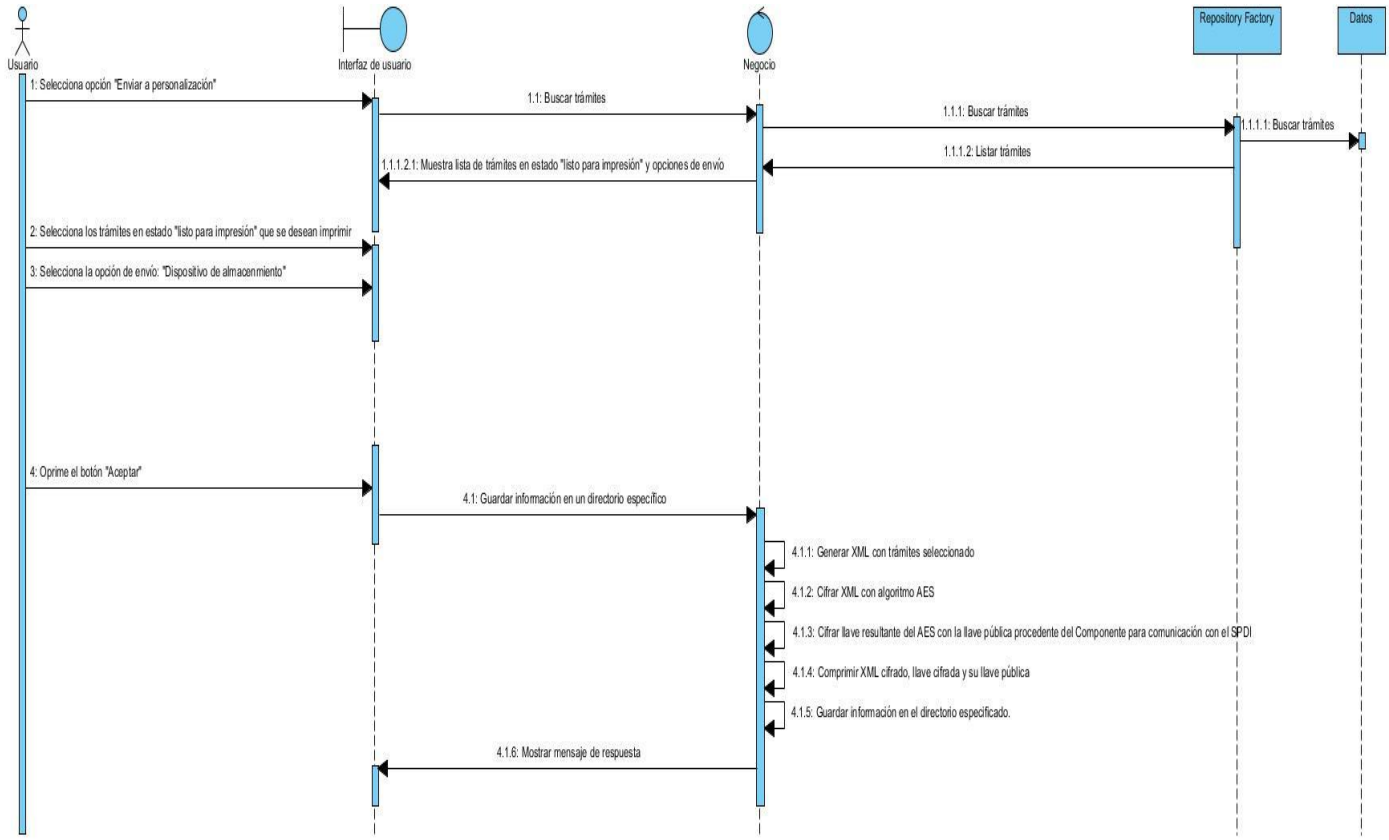


Figura 29: Diagrama de secuencia "Enviar a personalización". "Guardado en Dispositivo de Almacenamiento".

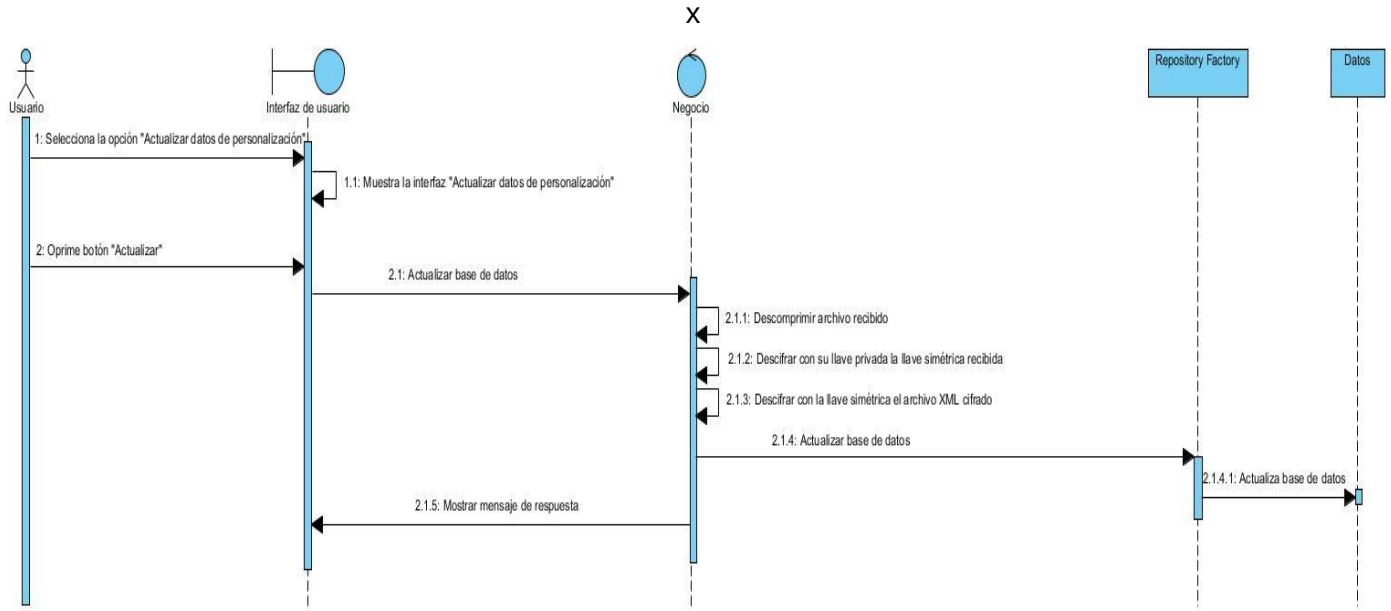


Figura 30: Diagrama de secuencia "Actualizar datos de personalización".

ANEXO 6: Diagrama de componentes del Componente para comunicación con el SPDI.

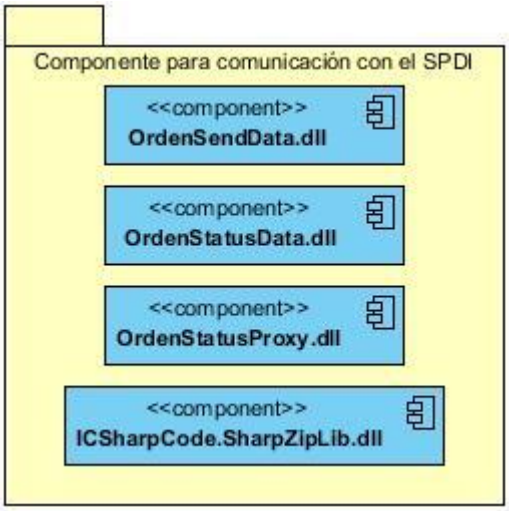


Figura 31: Diagrama de componentes del Componente para comunicación con el SPDI.

ANEXO 7: Casos de Prueba de Caja Negra.

Tabla 14: Descripción de las variables del caso de prueba Configuración de Correo Electrónico.

No.	Nombre del Campo	Clasificación	Valor Nulo	Descripción
1	Nombre del usuario	Campo de texto	Sí	Puede quedar vacío, pero el destinatario no sabrá quién envió el

				correo, por lo que se aconseja introducir un nombre. Puede aceptar letras.
2	Dirección de correo	Campo de texto	No	Debe quedar registrada la dirección electrónica del remitente para poder enviar correos. Acepta letras, número, caracteres especiales como (@, _, -, .).
3	Servidor de correo entrante	Campo de texto	No	Es necesario introducir este dato para poder recibir correos. El tipo de datos que acepta es un *string*, y debe cumplir con la expresión regular destinada para ello.
4	Servidor de correo saliente	Campo de texto	No	Es necesario introducir este dato para poder enviar correos. El tipo de datos que acepta es un *string*, y debe cumplir con la expresión regular destinada para ello.
5	Usuario	Campo de texto	No	Es necesario introducir el usuario para tener acceso al envío de correos. Puede aceptar letras y números y caracteres especiales como _.
6	Contraseña	Campo de contraseña	No	Es necesario introducir la contraseña para tener acceso al envío de correos. La contraseña debe contener: números, caracteres especiales, letras mayúsculas y minúsculas, además no debe ser menor a 7 caracteres.
7	Dirección de correo electrónico destino	Campo de texto	No	Debe quedar registrado al menos un correo electrónico destino para poder enviarlo. Acepta letras, número, caracteres especiales como (@, _, -, .).

Tabla 15: Caso de prueba Configuración de Correo Electrónico.

Escenario	Descripción	Nombre del usuario	Dirección de correo	Servidor de correo entrante	Servidor de correo saliente	Usuario	Contraseña	Dirección de correo destino	Respuesta del Sistema	Flujo Central
EC 1.1. Configurar el correo electrónico.	Para que el sistema pueda enviar correos electrónicos, será necesario que estén llenos estos campos para enviar un correo.	V	V	V	V	V	V	V	El sistema guardará la configuración y permitirá enviar correos electrónicos.	Opción Configuración General. Pestaña Configuración de correo electrónico. Guardar.
		I	V	V	V	V	V	V	El sistema guardará la configuración y permitirá enviar correos electrónicos.	
		V	I	V	V	V	V	V	El sistema guardará la configuración y no permitirá enviar correos electrónicos.	
		V	V	I	V	V	V	V	El sistema guardará la configuración y no permitirá enviar correos electrónicos.	
		V	V	V	I	V	V	V	El sistema guardará la	

									configuración y no permitirá enviar correos electrónicos.	
		V	V	V	V	I	V	V	El sistema guardará la configuración y no permitirá enviar correos electrónicos.	
		V	V	V	V	V	I	V	El sistema guardará la configuración y no permitirá enviar correos electrónicos.	
		V	V	V	V	V	V	I	El sistema guardará la configuración y no permitirá enviar correos electrónicos.	
		I	I	I	I	I	I	I	El sistema guardará la configuración y no permitirá enviar correos electrónicos.	

Tabla 16: Descripción de las variables del caso de prueba Configuración de FTP.

No.	Nombre del Campo	Clasificación	Valor Nulo	Descripción
1	Usuario	Campo de texto	No	Es necesario un nombre de usuario para poder acceder al FTP. Puede aceptar letras y números y caracteres especiales como _.

2	Contraseña	Campo de contraseña	No	Es necesaria una contraseña para poder acceder al FTP. La contraseña debe contener: números, caracteres especiales, letras mayúsculas y minúsculas, además no debe ser menor a 7 caracteres.
3	URL	Campo de texto	No	El puerto para que se pueda establecer la comunicación. El tipo de datos que acepta es un *string*, y debe cumplir con la expresión regular destinada para ello.
4	Dominio	Campo de texto	No	Se debe introducir el dominio al que pertenece el FTP. El tipo de datos que acepta es un *string*, y debe cumplir con la expresión regular destinada para ello.

Tabla 17: Caso de prueba Configuración de FTP.

Escenario	Descripción	Usuario	Contraseña	URL	Dominio	Respuesta del Sistema	Flujo Central
EC 1.1. Configurar el FTP.	Para que el sistema pueda guardar la información en un FTP, será necesario que estén llenos de manera correcta cada uno de estos campos, de lo contrario, lanzará un mensaje de error especificando el	**V**	**V**	**V**	**V**	El sistema guardará la configuración y permitirá guardar el fichero con la información a imprimir en el FTP.	Opción Configuración general. Pestaña Configuración de FTP. Guardar.
		I	**V**	**V**	**V**	El sistema guardará la configuración y no permitirá guardar la información en el FTP.	
		V	**I**	**V**	**V**	El sistema guardará la configuración y no permitirá guardar la	

	campo incorrecto.					información en el FTP.	
		V	**V**	**I**	**V**	El sistema guardará la configuración y no permitirá guardar la información en el FTP.	
		V	**V**	**V**	**I**	El sistema guardará la configuración y no permitirá guardar la información en el FTP.	
		I	**I**	**I**	**I**	El sistema guardará la configuración y no permitirá guardar la información en el FTP.	

ANEXO 8: Casos de Prueba de Integración.

Tabla 18: Caso de Prueba de Integración "Procesar información recibida del XABAL IDBIOACCESS".

Caso de prueba: Procesar información recibida del XABAL IDBIOACCESS.
Sistema al que se integra: SPDI.
Condiciones de ejecución: Que se haya recibido la información del XABAL IDBIOACCESS.
Descripción de la prueba: Comprobar que el módulo es capaz de descomprimir, descifrar y enviar a personalizar la información recibida del XABAL IDBIOACCESS.
Entradas/Pasos de ejecución: La información se descarga del FTP, del correo o del dispositivo de almacenamiento y se coloca en un directorio por defecto. El sistema la descomprime, la descifra, comprueba su autenticidad mediante la firma digital y la envía a personalización.
Resultado esperado: El sistema es capaz de tomar la información del directorio por defecto y enviarla a personalización.
Evaluación: Prueba satisfactoria.

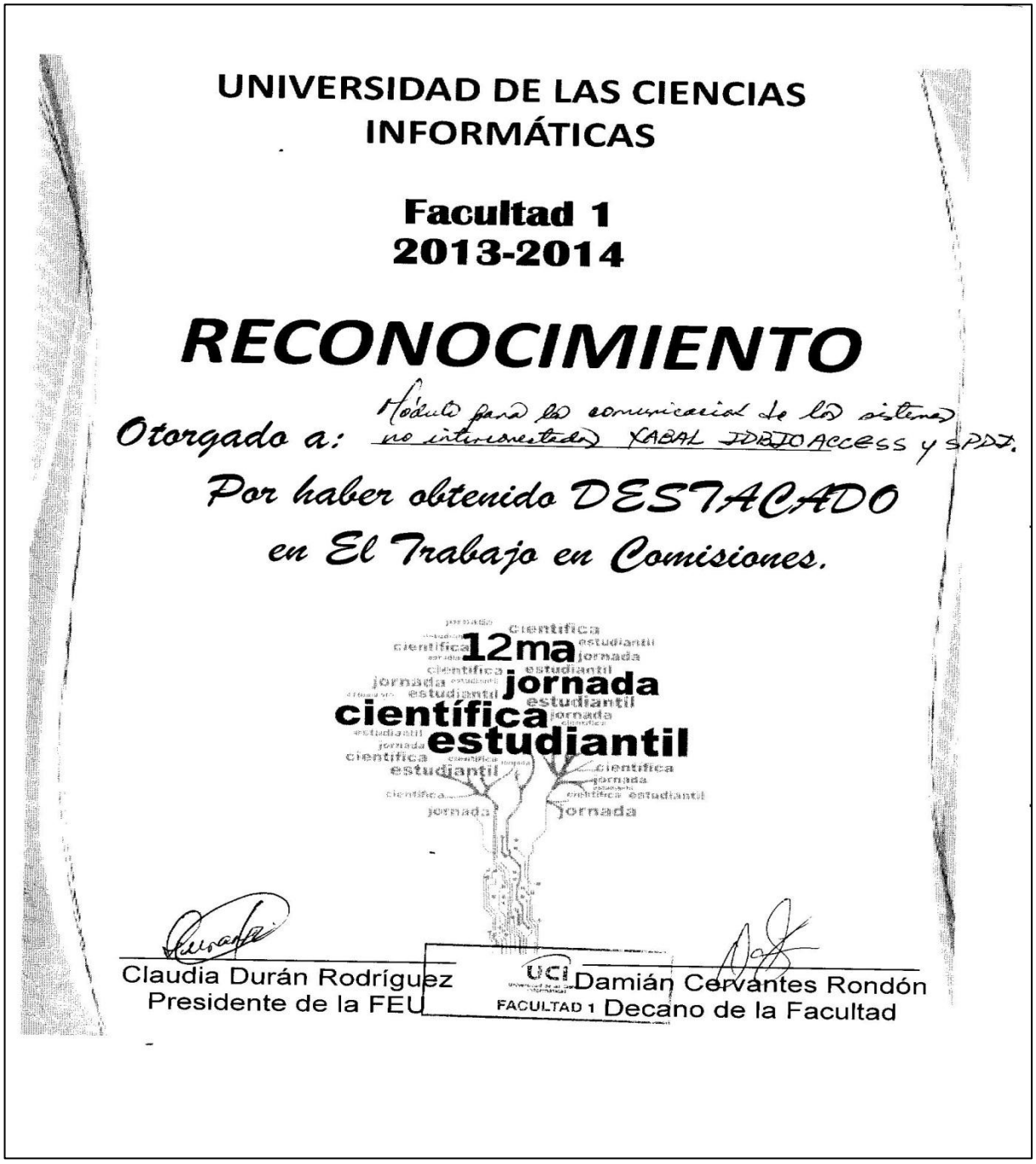
Tabla 19: Caso de Prueba de Integración "Preparar información para enviar al XABAL IDBIOACCESS".

Caso de prueba: Preparar información para enviar al XABAL IDBIOACCESS.
Sistema al que se integra: SPDI.
Condiciones de ejecución: Se deben haber personalizado las credenciales y el SPDI debe haber generado información actualizada para el XABAL IDBIOACCESS.
Descripción de la prueba: Comprobar que el módulo es capaz guardar la información generada en un archivo XML, que lo proteja con algoritmos criptográficos, lo firme digitalmente y lo comprima, que sea capaz de enviarlo por correo electrónico, o de guardarlo en un FTP o en un directorio específico.
Entradas/Pasos de ejecución: La información se genera, se firma digitalmente, se cifra, se comprime y se envía: por correo electrónico o se guarda en un FTP o directorio específico.
Resultado esperado: El sistema es capaz de enviar la información actualizada por correo electrónico, de guardarla en un FTP o en un directorio específico.
Evaluación: Prueba satisfactoria.

Tabla 20: Caso de Prueba de Integración "Procesar información recibida del SPDI".

Caso de prueba: Procesar información recibida del SPDI.
Sistema al que se integra: XABAL IDBIOACCESS.
Condiciones de ejecución: Que se haya recibido la información del SPDI.
Descripción de la prueba: Comprobar que el módulo es capaz descomprimir, descifrar, comprobar la firma digital y de actualizar la base de datos del sistema con la información recibida del SPDI.
Entradas/Pasos de ejecución: El sistema es capaz de tomar la información del directorio por defecto, lo descomprime, lo descifra, comprueba su autenticidad mediante la firma digital y actualiza la base de datos.
Resultado esperado: El sistema es capaz de actualizar la base de datos.
Evaluación: Prueba satisfactoria.

ANEXO 9: Reconocimiento a la aplicación.



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 1
2013-2014

RECONOCIMIENTO

Otorgado a: Módulo para la comunicación de los sistemas no interconectados XABAL JDBJO ACCESS y SPDT.

Por haber obtenido DESTACADO en El Trabajo en Comisiones.

12ma jornada científica estudiantil

Claudia Durán Rodríguez
Presidente de la FEU

Damián Cervantes Rondón
Decano de la Facultad

Figura 32: Reconocimiento a la aplicación.