



VICERECTORÍA DE TECNOLOGÍA
DIRECCIÓN DE INFORMATIZACIÓN

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 6

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

Título: *Módulo de gestión de solicitudes de
credenciales de identificación para el sistema
IDBIOACCESS.*



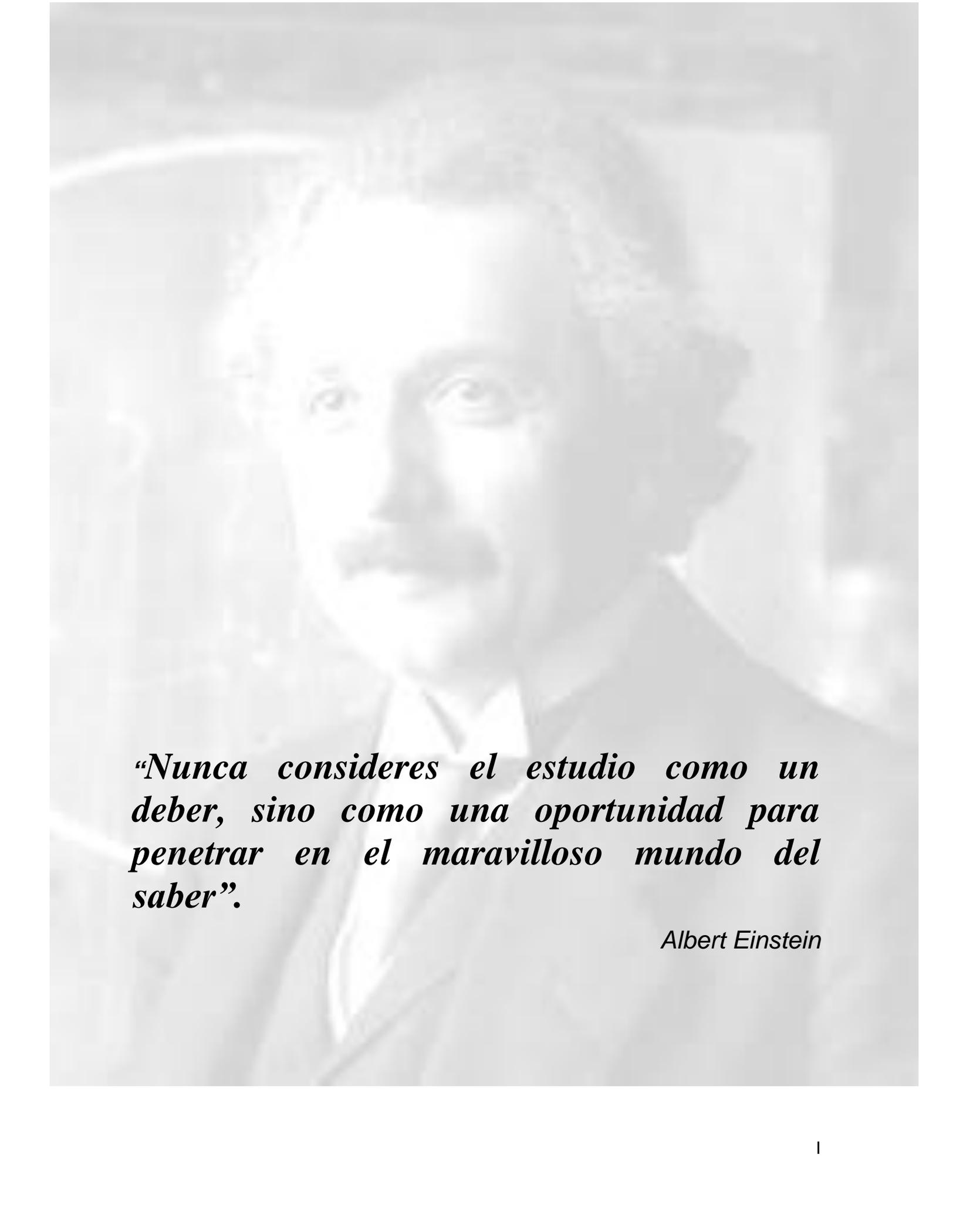
Autor: Estrella Ramos Pérez

Tutores: Ing. Nayla Socarras Monzón.

Ing. Addiel Alejandro Téllez Escalona.

Marzo del 2015

“Año 57 del Triunfo de la Revolución”



“Nunca consideres el estudio como un deber, sino como una oportunidad para penetrar en el maravilloso mundo del saber”.

Albert Einstein

DECLARACIÓN DE AUTORÍA.

Declaramos ser autores de la presente tesis que tiene por título: Módulo de Gestión de Solicitudes de credenciales de identificación para el Sistema IDBIOACCESS y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autor

Estrella Ramos Pérez

Tutores

Nayla Socaras Monzón

Addiel Alejandro Téllez Escalona

DATOS DE CONTACTO

Ing. Nayla Socarras Monzón: Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Especialista superior con tres años de experiencia. Correo electrónico: nsocarras@uci.cu.

Ing Addiel Alejandro Téllez Escalona: Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Especialista A con tres años de experiencia. Correo electrónico: addiel@uci.cu.

AGRADECIMIENTOS.

A Severino Hernández Pita por darme el impulso que necesité para comenzar y por todo su apoyo incondicional en cada dificultad presentada confiando en mí haciéndome entender que sí podía llegar hasta el final, logrando así que mi objetivo se tornara un compromiso moral con él y con el colectivo.

A Vismael, Florangel, Viera y todo el colectivo de Seguridad y Protección que en muchísimas ocasiones me escudaron para poder cumplir con mis deberes y funciones sin necesidad de ausentarme a clases.

A mi esposo, mi padrastro Gonzalo y Vantourt que me dieron buena instrucción política-cultural, consultas y aclaraciones de dudas, por lo que gracias a ellos obtuve buenas notas en algunas asignaturas.

A todos mis amigos de la universidad, en especial al grupo que me acogió con cariño a pesar de incorporarme dos años después. Todos han sido muy pacientes conmigo en mis malos momentos transmitiéndome alegrías de juventud. De cierta forma, gracias a todo ese apoyo incondicional, dándome ánimo y fuerzas para no rendirme, es que logré llegar hasta el final. Han sido para mí un ejemplo de firmeza y perseverancia. Fue un placer compartir todos estos años con ustedes. Que esta amistad perdure por siempre.

A todos los profesores que nos impartieron las asignaturas durante todos estos años, por todo ese esfuerzo personal, paciencia y tiempo dedicado con confianza a guiarnos y lograr de nosotros mejores profesionales.

A mis tutores Addiel y Nayla que se esforzaron muchísimo para guiarme y lograr de mí una buena tesista.

A Nilberto, Abelito, Cesar y Alexander, Mairelys, sin ellos, realmente, no lo hubiera conseguido.

Muy en especial a mi gran amigo Asnay, por todo su apoyo como instructor y amigo fiel, por toda su gran paciencia cuando lo molestaba con mis grandes dudas, por su gran amabilidad, por su dedicación con todo el colectivo, aun cuando no tenía ningún compromiso con nosotros. Gracias por estar siempre.

A mis amigos Yoisi y Raulito por dedicarme grandes espacios en sus tiempos libres para ayudarme solo a cambio de un mi éxito.

A todas las personas que de una forma u otra contribuyeron a mi superación profesional y a demostrar que lo más valioso para lograr un objetivo es la voluntad. A todos MUCHAS GRACIAS.

RESUMEN

Los sistemas de emisión de documentos de identificación son de gran importancia para todos los países del mundo. De manera cotidiana a la Universidad de las Ciencias Informáticas (UCI) acceden un gran número de personas por lo que se hace necesario saber si el personal que transita por las diferentes instalaciones está debidamente autorizado, tal y como establece el reglamento de Identificación (PG-006-02, 2013). Para esto se hace necesario llevar a cabo un sistema automatizado para la gestión de solicitudes de credenciales de la Universidad que sea capaz de brindar una información absoluta y auténtica sobre la solicitud por parte del usuario. El análisis realizado al sistema de identificación implementado dio a conocer la necesidad de perfeccionar la organización del trabajo, mejorar la gestión de solicitud en el proceso de identificación y elevar el nivel de seguridad en la ejecución de los procesos de forma tal que se garantice la autenticidad del funcionario. Para el buen desarrollo de la solución informática se decide utilizar la metodología *Feature Driven Development* (FDD). Visual Paradigm como herramienta de modelado y PostgreSQL como sistema gestor de base de datos (SGBD), Plataforma de desarrollo Microsoft .Net *framework*. Como lenguaje de programación C# usando Visual Estudio 2010. El presente trabajo de diploma expone una solución informática de manera que se gestione el proceso relacionado con la solicitud, recepción, coordinación, actualización, aprobación y emisión de credenciales. Además se logra informatizar la información referente al cobro de los solapines que se maneja en la oficina de identificación.

PALABRAS CLAVE: acreditación, credenciales, identificación, sistema, solicitud.

ABSTRACT

The identification documents emitting systems are of a great importance to all Emission systems of identification documents are of great importance to all countries around the globe. There are a significant number of people accessing to the University of Informatics Sciences (UCI) permanently so it becomes a necessity to determine if the staff passing through the different facilities is duly authorized as provided in the bylaws of ID (PG- 006-02, 2013). Consequently it is also a need to develop an automated system for managing requests for credentials from the University to be able to provide complete and authentic information over the application from users. The performed analysis upon the implemented identification system point out the need of: improvements on the related managerial processes on the organization, and to focus security levels in the execution processes so authenticity results official guaranty. Feature Driven Development (FDD) methodology was chosen in the aim of proper development of the software solution. So as Visual Paradigm, PostgreSQL, and Microsoft .Net framework system were the selected tools for modeling, data base management, and development platform respectively. Finally, the implementation language it result to be C# over Visual Studio 2010. Present dissertation presents a software solution so that the processes related with receiving, coordination, updating, approval and issuance of credentials for staff identification are managed. In addition is achieved computerize the information regarding the recovery of Special pin is handled in the office of identification.

KEYWORDS: accreditation, credentials cards, identification, system, application.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	1
Capítulo 1: Fundamentación Teórica	1
1.1. Introducción.....	1
1.2. Conceptos asociados al proceso de solicitud de credenciales	1
1.2.1. <i>Sistema IDBIOACCESS</i>	2
1.3. Sistemas analizados en la investigación	3
1.4. Tecnologías Software a utilizar	5
1.4.1. <i>Frameworks.....</i>	5
1.4.2. <i>Lenguajes de programación</i>	7
1.4.3. <i>Entorno de desarrollo</i>	8
1.4.4. <i>Gestor de Base de datos</i>	9
1.5. Metodologías de software.....	12
1.5.1. <i>Feature Driven Development (FDD). Desarrollo Basado en Funcionalidades</i>	13
1.6. Conclusiones parciales.....	19
Capítulo 2: Descripción de la propuesta de solución	20
2.1. Introducción	20
2.2. Propuesta de solución	20
2.2.1. <i>Descripción del Sistema</i>	21
2.3. Modelado del proceso de negocio	22
2.3.1. <i>Conceptos asociados al modelo del Negocio.....</i>	22
2.3.2. <i>Diagrama del Modelo de negocio</i>	23
2.3.3. <i>Diagrama de las actividades del sistema.....</i>	24
2.3.4. <i>Reglas del negocio</i>	28
2.4. Requisitos del Software	28
2.4.1. <i>Técnicas de obtención de requisitos.....</i>	29
2.4.2. <i>Requisitos Funcionales.....</i>	30
2.4.3. <i>Requisitos no funcionales.....</i>	31
2.5. Planeación por funcionalidad.....	34
2.6. Diseño por funcionalidad.....	35
2.6.1. <i>Arquitectura propuesta.....</i>	35
2.6.2. <i>Patrones de Diseño.....</i>	37
2.6.3. <i>Diagrama de Paquetes.....</i>	41

2.6.4. Diagrama de Clases	42
2.6.5. Modelo de datos.....	43
2.7. Conclusiones parciales	44
Capítulo 3: Implementación y Prueba	45
3.1. Introducción	45
3.2. Diagrama de componentes	45
3.3. Diagrama de despliegue	46
3.4. Estándares de codificación	47
3.5. Validación de la solución	49
3.5.1. Pruebas de Caja negra.	49
3.5.2. Entorno de las pruebas.	53
3.5.3. Validación de las variables de la investigación.....	54
3.5.4. Resultados.	55
3.6. Conclusiones parciales	55
Conclusiones	56
Recomendaciones	57
Referencias bibliográficas	58
Bibliografía consultada.	59
ANEXOS.	61
Anexo 1. Entrevistas.	61
Anexo 2: Requisitos Funcionales	62
Anexo 3. Descripción de los requisitos.....	63
Anexo 5. Descripción de las tablas de la base de datos.....	68

ÍNDICE DE FIGURAS

Figura 1: Diagrama del Modelo de negocio..... 24

Figura 2: Diagrama de actividades..... 25

Figura 3: Arquitectura cliente-servidor. 36

Figura 4: Vista Lógica de la arquitectura 37

Figura 5: Ejemplo del uso del patrón Creador..... 39

Figura 6: Ejemplo del uso del patrón Singleton..... 40

Figura 7: Diagrama de Paquetes del sistema. 42

Figura 8: Diagrama de clases del diseño. 43

Figura 9: Diagrama de componentes 46

Figura 10: Diagrama de Despliegue..... 47

Figura 11: Ejemplo de código con paréntesis..... 48

Figura 12: Resultado de las Pruebas de Caja Negra. 53

ÍNDICE DE TABLAS

Tabla 1: Actores que interactúan con el sistema..... 21

Tabla 2: Descripción del flujo de actividades 26

Tabla 3: Descripción de requisitos 32

Tabla 4: Descripción de la entidad idciudadano 44

Tabla 5: Criterios de calidad 49

Tabla 6: Caso prueba Solicitudes 51

Tabla 7: Resumen de no conformidades..... 52

Tabla 8: Comparación de los funcionamientos del sistema..... 54

INTRODUCCIÓN

La informatización es un proceso innegable que humaniza el trabajo y en la actualidad es cada vez más usado por el sector empresarial en el mundo. Cuba no está al margen de tal desarrollo por lo que en sus instituciones y empresas se hace un uso creciente de las Tecnologías de la Información y las Comunicaciones (TIC).

La Gestión de Información (GI) constituye un proceso mediante el cual se planifican, organizan, dirigen y controlan los recursos de información de una organización asegurando un adecuado tratamiento, intercambio y uso de este recurso, para contribuir al establecimiento de fortalezas organizacionales. Al desarrollar la GI la organización logra identificar y adquirir la información necesaria para satisfacer sus necesidades informativas, y organizarla para un acceso cómodo (Domínguez, 2008).

Los sistemas actuales de GI se basan en gran medida en la tecnología para recopilar y presentar datos. Un Sistema de Gestión de Información (SGI) puede facilitar la colaboración y la comunicación, pero el propósito principal de un SGI es lograr que la toma de decisiones por parte de los directivos sea más eficiente y productiva mediante la combinación de la información de una variedad de fuentes en una sola base de datos y la presentación de la información en un formato lógico.

La Universidad de las Ciencias Informáticas (UCI) tiene entre sus objetivos primarios la creación de una serie de proyectos informáticos para automatizar los procesos vitales que rigen la eficiencia de estos tipos de centros. La informatización de todos estos procesos consiste básicamente en la creación de sistemas de gestión de datos.

Estos sistemas informáticos posibilitan el acceso a toda la información de forma confiable, precisa y oportuna, la aceleración del flujo de información entre las distintas áreas del centro, la eliminación de datos y operaciones innecesarias o redundantes. Además de reducir los tiempos y los costos de realización de los procesos, disminuir gastos en insumos de oficina como: papel y cartuchos de tinta o toner para impresión. Otra de las ventajas de estos sistemas radica en la sencillez de manejo, mejorando las condiciones de trabajo del personal (Sánchez Reyes, 2007).

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

La identificación y diseño de credenciales ocupan un rol importante si de seguridad se trata en las organizaciones, las tarjetas para el reconocimiento del usuario permiten personalizar las mismas con los datos y la foto del usuario para carné y otros documentos de acreditación garantizando de esta forma una identificación única, difícil de duplicar o falsificar.

En la UCI se hace necesario agilizar la gestión de solicitudes en la oficina de Identificación y mejorar el control sobre la información que se lleva a cabo por concepto de pérdida o deterioro de la credencial ya que este proceso actualmente no está informatizado. La Universidad cuenta con un sistema de identificación llamado IDBIOACCESS, pero no cumple con todas las expectativas necesarias puesto que no le permite a los directivos de las diferentes áreas agilizar la tramitación de solicitud de credenciales para su emisión a los impersonales (en el caso de los visitantes) o cancelación y emisión por motivos de: pérdida, deterioro o baja con pérdida o baja con deterioro. Esto trae como consecuencia que el proceso sea lento y engorroso.

Toda la información relacionada a la solicitud de credenciales se archiva en copia dura, pero no queda constancia digital de la misma. La información referente a los vales de pago, documento oficial para cobro de la credencial, por pérdida o deterioro del mismo se almacena manualmente en un documento Excel, dificultando a los especialistas de la oficina de identificación obtener de forma efectiva la información consolidada por este concepto. Estos vales actualmente se recogen en la dirección de economía que a su vez paga para el servicio de impresión a otra entidad, incurriendo la UCI en gastos por este concepto. El especialista de la oficina de identificación llena los datos que requiere el vale de pago a mano y luego debe volver a dirigirse a la dirección de economía para hacer entrega del dinero recaudado. Por tanto se hace necesario diseñar y emitir el comprobante por concepto de pérdida de la credencial.

El análisis realizado al proceso de identificación de la UCI demostró la necesidad de perfeccionar la organización del trabajo, lograr una mayor eficiencia y mejorar el proceso de solicitud de credenciales de identificación, disminuir los gastos de materiales de oficina por concepto de solicitud, así como elevar el nivel de seguridad en la ejecución de los procesos de forma tal que se garantice la autenticidad del funcionario que autoriza.

Por lo anteriormente planteado se define como **Problema a resolver:**

¿Cómo contribuir a la mejora de la gestión del proceso de solicitudes de credenciales de identificación por el sistema IDBIOACCESS de manera que se agilice el proceso de

trámites de solicitud de credenciales y aumente el control de información por concepto de cobro de credencial en la oficina de Identificación de la UCI?

De ahí que el **Objeto de estudio** lo constituya:

Procesos de identificación.

Enmarcado en el **Campo de acción**:

Proceso de gestión de solicitudes de credenciales de identificación en la UCI.

La investigación está encaminada a cumplir con el **Objetivo general**:

Desarrollar un módulo de gestión de solicitudes de credenciales para el sistema IDBIOACCESS que contribuya a agilizar el proceso de trámites de solicitud de credenciales y aumentar el control de información por concepto de cobro de credencial.

Desglosados en los siguientes **Objetivos específicos**:

- Caracterizar los procesos de gestión de solicitud de credenciales para elaborar el marco teórico de la investigación científica.
- Modelar el proceso de gestión de solicitud de servicio de emisión de credenciales.
- Conceptualizar una propuesta de desarrollo que permita diseñar y emitir el comprobante por concepto de pérdida de la credencial.
- Implementar la propuesta de desarrollo a partir de los requisitos identificados utilizando las tecnologías y técnicas definidas.
- Realizar la validación de las funcionalidades que fueron implementadas en el sistema.

Para darle cumplimiento a los objetivos planteados se proponen las siguientes **Tareas de investigación**:

- Descripción de los principales conceptos asociados a los procesos de gestión de solicitud de credenciales para obtener una base teórica necesaria para el desarrollo de la solución.
- Caracterización de las soluciones informáticas de gestión de solicitudes de credenciales de identificación existentes.
- Descripción del diseño que permita emitir el comprobante por concepto de pérdida de la credencial.

- Especificación de las herramientas informáticas y metodologías a usar para el desarrollo del Módulo de gestión de solicitudes.
- Modelación y descripción de los procesos de negocio del Módulo de gestión de solicitud de credenciales de identificación.
- Implementación de la propuesta de desarrollo a partir de los requisitos identificados utilizando las tecnologías y técnicas definidas.
- Realización de las pruebas de caja negra al Módulo de Gestión de Solicitudes en función comprobar que el sistema funciona correctamente.
- Validación de las funcionalidades que fueron implementadas en el sistema.

La investigación se basa en la **Hipótesis** siguiente:

Si se desarrolla un módulo al sistema IDBIOACCESS que contribuya a mejorar la gestión de solicitudes de credenciales de identificación se agilizará el proceso de trámites de solicitud de credenciales y aumentará el control de información por concepto de cobro de solapín de la oficina de Identificación en la UCI.

Del cual se derivan como **Variables**:

Variable independiente: Módulo para el sistema IDBIOACCESS.

Variables dependientes:

- Proceso de trámites de solicitud de credenciales.
- Control de información por concepto de cobro de solapín.

Para darle cumplimiento a los objetivos trazados se emplean los siguientes Métodos:

Métodos teóricos

- Histórico – lógico.

Este método se ha utilizado al realizar un análisis de la trayectoria de los procesos de identificación de la UCI definiéndose las tendencias actuales y los cambios que se han generado a través de las experiencias adquiridas en las diferentes etapas de desarrollo social en la Universidad.

- Analítico-Sintético

El método de análisis y síntesis ha sido utilizado para la caracterización del proceso de solicitud de credenciales de identificación en la UCI e identificar sus conceptos

principales. También para analizar toda la bibliografía disponible para hacer un estudio completo de las funcionalidades que genera el sistema IDBIOACCESS y las necesidades de los cambios a realizar.

Métodos Empíricos

- La entrevista individual

Para cumplir el objetivo de agilizar el proceso de trámites de solicitud de credenciales de la oficina de Identificación y aumentar el control de información por concepto de cobro del solapín, se hace imprescindible realizar entrevistas con los especialistas que atienden el proceso de desarrollo de Gestión Universitaria y a los especialistas de la oficina de Identificación que están vinculados al desarrollo del proyecto ya que son los que están en contacto directo con el software al que se va a integrar este módulo.

El trabajo de diploma está estructurado en 3 capítulos.

Capítulo 1: Fundamentación teórica, donde se describen las herramientas y metodologías utilizadas durante la investigación. Se describen los conceptos fundamentales asociados al dominio del problema, además se definen los principales conceptos y definiciones que se deben conocer de las credenciales de identificación. Se describen las herramientas a utilizar para el diseño y se analizan las soluciones existentes tanto en el ámbito internacional como en el nacional.

Capítulo 2: Descripción de la propuesta de solución, contiene los elementos relacionados a la especificación de los requisitos a tener en cuenta durante el desarrollo de la propuesta de solución, se realiza el análisis de la aplicación, la descripción de la arquitectura y el diseño de la estrategia a elaborar.

Capítulo 3: Implementación y prueba, se presentan la implementación del sistema elaborado, las políticas de salva y recuperación ante fallos y la validación funcional del mismo.

Capítulo 1: Fundamentación Teórica

1.1. Introducción

En este capítulo se hace una reseña de las tecnologías existentes más relevantes para la identificación de personas, además de las herramientas, metodologías y softwares relacionados con el proceso y gestión de solicitudes a nivel internacional. También se exponen los conceptos fundamentales y aspectos teóricos que se tienen en cuenta en el desarrollo de la investigación.

Para poder entender la gestión de solicitudes de credenciales se debe tener conocimiento de los siguientes conceptos que están estrechamente relacionados con el dominio del problema:

1.2. Conceptos asociados al proceso de solicitud de credenciales

- Identificación

El término identificación se usa para designar al acto de identificar, reconocer o establecer los datos e información principal sobre una persona. La identificación, también puede ser el nombre que llevan determinadas documentaciones que tienen por objetivo justamente establecer la identidad de una persona o individuo.

- Gestión de identificación

Acciones o trámites que se realizan con el objetivo de obtener un documento identificativo.

- Procesos de Identificación

Son todas las acciones que la oficina de identificación realiza a una persona contratada por una entidad para obtener un documento que la identifique legalmente por la función o rol que ejerce dentro de la entidad.

- Sistema de identificación

La emisión de un documento de identificación es un proceso complejo, que requiere un alto nivel de seguridad y control, basada en normas legislativas establecidas para el lugar donde el documento tiene efecto legal (Llanes Embade, 2011).

- Sistema de gestión

Un sistema de gestión es un conjunto de etapas unidas en un proceso continuo, que permite trabajar ordenadamente una idea hasta lograr mejoras y su continuidad. Gestión informática es llevar a cabo la acción de adicionar, modificar y eliminar entidades en sistemas informáticos (López Rivera, 2013).

- Sistema de solicitudes

La solicitud es un documento por medio del cual se pide el goce de un derecho, beneficio o servicio que se considera que puede ser atendido. El sistema de solicitudes facilita la organización y seguimiento de los servicios solicitados por los usuarios.

- Gestión de solicitud de credencial

Es la acción que se realiza para obtener de forma legal el servicio de emisión de credenciales.

- Credencial

La credencial es un documento por medio del cual se acredita a una persona para que ejerza cargo o función, conforme a normas legales establecidas, o a un trabajador para que ejerza sus funciones diarias en forma permanente.

- Sistema de emisión de credenciales de identificación

Es una aplicación de software de escritorio para la creación y emisión de las credenciales como documento identificativo legal para todo el personal que radica o trabaja dentro de la UCI.

1.2.1. Sistema IDBIOACCESS

El Sistema de identificación IDBIOACCESS es un sistema para la captura de datos, captura de imagen y supervisión de imágenes y datos (Danny, 2013). Es el encargado de llevar todo el proceso de identificación del personal que reside o trabaja en la universidad, desde la solicitud de un documento de identificación, captura de datos, captura de imágenes, revisión de datos e imágenes, impresión de documentos de identificación, control de la calidad, y termina con la entrega de dicho documento. El análisis realizado al sistema proyectó que el mismo requiere de una mejora pues se hace necesario agilizar la gestión de solicitudes a la Oficina de Identificación y lograr un mejor control sobre la información referida al cobro que se lleva a cabo por concepto de pérdida o deterioro de la credencial. Para la realización de un módulo a este sistema se estudiaron otros sistemas semejantes que pudieran aportar información para el desarrollo del mismo.

1.3. Sistemas analizados en la investigación

La elección de la herramienta más conveniente depende de las necesidades de los clientes, así como de las funcionalidades que ofrezcan de base para soportar una implementación que responda a dichas necesidades. Durante el análisis de los diversos sistemas de solicitudes existentes que manejan información sobre el flujo de este proceso, se pudo constatar que poseen una variedad de funcionalidades competentes al área donde son aplicados.

En la búsqueda de respuestas a las necesidades que presenta el proceso de solicitud para la acreditación de la UCI, se analizaron diferentes sistemas ya creados para la gestión de solicitudes y acreditación. A través de este análisis se pretende conocer su funcionamiento y las características que pudieran servir de aporte para el desarrollo de la aplicación.

A continuación se ofrece una descripción de dichos sistemas de identificación de personas y de servicios de solicitudes en el mundo.

GREGAL (Sistema de gestión de soluciones informáticas):

GREGAL es un sistema de Gestión de solicitudes e incidencias informáticas. Sus objetivos son los de diseñar, desarrollar, documentar y mantener cuantas aplicaciones de gestión se realizan en el ASIC. Para ello, desarrolla aplicaciones que permiten realizar a través de Internet no solo consultas sino también procesos de gestión, e incluso todo tipo de pagos. Todos los servicios que proporciona el Servicio de Aplicaciones se basan en la infraestructura de red, servidores y comunicaciones corporativas, lo que requiere una intensa colaboración con el Servicio de Sistemas y Redes de Comunicación. Este sistema a pesar de brindar un servicio de solicitudes, no está acorde a las necesidades requeridas ya que el flujo de sus procesos no se relaciona con la propuesta de solución (Valencia, 2012).

EVOLIS EMEDIA SOFTWARE:

Es un programa que permite al usuario el diseño personalizado de credenciales para su posterior salida en las impresoras de tarjetas de PVC. En sus distintas versiones permite utilizar los datos automáticamente accediendo a una base de datos existente, la cual posee la información necesaria para la impresión en la tarjeta plástica. Posee un alto nivel de configuración y trabajo con imágenes. Este software es líder en impresoras de PVC de alta calidad a nivel mundial (Ferrer Martínez, 2008). Tiene más

de 10 años de experiencia y ha sido usado en más de 70 países. De este sistema se pudo identificar algunos servicios que si se relacionan con el proyecto a desarrollar, pero no cuenta con una plataforma para gestionar los procesos de solicitudes por lo cual no nos brinda información necesaria para el desarrollo de la propuesta de solución.

NUMBER FIVE CARDFIVE:

Es un software para el diseño de credenciales con diseño personalizado y soporte de las mayores impresoras de tarjetas PVC. El mismo ofrece flexibilidad para la creación y diseño de las propias credenciales PVC. El diseño a utilizar es fácil y avanzado. En sus distintas versiones permite utilizar los datos automáticamente mediante el acceso a una base de datos existente, donde se encuentra la información necesaria para la impresión en la tarjeta plástica (Gómez Urquiza, 2007). Este sistema a pesar de su semejanza en cuanto al flujo de los procesos con el sistema ya implementado para la emisión de credenciales, no cuenta con una plataforma para gestionar los procesos de solicitudes, por tanto no nos brinda información necesaria para el desarrollo de la propuesta de solución.

SUIN (Sistema Único de Identificación Nacional de la República de Cuba)

SUIN es un sistema desarrollado para la informatización de los procesos llevados a cabo en las oficinas del CIRP (Carné de Identidad y Registro de la Población). El mismo permite la administración y configuración en sistemas de gestión de información. El SUIN se compone de siete módulos. Estos tienen procesos que contienen mucha información y responsabilidades que son comunes entre ellos. Por esta razón el sistema se vale de un módulo de administración para la gestión y configuración del resto de los módulos, lográndose de esta manera:

- Configurar y gestionar los elementos que serán utilizados en la realización de los procesos de cada uno de los módulos.
- Recoger las funcionalidades principales de administración y seguridad de un sistema, así como aquellas que son indispensables para llevar a cabo cada uno de los procesos en dicho sistema.
- Llevar a cabo la gestión de los elementos fundamentales en la realización de cada uno de los trámites que se realicen, con el fin de garantizar el control de toda la información referente a los mismos.

El Módulo de Administración del SUIN posee tres roles fundamentales encargados de garantizar la administración según el nivel de acceso a la gestión de la información, ellos son: Administrador Nacional, Administrador Provincial y Administrador de Unidad. Con la implantación del Módulo de Administración para el SUIN se logra una estandarización en los procesos de gestión de direcciones de las personas (funcionalidad más importante que implementa este módulo), así como la manipulación normalizada de los nomencladores utilizados y la asignación homogénea de las direcciones a todos los ciudadanos cubanos y extranjeros residentes en el país (González, 2010). Es un sistema desarrollado en web por lo que no puede aportar toda la información que se requiere para el desarrollo de la solución propuesta.

Luego del análisis a estos sistemas, se identificaron un número de prestaciones y servicios que ofrecen, convirtiéndolos en una solución no viable, debido a que en algunos de los casos no se cuenta con documentación sobre el proceso de desarrollo de los mismos y aunque se tiene acceso a las tecnologías utilizadas, no se corresponden a la necesidad de gestionar los procesos de solicitud de credenciales ya que solo responden a las especificaciones de cada uno.

En la investigación realizada no se encontró ninguna aplicación que reúna todas las características que permitan gestionar procesos de solicitud para la emisión de credenciales de identificación, de ahí la necesidad de desarrollar una solución informática que estandarice la gestión de solicitud de credenciales a la oficina de Identificación de la UCI.

1.4. Tecnologías Software a utilizar

1.4.1. Frameworks

En el desarrollo de software, Marco de trabajo o *Framework* en su término en inglés, es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un *framework* puede incluir soporte de programas, bibliotecas y un lenguaje interpretado, entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Define una arquitectura adaptada a las particularidades de un determinado dominio de aplicación, definiendo de forma abstracta una serie de componentes y sus interfaces, y estableciendo las reglas y mecanismos de interacción entre ellos.

El interés en reutilizar software ha sido cambiado de la reutilización de componentes simples a diseño de sistemas enteros o estructuras de aplicaciones. Un sistema software que pudiera ser reutilizado en este nivel para la creación de aplicaciones completas es llamado *framework*. Los *Frameworks* están basados

en la idea que deberían permitir la producción fácil de un conjunto de sistemas específicos pero similares, dentro de un cierto dominio comenzando desde una estructura genérica. Brevemente, los *frameworks* son arquitecturas genéricas integradas por un extensible conjunto de componentes. Además, los *frameworks* pueden contener *subframeworks* los cuales representen subconjuntos de componentes de un sistema más grande.

NET Framework 4.0

El *Framework*. Net es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones, mediante la cual se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables.

Los principales componentes de este entorno son:

- Lenguajes de compilación
- Biblioteca de clases.Net
- CLR (Common Language Runtime)
- Ventajas:

Código administrado: El CLR realiza un control automático del código para que este sea seguro, es decir, controla los recursos del sistema para que la aplicación se ejecute correctamente.

Interoperabilidad multilenguaje: El código puede ser escrito en cualquier lenguaje compatible con .Net ya que siempre se compila en código intermedio (MSIL).

Compilación just-in-time: El compilador JIT incluido en el *Framework* compila el código intermedio (MSIL) generando el código máquina propio de la plataforma. Se aumenta así el rendimiento de la aplicación al ser específico para cada plataforma.

Garbage collector: El CLR proporciona un sistema automático de administración de memoria denominado recolector de basura (garbage collector). El CLR detecta cuándo el programa deja de utilizar la memoria y la libera automáticamente. De esta forma el programador no tiene por qué liberar la memoria de forma explícita aunque también sea posible hacerlo manualmente (mediante el método `dispose()` se libera el objeto para que el recolector de basura lo elimine de memoria).

Seguridad de acceso al código: Se puede especificar que una pieza de código tenga permisos de lectura de archivos pero no de escritura. Es posible aplicar distintos niveles de seguridad al código, de forma que se puede ejecutar código procedente de la web sin tener que preocuparse si esto va a estropear el sistema.

Despliegue: Por medio de los ensamblados resulta mucho más fácil el desarrollo de aplicaciones distribuidas y el mantenimiento de las mismas. El *Framework* realiza esta tarea de forma automática mejorando el rendimiento y asegurando el funcionamiento correcto de todas las aplicaciones.

NHibernate 3.1.0

NHibernate es un marco que permite hablar de una base de datos relacional de una manera orientada a objetos. Se puede almacenar (o como también se suele decir, "persistir") los objetos de una base de datos y cargar los objetos de la base de datos más adelante. *NHibernate* genera las sentencias SQL necesarias para insertar, actualizar, eliminar y cargar datos. No es el único marco ORM para .NET, pero es probablemente el más maduro y rico de las características de todos (Nicolas Schenker, 2011).

1.4.2. Lenguajes de programación

Un lenguaje de programación es la notación para la descripción precisa de algoritmos o programas informáticos. Son el conjunto de instrucciones que permiten al programador pensar de forma clara sobre la complejidad del problema a resolver, de manera que pueda ordenarlas para la creación de un programa ejecutable por la computadora.

C#

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común. Se utiliza para crear aplicaciones cliente de Windows, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos.

Entre sus principales características se destacan:

Sencillez: C# elimina elementos que otros lenguajes incluyen y que son innecesarios en .NET, por ejemplo: El código escrito en C# es auto contenido, lo que significa que no necesita de ficheros

adicionales tales como ficheros de cabecera. El tamaño de los tipos de datos básicos es fijo e independiente del compilador.

Orientación a componentes: La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir plácidamente propiedades (similares a campos de acceso controlado), eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros).

1.4.3. Entorno de desarrollo

Los Entornos de Desarrollo Integrado (IDE, por sus siglas en inglés) son programas con un conjunto de herramientas que brindan amplias comodidades en el desarrollo de aplicaciones. Normalmente está compuesto por:

- Un editor de código fuente
- Un compilador y/o un intérprete
- Automatización de generación de herramientas
- Un depurado

Las ventajas de usar IDE en el desarrollo de la problemática planteada son: completamiento de código, disminución en el tiempo de desarrollo, depuración de código y resaltado de sintaxis.

Entorno de Desarrollo Integrado: Microsoft Visual Studio.NET 2010

Microsoft Visual Studio.Net 2010 es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic .NET, F#, Java, *Python*, Ruby, PHP; al igual que entornos de desarrollo web como ASP.NET MVC, Django. Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET. Visual Studio 2010 Professional es la herramienta esencial para realizar tareas de desarrollo básico. Simplifica la compilación, la depuración y el despliegue de las aplicaciones en una variedad de plataformas incluyendo *SharePoint*. También viene con el soporte integrado para el desarrollo con pruebas y con las herramientas de depuración que ayudan a garantizar unas soluciones de alta calidad.

1.4.4. Gestor de Base de datos

- Características de los sistemas de gestión de bases de datos

Un sistema de gestión de bases de datos se puede definir como una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. Es un conjunto coordinado de programas, procedimientos, lenguajes, etc. que suministra, tanto a los usuarios no informáticos como a los analistas, programadores o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad. Posibilita que los tratamientos sean más eficientes y rápidos, dando la mayor flexibilidad posible a los usuarios.

Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad así como mejorar los mecanismos de seguridad de los datos y la privacidad (Ruiz., 2007)

Sus objetivos fundamentales son:

- Independencia de los datos y los programas de aplicación.
- Minimización de la redundancia.
- Integración y sincronización de las bases de datos.
- Integridad de los datos.
- Seguridad y protección de los datos.
- Facilidad de manipulación de la información.
- Control centralizado.

PostgreSQL

PostgreSQL, versión 9.1, es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia distribución de software Berkeley y con su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

- Características de este gestor de bases de datos:
 - Implementación del estándar SQL92/SQL99.

- Soporta distintos tipos de datos: Además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP) y cadenas de bits. También permite la creación de tipos propios.
 - Incorpora una estructura de datos array.
 - Incorpora funciones de diversa índole: manejo de fechas, geométricas y orientadas a operaciones con redes.
 - Permite la declaración de funciones propias, así como la definición de disparadores.
 - Soporta el uso de índices, reglas y vistas.
 - Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Ventajas:
 - PostgreSQL usa una estrategia de almacenamiento de filas para conseguir una mejor respuesta en ambientes de grandes volúmenes.
 - Herramientas gráficas de diseño y administración de BD: Existen varias herramientas gráficas de alta calidad para administrar las bases de datos PgAdmin, pgAccess).
 - Instalación ilimitada.
 - PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

Administrador de base de datos: PgAdmin III 1.12.1

PgAdmin III es una aplicación gráfica para administrar el sistema gestor de bases de datos PostgreSQL, siendo una de las más completas con licencia de código abierto. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL. PgAdmin III está diseñado para responder a las necesidades de los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La aplicación incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar comandos programados y soporte para la replicación. La conexión al servidor puede hacerse mediante la familia de

protocolos de internet (TCP/IP por sus siglas en inglés) y encriptarse mediante el protocolo de Capa de Conexión Segura (SSL por sus siglas en inglés) para mayor seguridad.

Sistemas de Gestión de Bases de Datos Documentales (SGBDD) o Sistemas de Recuperación de Información (SRI)

Las bases de datos que generan se construyen con información no estructurada tipo texto (documentos) sobre uno o varios temas. Cuando un usuario realiza una consulta en una base de datos documental, el sistema presenta como resultado, no una respuesta exacta, sino documentos útiles para satisfacer la pregunta del usuario. A este tipo de bases de datos se les denomina bases de datos documentales, y a los sistemas que las gestionan, Sistemas de Gestión de Bases de Datos Documentales (SGBDD) o Sistemas de Recuperación de Información (SRI).

Sistemas de Gestión de Bases de Datos Relacionales (SGBDR)

Las bases de datos que generan se construyen con información muy estructurada (datos) acerca de una organización o empresa determinada. Cuando un usuario realiza una consulta en una base de datos relacional, el sistema presenta como resultado la respuesta exacta a lo que se busca. A este tipo de bases de datos se les denomina bases de datos relacionales, y a los sistemas que las gestionan, Sistemas de Gestión de Bases de Datos Relacionales (SGBDR).

Se pueden distinguir cuatro tipos de contextos para usar mecanismos de seguridad: contra accesos indebidos a los datos, contra accesos no autorizados a la BD, contra destrucción causada por el entorno (fuego, inundación, robo), y contra fallos del propio sistema (fallos del hardware, del software, etc.)

Entre los SGBD comúnmente utilizados en el mundo se tiene Oracle, MySQL, Microsoft SQL Server, PostgreSQL, Interbase, entre otros.

Herramienta CASE seleccionada

A partir de un análisis realizado a las principales herramientas CASE anteriormente expuestas se concluyó que para el Modelado de la Base de Datos se utilizará el Visual Paradigm UML en su versión 8.0. Este software brinda múltiples ventajas al usuario como son: es fácil de instalar y actualizar, es un software multiplataforma. Visual Paradigm permite realizar todo tipo de diagramas de clases, generar código desde diagramas y generar documentación. Además, facilita la generación de bases de datos, transformación de diagramas de Entidad-Relación en tablas de base de datos e ingeniería inversa (Suárez, 2010).

Los beneficios de utilizar Visual Paradigm como herramienta CASE de modelado de software se muestran a continuación:

- Navegación intuitiva entre código y el modelo.
- Demanda en tiempo real, modelo incremental de viaje redondo y sincronización de código fuente.
- Superior entorno de modelado visual.
- Soporte completo de notaciones UML.
- Diagramas de diseño automático sofisticado.

El análisis de las diferentes tecnologías, metodologías y herramientas según las necesidades de la solución determinó la utilización de la metodología FDD Ágil, como lenguaje de modelado BPMN, Visual Paradigm 8.0 como herramienta de modelado y

PostgreSQL 9.1 como SGBD, Plataforma de desarrollo Microsoft .Net *framework v4.0*. Como lenguaje de programación C# usando el IDE de desarrollo Visual Estudio 2010.

Esta decisión ha sido tomada además de tener en cuenta las características analizadas de cada herramienta, porque la solución anterior fue construida haciendo uso del Visual Studio .NET.

1.5. Metodologías de software

Las metodologías software ocupan un lugar importante para producir software de calidad en cualquier contexto de desarrollo pues determinan la planeación y seguimiento del proyecto que se desarrolla. Una metodología puede ser una guía que va indicando qué hacer y cómo actuar cuando se quiere obtener un objetivo, o en términos informáticos, cuando se quiere desarrollar un producto de software.

Una Metodología de Desarrollo de Software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información (Boehm, 1988).

Se podría decir que en estos últimos años se han desarrollado dos corrientes en lo referente a los procesos de desarrollo, los llamados métodos pesados o tradicionales y ligeros. La diferencia fundamental entre ambos es que mientras los métodos pesados intentan conseguir el objetivo común por medio del orden y documentación, los métodos ligeros (también llamados métodos ágiles), tratan de mejorar la calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen en el proceso.

El **desarrollo ágil de software** refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan mediante la colaboración de grupos autos organizados y multidisciplinarios. Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos desarrollando software en lapsos cortos. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requisitos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, sino que la meta es tener una «demo» (sin errores) al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto (Ching, 2004).

Los métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación. La mayoría de los equipos ágiles están localizados en una simple oficina abierta, a veces llamadas "plataformas de lanzamiento" (*bullpen* en inglés). La oficina debe incluir revisores, escritores de documentación y ayuda, diseñadores de iteración y directores de proyecto. Los métodos ágiles también enfatizan que el software funcional es la primera medida del progreso. Combinado con la preferencia por las comunicaciones cara a cara, generalmente los métodos ágiles son criticados y tratados como "indisciplinados" por la falta de documentación técnica.

Por el contrario las metodologías tradicionales, están basadas en normas provenientes de estándares, seguidos por el entorno de desarrollo, con cierta resistencia a los cambios. El proceso es mucho más controlado, con numerosas políticas o normas, existe un contrato prefijado y el cliente interactúa con el equipo de desarrollo mediante reuniones, los grupos de trabajos son grandes y posiblemente distribuidos. Generan gran cantidad de artefactos y roles. La arquitectura del software es esencial y se expresa mediante modelos. Una de las más usadas es Proceso de Desarrollo Unificado (RUP).

1.5.1. *Feature Driven Development (FDD). Desarrollo Basado en Funcionalidades*

Es un proceso de desarrollo de software diseñado por Peter Coad, Erich Lefebvre y Jeff De Luca. Se basa en un proceso iterativo con iteraciones cortas que producen software funcional que el cliente y la dirección pueden ver y monitorizar.

FDD es un método ágil, iterativo y adaptativo. A diferencia de otras MA, no cubre todo el ciclo de vida sino solo las fases de diseño y construcción y se considera adecuado para proyectos grandes y de misión

crítica. Aunque hay coincidencias entre la programación orientada por rasgos y el desarrollo guiado por rasgos, FDD no necesariamente implementa *Feature Oriented Programming* (FOP).

FDD no requiere un modelo específico de proceso y se complementa con otras metodologías. Enfatiza cuestiones de calidad y define claramente entregas tangibles y formas de evaluación del progreso. Los principios de FDD son pocos y simples: Se requiere un sistema para construir sistemas si se pretende escalar a proyectos grandes. Un proceso simple y bien definido trabaja mejor. Los pasos de un proceso deben ser lógicos y su mérito inmediatamente obvio para cada miembro del equipo. Vanagloriarse del proceso puede impedir el trabajo real. Los buenos procesos van hasta el fondo del asunto, de modo que los miembros del equipo se puedan concentrar en los resultados. Los ciclos cortos, iterativos, orientados por rasgos son mejores.

FDD se utilizó por primera vez en grandes aplicaciones bancarias a fines de la década de 1990. Los autores sugieren su uso para proyectos nuevos o actualizaciones de sistemas existentes, y recomiendan adoptarlo en forma gradual. Aunque no hay evidencia amplia que documente sus éxitos, las grandes consultoras suelen recomendarlo incluso para delicados proyectos de misión crítica.

- Comparación con otras metodologías

Puesto que todos los procesos se centran en la producción de software es deseable una comparación, no en su conjunto, sino según los medios que emplean y sus resultados. A continuación se realiza una comparación entre FDD, RUP y XP.

- Tamaño de los equipos: RUP está pensado para proyectos y equipos grandes, en cuanto a tamaño y duración. FDD y XP se implementan mejor para proyectos cortos y equipos más pequeños, siendo quizás FDD más escalable que XP.
- Obtención de requisitos: RUP y XP crean como base *UseCases* y *UserStories*, por lo contrario FDD no define explícitamente esa parte del proyecto sobre la adquisición de requisitos.
- Evaluación del estado del proyecto: FDD es posiblemente el proceso más adecuado para definir métricas que definan el estado del proyecto, puesto que al dividirlos en unidades pequeñas es bastante sencillo hacer un seguimiento de las mismas. XP también define esos componentes pequeños. RUP por su parte, es tan grande y complejo en este sentido como en el resto, por lo que manejar el volumen de información que puede generar requiere mucho tiempo.

- Carga de trabajo: XP es un proceso ligero, esto es, que los creadores del proceso han tenido cuidado de no poner demasiadas tareas organizativas sobre los desarrolladores. RUP es un proceso pesado, basado mucho en la documentación, en la que no son deseables todos esos cambios volátiles. FDD es por su parte un proceso intermedio, en el sentido de que genera más documentación que XP pero menos que RUP.
 - Relación con el cliente: Con RUP se presentarán al cliente los artefactos del final de una fase, en contrapartida, el aseguramiento de la calidad en XP y FDD no se basa en formalismos en la documentación, si no en controles propios y una comunicación fluida con el cliente.
 - Conocimiento sobre la arquitectura: En RUP se intenta reducir la complejidad del software a producir a través de una planificación intensiva. En XP se consigue a través de la programación a pares que ya en la creación del código se puedan evitar errores y malos diseños. En FDD sin embargo se usan las sesiones de trabajo conjuntas en fase de diseño para conseguir una arquitectura sencilla y sin errores.
- ¿Por qué FDD?

Debido al estudio de las características del Sistema de identificación implementado con anterioridad y en explotación actualmente, se decide utilizar la misma metodología de este, además es una metodología de desarrollo ágil, que disminuye el riesgo de los proyectos, pues gracias a sus entregas tangibles y al constante monitoreo de su calidad, se asegura el firme avance del mismo. Al usar una metodología de Software acorde con el proyecto que se va a desarrollar se obtienen un gran número de ventajas.

- Ventajas:
 - Se preocupa por la calidad del software, por lo que incluye un monitoreo constante del proyecto.
 - Ayuda a contrarrestar situaciones como el exceso en el presupuesto, fallas en el programa o el hecho de entregar menos de lo deseado.
 - Propone tener etapas de cierre cada dos semanas. Se obtienen resultados periódicos y tangibles.
 - Se basa en un proceso iterativo con iteraciones cortas que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorear.

- Define claramente entregas tangibles y formas de evaluación del progreso del proyecto.
- No hace énfasis en la obtención de los requerimientos sino en cómo se realizan las fases de diseño y construcción.
- Procesos:
 1. Desarrollo de un modelo global
 2. Construcción de una lista de funcionalidades
 3. Planeación por funcionalidad
 4. Diseño por funcionalidad
 5. Construcción por funcionalidad.
 - Desarrollar un modelo global: Al inicio del desarrollo se construye un modelo teniendo en cuenta la visión, el contexto y los requisitos que debe tener el sistema a construir. Este modelo se divide en áreas que se analizan detalladamente. Se construye un diagrama de clases por cada área.
 - Construir una lista: Se elabora una lista que resuma las funcionalidades que debe tener el sistema, cuya lista es evaluada por el cliente. Cada funcionalidad de la lista se divide en funcionalidades más pequeñas para un mejor entendimiento del sistema.
 - Planear: Se procede a ordenar los conjuntos de funcionalidades conforme a su prioridad y dependencia, y se asigna a los programadores jefes.
 - Diseñar: Se selecciona un conjunto de funcionalidades de la lista. Se procede a diseñar y construir las funcionalidades mediante un proceso iterativo, decidiendo qué funcionalidades se van a realizar en cada iteración. Este proceso iterativo incluye inspección de diseño, codificación, pruebas unitarias, integración e inspección de código.
 - Construir: se procede a la construcción total del proyecto.
- Roles

Existen tres categorías de rol en FDD:

1. Roles claves.
2. Roles de soportes.

3. Roles adicionales.

Roles Claves

- Director del proyecto
- Arquitecto jefe
- Director de desarrollo
- Programador Jefe
- Expertos de dominio
- Propietario de clases

El equipo de trabajo está estructurado en jerarquías, siempre debe haber un jefe de proyecto, y aunque es un proceso considerado ligero también incluye documentación (la mínima necesaria para que algún nuevo integrante pueda entender el desarrollo de inmediato). Como parte del proceso fue necesario definir los roles y las responsabilidades para realizar las actividades del área de Gestión de la Configuración de Software, así como los involucrados relevantes en el proceso.

Director del Proyecto: Líder administrativo y financiero del proyecto. Protege al equipo de situaciones externas.

Arquitecto jefe: Realiza el diseño global del sistema. Ejecución de todas las etapas.

Director de desarrollo: Lleva diariamente las actividades de desarrollo. Resuelve conflictos en el equipo. Resuelve problemas referentes a recursos.

Programador Jefe: Analiza los requerimientos. Diseña el proyecto. Selecciona las funcionalidades a desarrollar de la última fase del FDD.

Expertos de dominio: Puede ser un usuario, un cliente, analista o una mezcla de estos. Poseen el conocimiento de los requerimientos del sistema. Pasa el conocimiento a los desarrolladores para que se asegure la entrega de un sistema completo.

Propietario de clases: Responsable del desarrollo de las clases que se le asignaron como propias. Participa en la decisión de que clase será incluida en la lista de funcionalidades de la próxima iteración.

Roles De Soporte

- Administrador De Dominio (*Domain Manager*).
- Director de liberaciones (*Release Manager*).
- Administrador del sistema (*System Administrator*).

Director de dominio: Lidera al grupo de expertos del dominio, resuelve sus diferencias de opinión concernientes a los requerimientos del sistema.

Director de liberaciones: Controla el avance del proceso mediante la revisión de los reportes del programador jefe. Reporta resultados obtenidos semanalmente al gerente y al cliente donde incluye el porcentaje de avance de cada funcionalidad.

Administrador del sistema: Configura, administra y repara los servidores, estaciones de trabajo y equipos de desarrollo y testeo utilizados por el equipo.

Roles Adicionales

- Probador (*Tester*).
- Escritores de documentos técnicos (*Technical Writer*).

Probador: Verifica que el sistema recién creado cumpla con los requerimientos del cliente, puede llegar a ser una persona independiente del equipo del proyecto.

Escritores de documentos técnicos: Prepara la documentación para los usuarios, que pueden formar parte o no del equipo del proyecto.

1.6. Conclusiones parciales

Se analizó el sistema IDBIOACCESS y se abordaron conceptos asociados a la gestión de solicitud de credenciales de identificación, permitiendo una mejor comprensión del objeto de estudio. El análisis de sistemas de similar propósito permitió ultimar que, como no satisfacen las necesidades y condiciones requeridas, es necesario desarrollar una solución informática que estandarice la gestión de solicitud de credenciales a la oficina de identificación. La decisión de las herramientas, lenguajes y proceso de desarrollo de software definidos por el Departamento de Desarrollo de la Dirección de Informatización (DIN), permitió progresar en los conocimientos necesarios para el desarrollo de una solución informática que facilite la mejora de gestión de solicitudes de credenciales a la oficina de Identificación en la UCI.

Capítulo 2: Descripción de la propuesta de solución

2.1. Introducción

En el presente capítulo se describen los procesos del negocio asociados a la gestión de solicitud de credenciales por diferentes motivos a la Oficina de Identificación en la UCI. Se presenta una descripción de los requisitos funcionales y no funcionales con los que debe cumplir la propuesta de solución así como las técnicas empleadas para su obtención. Se define la arquitectura y los patrones de diseño a utilizar en la propuesta.

2.2. Propuesta de solución

Luego de haber realizado un análisis sobre los problemas existentes en la Oficina de Identificación de la UCI, se propone el desarrollo de un módulo en el sistema de identificación IDBIOACCESS que sea capaz de agilizar el proceso de trámites de solicitud de credenciales y aumentar el control de información que se maneja por concepto de cobro de la credencial. Este sistema para su correcto funcionamiento debe mediante un servicio web consumir del SGU, considerada como el sistema que maneja los datos primarios de las personas de la UCI, la información necesaria para conformar la lista de solicitudes de credenciales. Estos datos son: nombre y apellidos de la persona, CI y el número de expediente.

Una vez recibida la solicitud, el sistema debe permitir a los usuarios la posibilidad de listar todas las solicitudes, crear una planilla de solicitud, generar un vale de cobro de la credencial en caso de pérdida o deterioro, imprimir el vale y al mismo tiempo el sistema anula la credencial anterior y la pasa al estado de captación de imagen para poder continuar con el proceso de acreditación, automáticamente se guardan todos los datos necesarios para la información a los especialistas. El usuario puede acceder a aquellas opciones a las que está autorizado y realizar en las interfaces solo los cambios que le sean permitidos dependiendo de los niveles de acceso que posea. Una vez concluido el desarrollo del módulo el sistema va a permitir agilizar el proceso de trámites de solicitud de credenciales dentro de la UCI y aumentar el control de información referente al cobro de la credencial en la oficina de Identificación.

- Información requerida

El sistema debe permitir actualizar y guardar la información necesaria que debe aparecer en las credenciales, como: nombre y apellidos de la persona, foto, nombre del área a la que pertenece, cargo,

número de solapín y el código de barras, además se hace necesario trabajar con el rol que desempeña, ID expediente y el número del CI de la persona.

2.2.1. Descripción del Sistema

Con el fin de mejorar el sistema de identificación y emisión de credenciales que se lleva a cabo en la UCI actualmente se ha decidido la realización del Módulo para la gestión de solicitudes de credenciales a la oficina de identificación, un sistema que sea capaz de dar solución a todas las dificultades existentes con respecto a la tramitación de las solicitudes por parte de los directivos de las diferentes áreas y aumentar el control de información referente al cobro de la credencial en la oficina de identificación. Este módulo de gestión de solicitudes está dividido en dos subprocesos: el subproceso de solicitud y el subproceso de emisión de vales.

Subproceso Solicitud: Comienza cuando un directivo envía una solicitud de emisión o cancelación de credenciales por motivos de pérdida, deterioro, baja con pérdida, baja con deterioro o una credencial impersonal para el caso de los visitantes. El sistema debe recibir estas solicitudes con todos los datos necesarios de la persona que se va a acreditar y del directivo que autoriza de forma tal que se garantice la veracidad de la información. Estos datos se validan por el sistema de SGU y se almacenan.

Subproceso Emisión de vale: Comienza cuando una solicitud, en dependencia del tipo que sea, requiere del vale para el cobro de la credencial, luego de ser verificada la solicitud, es aceptada y guardada por el sistema, inmediatamente se crea el vale con los datos necesarios referente a la persona que se va a acreditar, al tipo de pago que se va a efectuar y la opción para imprimir.

- Actores involucrados en la solución

Tabla 1: Actores que interactúan con el sistema

Personas relacionadas con el sistema	Justificación
Directivo	Es aquel que tiene acceso limitado en la aplicación ya que solamente puede autorizar y enviar la solicitud de credenciales.
Usuario	Es el técnico o especialista que trabaja con el sistema encargado de controlar toda la gestión de las credenciales. Es la persona que por el sistema puede listar todas las solicitudes recibidas, crea la planilla de solicitud, valida los datos, cancela la solicitud en caso necesario, genera el vale para el cobro de la credencial, define el

	tipo de pago que se va a efectuar, imprime, anula la credencial anterior y crea una nueva según el caso. Por último entrega el vale de cobro impreso a la persona para su firma y continúa con el proceso de acreditación.
Administrador	Es el encargado de restringir los permisos a los usuarios dentro del sistema.

2.3. Modelado del proceso de negocio

Un modelado de procesos de negocio es un conjunto estructurado y medible de actividades diseñadas para producir un producto especificado para un cliente o mercado específico. Implica un fuerte énfasis en cómo se ejecuta el trabajo dentro de la organización, en contraste con el énfasis en el qué, característico de la focalización en el producto (Davenport, 2010). Para el desarrollo eficiente del sistema, y satisfacer las peticiones del cliente, así como lograr que la aplicación responda a los requerimientos, se hace necesario comprender bien todos los procesos relacionados con las solicitudes para la identificación de personas de la Universidad.

2.3.1. Conceptos asociados al modelo del Negocio

Luego de verificar la no existencia de un proceso de gestión de solicitudes de credenciales en la oficina de Identificación de la Universidad de las Ciencias Informáticas, se realiza una propuesta del flujo de actividades, a través de la modelación del proceso de gestión de solicitudes para lograr un completo entendimiento de dominio.

Persona: Todo ciudadano que reside en la Universidad o accede a ella y porta una credencial identificativa de la UCI, ejemplo: estudiantes, profesores, trabajadores, tercerizados, familiares, visitantes y otros que prestan servicios en ella.

Solicitud: Documento para autenticar la veracidad de la necesidad de crear una credencial a determinada persona por un motivo específico. Debe incluir los datos personales de la persona que solicita y del directivo que autoriza como son: Nombre y apellidos, número de solapín, número de Carné de Identidad, número ID, área en la que trabaja y rol que desempeña.

Planilla: Documento que se crea para insertar y guardar en BD toda la información necesaria referente a la persona que solicita ser acreditado y del directivo que autoriza a realizar esta acción. Incluye además como datos, el tipo de solicitud que puede ser por deterioro, por pérdida, por baja con pérdida de la

credencial, baja con deterioro del de la credencial o impersonal para el caso de los visitantes, la fecha que se solicita y una descripción de los motivos por lo que se hace la solicitud.

Vale de Pago: Un vale emitido, foliado y aprobado por la dirección de economía para la constancia del saldo que se le cobra a la persona por concepto de pérdida o deterioro de la credencial. Lleva como datos el nombre de la entidad, número de serie, fecha de emisión, rol de la persona, nombre y apellidos, tipo de pago (efectivo o por descuento), firma y cuño del funcionario y de la persona que paga.

Credencial: Documento para identificar a las personas pertenecientes a la UCI, el mismo posee varios datos como: nombre completo de la persona, número de solapín, número de serie, área en la que trabaja, código de barra que lo identifica, rol que desempeña y si la persona vive o no en el centro.

Tipo de solicitud: Motivo por el cual el directivo realiza la solicitud para una nueva credencial a un trabajador o estudiante. Los motivos pueden ser 5: por pérdida, por deterioro de la credencial, impersonal en el caso de los visitantes, por baja con pérdida o baja con deterioro de la credencial.

Registro: Cada planilla después de ser procesada genera un registro para archivar en base de datos la información de forma persistente.

2.3.2. Diagrama del Modelo de negocio

La metodología FDD recomienda para un mejor entendimiento del producto a desarrollar crear un modelo global, el cual, en este trabajo, se evidencia a través del modelo del negocio. La siguiente figura muestra el diagrama del negocio actual donde se refleja con color rosado las actividades ya informatizadas que se realizan por el sistema IDBIOACCESS, las reflejadas en color amarillo son las que actualmente se realizan a mano indicando la necesidad y la importancia de informatizar esta parte del flujo del proceso de identificación.

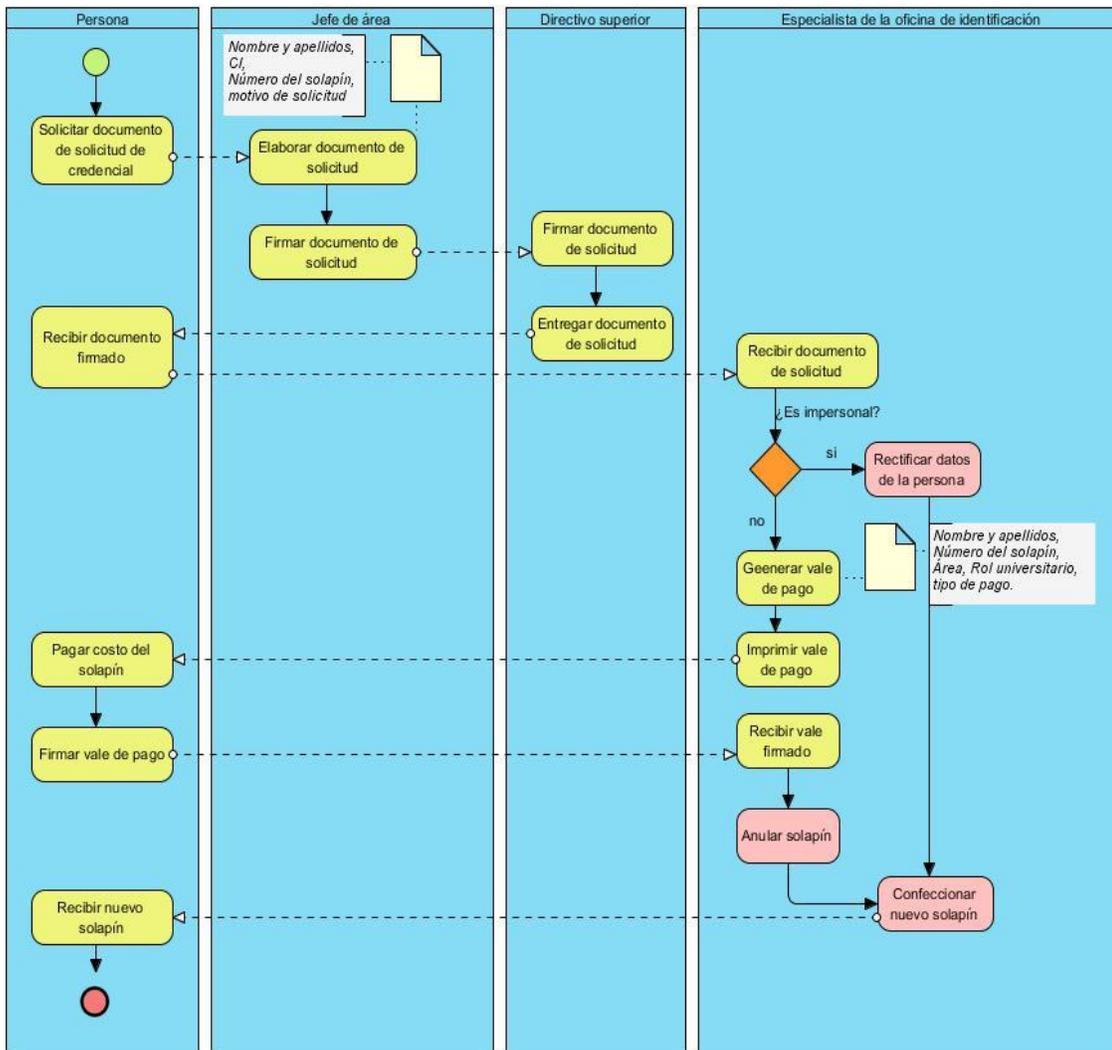


Figura 1: Diagrama del Modelo de negocio

2.3.3. Diagrama de las actividades del sistema

Para lograr un entendimiento del flujo de acciones que se llevarán a cabo para gestionar las solicitudes, se muestra en la figura 2 un diagrama de las actividades que conforman el proceso y se realiza la descripción de cada una de ellas.

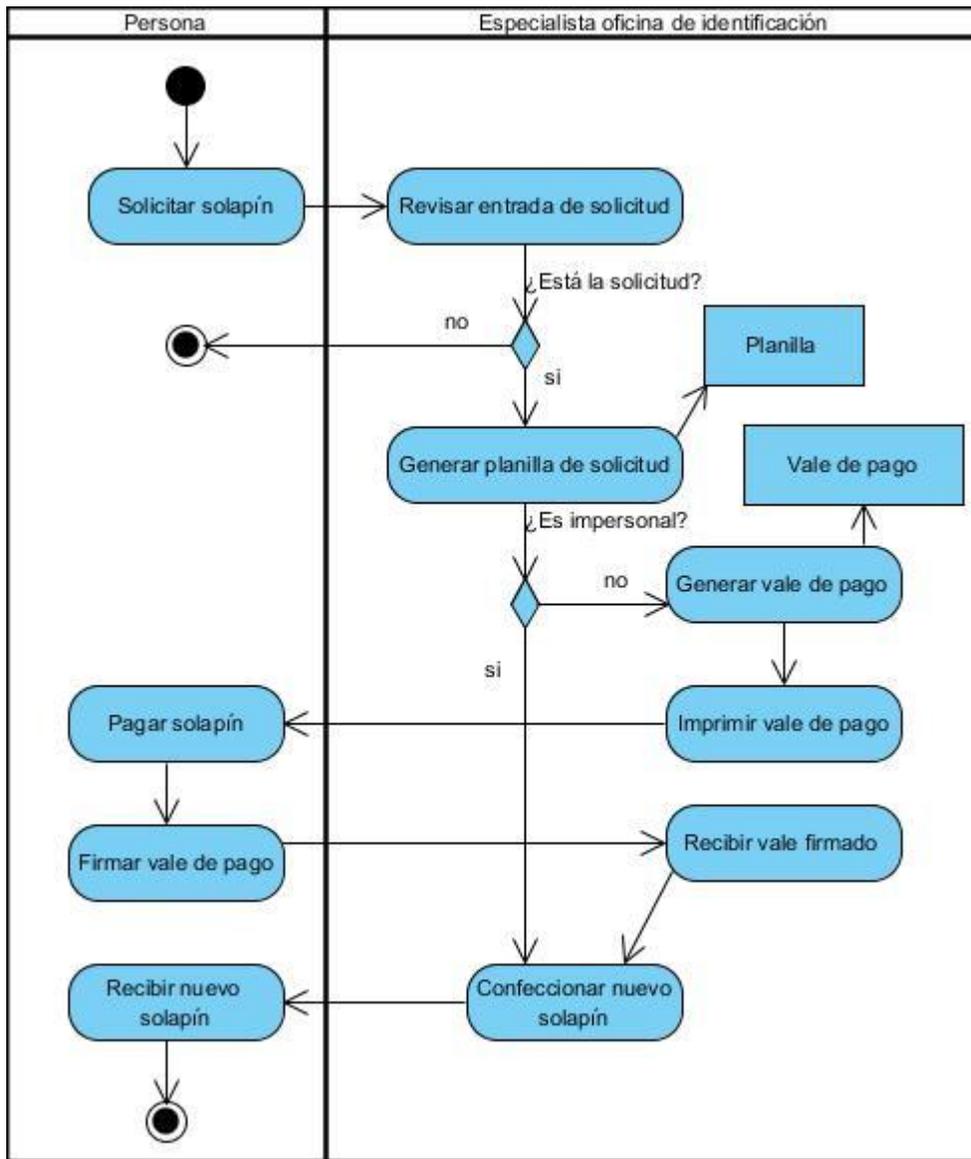


Figura 2: Diagrama de actividades.

En la siguiente tabla se muestra la descripción de cada una de las actividades que conforman el proceso de solicitud de credenciales.

Tabla 2: Descripción del flujo de actividades

Actividad	Descripción	Responsable	Entrada	Salida
Presentarse a la oficina de identificación a solicitar la credencial	La persona se dirige a la oficina donde será atendido por un especialista autorizado a registrar una solicitud por el sistema.	Persona	Necesidad de obtener el documento de identificación	Se dirige personalmente a la oficina.
Verificar solicitud por sistema IDBIOACCE SS.	El especialista de la oficina verifica que por el sistema ya esté recibida la solicitud enviada por el directivo responsable para poder tramitarla.	Especialista	Solicitud recibida en el sistema.	Solicitud recibida en el sistema.
Revisión de los datos en la solicitud.	El especialista responsable por las solicitudes verifica que según los datos de la solicitud recibida, el ciudadano cumple con los requisitos necesarios de la institución para la emisión del documento, verificando además que todos los datos estén correctos.	Especialista	Datos de la solicitud registrados en el sistema.	Datos de la solicitud revisados.
Aceptar solicitud.	Una vez que el especialista autorizado a registrar una solicitud en el sistema tiene los datos necesarios, procede a aceptar la solicitud en la aplicación.	Especialista	Solicitud lista	Solicitud aceptada.
Registrar información del ciudadano.	El sistema registra los datos de la solicitud aceptada por el usuario, realizando las validaciones necesarias.	Especialista	Solicitud aceptada.	Datos de la solicitud registrados en el sistema.
Notificar datos guardados.	En el caso de que la solicitud sea aprobada el sistema notifica que todos los datos fueron salvados.	Especialista	Datos de la solicitud registrados en el sistema.	Datos de la solicitud guardados por el sistema.
Crear el	En el caso de que la	Especialista	Disponibilidad	Vale creado

Capítulo 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.

comprobante de pago.	solicitud sea por motivos de pérdida, deterioro, baja con pérdida o baja con deterioro el sistema brinda al especialista la opción de generar vale.		de generar el vale.	por el sistema.
Registrar información del ciudadano y tipo de pago.	El sistema registra los datos del vale realizando las validaciones necesarias en cada campo de datos.	Especialista	Datos del vale validados por el sistema.	Datos del vale registrados en el sistema.
Emisión del vale de pago.	El especialista acepta la opción de imprimir vale en la aplicación.	Especialista	Disponibilidad para la impresión del vale.	Impresión aceptada.
Emisión del vale para el pago del solapín.	El sistema emite al especialista y a la persona un comprobante de pago donde queda reflejado los datos del ciudadano, el precio de la credencial y el tipo de pago.	Especialista	Disponibilidad comprobada.	Comprobante de pago emitido por el sistema.
Anulación del solapín anterior.	El sistema, una vez impreso el vale de pago automáticamente anula la credencial anterior y lo pasa al estado de captación de imagen.	Especialista	Vale de pago impreso.	Solapín anterior anulado.
Notificar datos guardados.	Una vez que el vale sea impreso el sistema notifica que todos los datos fueron salvados correctamente.	Especialista	Vale impreso.	Datos del vale guardados por el sistema.
Recibir comprobante de pago.	La persona recibe el comprobante emitido por el sistema y lo firma como constancia, además debe ser firmado por el especialista.	Persona y especialista	Comprobante de pago emitido por el sistema.	Comprobante de pago recibido por la persona y el especialista.

2.3.4. Reglas del negocio

Las reglas del negocio son restricciones que las organizaciones tienen definidas y son vitales para lograr sus objetivos, además se deben tener en cuenta para modelar o automatizar el proceso de solicitud. Dichas reglas son:

- La persona debe estar registrada por SGU de la UCI.
- El especialista debe recibir la solicitud antes de acreditar a la persona.
- La persona debe traer una identificación válida para comprobar los datos de la solicitud y poder acreditarse.
- La persona no debe tener más de una credencial válida que la identifique.
- La persona debe traer su credencial en caso de cambio por deterioro.
- La persona debe reportar su credencial en caso de pérdida o deterioro.
- La persona debe pagar el dinero en caso de pérdida o deterioro de su credencial.
- El vale para el cobro de la credencial debe ser único a través de un número de serie consecutivo.

Cuando en la oficina de Identificación se recibe una solicitud para crear una nueva credencial en caso de los impersonales se verifican los datos con el carné de identidad de la persona, se crea la planilla de solicitud por sistema con los datos necesarios, se acepta la solicitud, se guardan los datos para los reportes e informes de la oficina y se procede a realizar por captación de imagen el proceso de identificación completo al trabajador. En caso de pérdida, baja con deterioro, baja con pérdida o deterioro, la credencial debe estar en estado activo, se toman de la solicitud los datos de la persona, se crea la planilla por sistema, se verifica el rol de la persona y se genera el vale para el cobro de la credencial que construye el sistema, el usuario paga, se imprime el vale, automáticamente la credencial anterior se anula y pasa a estado de captación de imagen para procesar una nueva credencial. Todos estos datos correspondientes a la planilla y al vale de pago quedan guardados para reportes necesarios.

2.4. Requisitos del Software

Un requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de este. En el otro extremo, es una definición detallada y formal de una función del sistema (Sommerville, 2004).

2.4.1. Técnicas de obtención de requisitos

La obtención de requisitos es el proceso mediante el cual los interesados en un sistema de software descubren, revelan, articulan y entienden con diagramas, matrices y métodos de evaluación para extraer los requisitos esperados e intentar obtener requisitos innovadores (Pressman, 2005).

La obtención de requerimientos es una de las cuatro actividades que define la Ingeniería de Requisitos, para ayudar a reconocer la importancia que tiene una especificación y administración adecuada de los requerimientos de los usuarios. Existen diferentes técnicas para identificar los requisitos, dentro de las que se encuentran: sesiones de tormentas de ideas, entrevistas, observación de campo, revisión de la documentación técnica, análisis de sistemas existentes, ingeniería inversa, simulaciones y prototipos. (Pressman, 2007).

A continuación se describen las técnicas utilizadas durante el proceso de desarrollo del software para recopilar los requisitos de las actividades de gestión de solicitudes de credenciales a la Oficina de Identificación de la UCI:

Entrevista: se realizaron entrevistas no formales a los técnicos, a los especialistas que trabajan con el SGU y al personal que trabaja con el sistema de identificación en la oficina, lo que permitió la interacción con el especialista superior del área a tratar y entender el negocio. Además se entrevistaron a estudiantes y asistentes de control de algunos directivos, para conocer las dificultades o inconformidades que se les presentan cuando se pierde una credencial. Todas estas opiniones ayudaron a entender cuáles eran las mayores prioridades de los usuarios y comprender la definición de los requisitos del sistema.

Sesiones de tormentas de ideas: es una técnica de reuniones en grupo que permite generar ideas originales en un ambiente libre de críticas. Se realizaron reuniones con la participación del cliente e integrantes del proyecto, con el objetivo de definir las funcionalidades y esclarecer las dudas en el proceso de gestión de solicitud de credenciales en la universidad. A partir de las reuniones se encontraron algunas funcionalidades e interfaces que sirvieron de base para la propuesta de solución.

Sistemas existentes: se realizó un análisis a distintos sistemas ya desarrollados que están relacionados con la gestión de solicitudes o de emisión de credenciales. Esto permitió llegar a la conclusión que ninguno de estos sistemas aporta elementos necesarios para el desarrollo de la propuesta de solución.

El resultado de las entrevistas realizadas se pueden observar en el [Anexo 1](#).

2.4.2. Requisitos Funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, la manera en que este debe reaccionar a entradas de datos y cómo debe comportarse en situaciones particulares. En algunos casos, los requisitos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (Sommerville, 2004).

1. Mostrar solicitudes de credenciales.
2. Generar planilla de solicitud.
3. Aceptar solicitud de credencial.
4. Obtener lista de solicitudes aceptadas.
5. Imprimir solicitud.
6. Cancelar solicitud.
7. Generar vale de pago.
8. Mostrar vale de pago.
9. Imprimir vale.
10. Anular solapín.
11. Actualizar estado de solapín.
12. Emitir reportes por tipo de solicitud.
13. Emitir reportes de solicitud por fecha.
14. Emitir reportes de solicitud por área.
15. Emitir reportes por cantidad total de solicitudes.
16. Emitir reportes por cantidad total de vales.
17. Emitir reportes de vales por rol de la persona.
18. Emitir reportes de vales por áreas.
19. Emitir reportes de vales por fecha.

Luego de haber definido los principales requisitos funcionales se procede a determinar la complejidad de cada uno de ellos basados en el expediente de proyecto del proceso de mejora donde se analizan una serie de criterios individualmente concluyendo si el requisito es de complejidad alta, media o baja.

La clasificación de la complejidad permite estimar el esfuerzo de implementación del requisito y contribuye a la decisión sobre la inclusión en las etapas del desarrollo del software.

La lista de todos los requisitos y su prioridad se pueden ver en el [Anexo 2](#).

2.4.3. Requisitos no funcionales

Los requisitos no funcionales imponen restricciones en el diseño o la implementación. Son propiedades o cualidades que el producto debe tener. (Sommerville, 2002). Una vez que se conoce lo que el sistema debe hacer, se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Los requisitos no funcionales son fundamentales en el éxito del producto.

A continuación se muestran los requisitos no funcionales definidos para la propuesta de solución.

Usabilidad

- Menús laterales y desplegados que permitan el acceso rápido a la información.
- Las vistas del sistema deben indicar en cada momento la acción que se está realizando así como los menús deben estar acordes a la acción que se realiza.

Software

- Cliente- Sistema operativo Windows7 o superior, Microsoft *.Net framework* v4.0. Además del software necesario para cada uno de los dispositivos utilizados por el sistema.
- Servidor- Sistema operativo Windows Server 2008 R2 con SP2, Microsoft *.Net framework* v4.0. IIS 7.5 y PostgreSQL 9.1.

Hardware

- Cliente- PC Pentium 4 a 1 GHz o superior, mínimo 1 GB de RAM, 250 GB o superior de disco duro.
- Servidor- PC Pentium 4 a 2 GHz o superior, mínimo 2 GB de RAM, 250 GB o superior de disco duro.

Seguridad

- Solo se debe mostrar a los usuarios aquellas acciones o informaciones a las que por su responsabilidad o rol dentro del negocio necesitan acceder.
- El tratamiento de las excepciones permitirá un seguimiento hasta guardar información acerca del lugar donde se produjo el error y de los parámetros de configuración del sistema que lo provocaron.

Restricciones de diseño

- Plataforma de desarrollo.NET 4.0 utilizando Visual Studio 2010.
- Para el acceso a datos se utilizará el ORM NHibernate 3.1.0. Los artefactos de diseño del software se encuentran en Visual Paradigm.
- Descripción de requisitos.

La descripción de requisitos es la base que permite verificar si se alcanzaron o no los objetivos establecidos en el proyecto, ya que estos son un reflejo detallado de las necesidades de los clientes o usuarios del sistema (GIRALDO, 2007). La metodología FDD plantea que los requisitos funcionales del sistema se deben dividir en subconjuntos, agrupándolos según la afinidad y dependencia que exista entre ellos. Luego se describen estos requisitos detalladamente con el objetivo de definir paso a paso las acciones que deben seguir los desarrolladores, para garantizar una mayor coherencia de la aplicación con las expectativas del cliente.

A continuación, se muestra el requisitos "Generar Vale de pago" descrito. Para consultar las descripciones de requisitos restantes, ver [Anexo 3](#)

Tabla 3: Descripción de requisitos

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF 7	Generar Vale de pago	El requisito permite crear un vale para el cobro de la credencial. El requisito comienza cuando el técnico o especialista que labora con el sistema después de haber atendido la solicitud por motivos de pérdida, baja con pérdida, deterioro, o baja con deterioro y comprobar todos los datos referentes a la persona, da un click en Continuar.	Alta	Alta

	<p>Inmediatamente el sistema genera un vale.</p> <p>Se muestra un formulario de Cobro de Solapín.</p> <p>En este formulario se muestran los datos: nombre de la entidad, fecha de creado el vale, número de folio que es único y consecutivo, nombre del área o dirección a la que pertenece la persona que solicita, nombre y apellidos, ID expediente y número CI, rol de la persona, tipo de pago (efectivo o descuento) y pie de firma.</p> <p>El usuario debe especificar el tipo de pago que se va a efectuar marcando una de las dos opciones que se brindan. Se imprime el vale como constancia de la acción e inmediatamente el sistema anula la credencial anterior y lo pasa al estado Captación de imagen para continuar con el proceso.</p>		
--	--	--	--

Funcionalidades Tratadas: RF 8 Mostrar vale, RF 9 Definir tipo de pago, RF 10 Imprimir vale.

Prototipo generar vale.

The screenshot displays the XABAL IDBIOACCESS web application interface. The header includes the logo and the text 'Plataforma Modular de Identificación y Control de Acceso'. The user 'Estrella Ramos Pérez' is logged in. The main content area is titled 'Impresión de Datos' and shows a table with the following data:

Nombre Completo	Solapín	No. CI
Daniel Pelaez Garcia	E126012	90042
Franck Carlos Arreguiña Garcia	92100	

Below the table, there is a form for generating a receipt. The form includes the following fields and options:

- UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS N° 8451 Folio: [] Fecha: []
- DIRECCIÓN DE CONTABILIDAD Y FINANZAS
- COBRO POR PERDIDA DEL SOLAPÍN
- PARA USO DE LA DIRECCIÓN DE SEGURIDAD Y PROTECCIÓN
- TRABAJADOR ESTUDIANTE
- Nº. SOLAPÍN: [] ÁREA: []
- NOMBRE Y APELLIDOS: []
- FORMA DE PAGO:
 - EFFECTIVO
 - DESCUENTO POR NÓMINA
 - IMPORTE: \$ []
 - La deducción se realizará en un solo pago
- Firma del que autoriza [] Firma del que recibe y Cede []

At the bottom of the form, there are buttons for 'Actualizar' and 'Imprimir', and a 'Total: 2' indicator.

Observaciones	<ol style="list-style-type: none">1. Para realizar la acción el usuario tiene que estar autenticado en el sistema.2. Solo se mostrarán las solicitudes recibidas.3. Una persona solo puede tener asignado una credencial activa.4. Cada vez que se imprima un vale de cobro se debe anular la credencial activa perteneciente a la persona.5. Si el usuario presiona el botón Cancelar se muestra el mensaje de confirmación: “¿Está seguro de realizar la acción?”, si presiona Aceptar regresa al listado de solicitudes. Si presiona el botón Cancelar se mantiene en la página sin realizar ninguna operación.
----------------------	--

2.5. Planeación por funcionalidad.

Después de analizar las funcionalidades del módulo, la prioridad de cada requisito por la complejidad que tienen en el negocio y de haber sido evaluada por el cliente, la metodología FDD, como parte de la Fase Planeación por funcionalidades sugiere realizar el “Plan de Iteraciones”. Este consiste en asignarle a cada iteración las funcionalidades según su prioridad, determinando el tiempo de duración de cada iteración y el orden de desarrollo.

En base a lo antes mencionado se decide realizar el sistema en 3 iteraciones con duración de 3 a 4 semanas cada una, donde 1 semana equivale a 5 días y las jornadas laborales serán de 8 horas por día, a continuación se detallan las iteraciones:

Iteración 1

En esta iteración se desarrollan las funcionalidades de prioridad Alta, las relacionadas con el subproceso Crear solicitud, Crear vale y Generar reportes. Las pruebas de caja negra se efectuaron para probar que la aplicación permita crear una planilla y genere el vale de pago con los datos correspondiente a dicha planilla.

Iteración 2

En esta iteración se desarrollan las funcionalidades de prioridad Media, las relacionadas con el subproceso de Listar solicitudes y emitir notificaciones. Las pruebas de funcionalidades se realizaron probando que la aplicación mostrara todas las solicitudes recibidas.

Iteración 3

En esta iteración se desarrollan las funcionalidades de prioridad Baja, las relacionadas con el subproceso de impresión de solicitud e impresión de vale. Se probó a través de las pruebas de funcionalidades que la aplicación mostrara la opción de imprimir y permitiera realizar la acción.

A cada iteración se le realizan las pruebas de funcionalidades para ir depurando errores y entregar al cliente un buen resultado de esta etapa de desarrollo.

Luego de algunas iteraciones es recomendable realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario

2.6. Diseño por funcionalidad.

La metodología FDD propone un diseño sencillo, donde siempre se intenta tener el código más simple, menos redundante y con las funcionalidades estrictamente necesarias. No requiere de la descripción del sistema por medio de diagramas de clase utilizando notación UML, sino que se guía por técnicas. Esto no implica que no se utilicen los diagramas para obtener una mejor visión y comunicación entre el equipo de trabajo, siempre y cuando su complejidad no sea alta y defina información importante.

2.6.1. Arquitectura propuesta.

La arquitectura de software es el diseño de más alto nivel de un sistema describiendo cada uno de los componentes que lo integran así como la comunicación entre los mismos. Se establecen los fundamentos para que analistas, diseñadores, programadores trabajen en una línea común que permita alcanzar los objetivos del sistema. La arquitectura de software para la propuesta de solución tiene la responsabilidad de:

- Definir los módulos principales.
- Definir las responsabilidades que tendrá cada uno de estos módulos.
- Definir la interacción que existirá entre dichos módulos.
- Control y flujo de datos.
- Secuenciación de la información.
- Protocolos de interacción y comunicación.
- Ubicación en el hardware.

El módulo se encuentra desarrollado sobre la plataforma .Net, en su versión 4.0, es una solución Cliente–Servidor que define la relación entre dos aplicaciones en las cuales una de ellas (Cliente) envía peticiones a la otra (Servidor) y este último le envía las respuestas. Para que los clientes y los servidores puedan establecer una comunicación se necesita una Infraestructura de comunicaciones o protocolo de comunicación, estos brindan los mecanismos básicos para el direccionamiento y transporte de los datos. Esta arquitectura presenta disimiles ventajas ya que la aplicación cliente puede actuar tanto como una sola entidad que como entidades separadas y las funciones de cada uno pueden correr sobre plataformas diferentes, además de que la aplicación cliente no necesita conocer la lógica del servidor, ya que solo interactúa con esta a través de una interfaz y los cambios que puedan surgir en el servidor implican pocos o ningún cambio en el cliente.

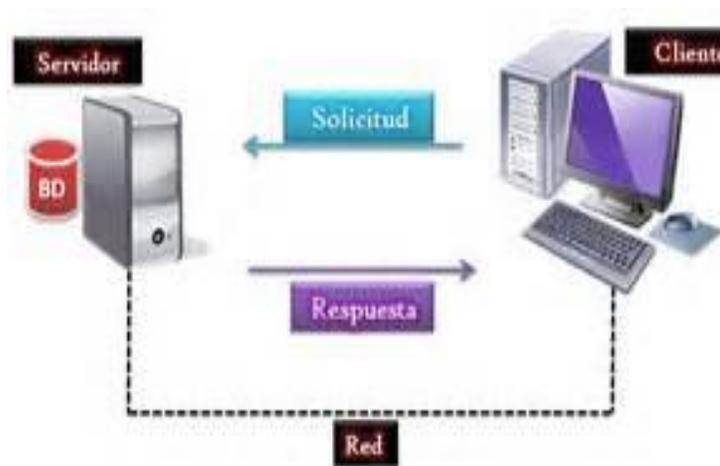


Figura 3: Arquitectura cliente-servidor.

El estilo arquitectónico en capas se basa en una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver. El sistema se encuentra dividido en 3 capas definiendo claramente las responsabilidades de cada una. Estas capas están distribuidas de manera tal que los componentes de una capa solo pueden acceder a los componentes de la capa inmediata inferior, de esta manera se reduce la dependencia entre capas, ya que las capas inferiores no tienen conocimiento sobre los detalles de las capas superiores. Esta vista permite la realización de cambios en las capas sin realizar grandes cambios en las demás.

En la figura 4 se muestra la Vista lógica de las capas.

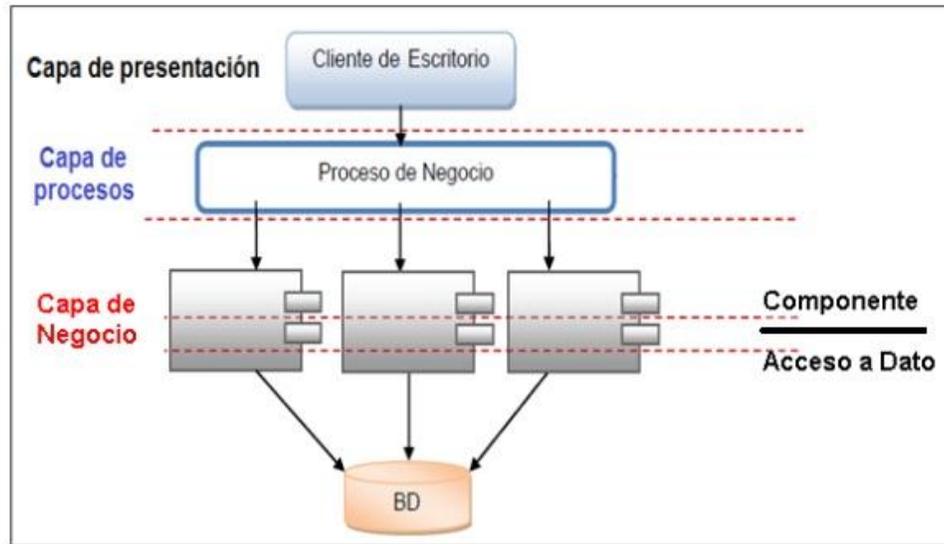


Figura 4: Vista Lógica de la arquitectura

Capa de Presentación: Es la capa donde el sistema interactúa con el usuario, haciendo uso de varias tecnologías para la validación de los datos de entrada así como el uso de componentes. En esta capa se encuentran todas las interfaces que serán mostradas a los usuarios.

Capa de Negocio: Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. En esta capa se encuentra el ORM (NHibernate), herramienta que tiene como objetivo principal mapear los objetos desde una aplicación .Net a una base de datos relacional.

Capa de Datos: Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

2.6.2. Patrones de Diseño.

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Para que una solución sea considerada un patrón debe poseer varias características como: haber comprobado su efectividad resolviendo problemas similares en ocasiones

anteriores, debe ser reutilizable, o sea, ser aplicable a diferentes problemas de diseño en distintas circunstancias.

Los patrones de diseño tienen como objetivo:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores sintetizando conocimiento ya existente.

Patrones GRASP (para la Asignación General de Responsabilidad): Estos patrones son utilizados en la mayoría de las clases diseñadas para el desarrollo de las aplicaciones. Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

- **Bajo Acoplamiento:** Uno de los principales síntomas de un mal diseño y alto acoplamiento es una herencia muy profunda. Debe haber pocas dependencias entre las clases. Uno de los principios para proteger al software frente al cambio es mantener bajo el acoplamiento entre clases, cuanto menor sea el acoplamiento entre clases, menor influencia tendrán los cambios. Cada clase se relaciona solo con quien lo necesita para realizar sus funcionalidades. Ejemplo: la clase "SolicitudCredencial" solo se relaciona con la clase "INegocioIdentificacion".
- **Patrón Experto:** Es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Con él no se pretende designar una idea oscura ni extraña; expresa simplemente la "intuición" de que los objetos hacen cosas relacionadas con la información que poseen. Ejemplo: la clase "SolicitudCredencial" la cual contiene entre sus métodos `MostrarSolicitudes()`, `MostrarReportes()`, puesto que es la clase indicada en estas funciones.
- **Patrón Creador:** Los patrones de diseño de creación abstraen el proceso de creación de instancia. Ayudan a hacer a un sistema independiente de cómo se crea, se compone y se representan sus

objetos. Estos patrones encapsulan el conocimiento sobre las clases concretas que usa el sistema y ocultan como se crean y se asocian las instancias de estas clases. Ejemplo, asigna a la clase RepositoryFactory la responsabilidad de crear una instancia de la clase Ciudadano.

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

Figura 5: Ejemplo del uso del patrón Creador.

Los patrones GOF que contribuyen a la construcción de un diseño más elegante y robusto. Dentro de los patrones GOF (Patrones de diseño a objeto) están los llamados Patrones de Creación, Estructurales y de Comportamiento. En el diseño del módulo se visualizan los siguientes patrones.

- Creacionales:

Patrones creacionales tratan con las formas de crear instancias de objetos. El objetivo de estos patrones es de abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados.

- Singleton (Instancia única): Este patrón se utilizó para garantizar que una clase solo tenga una única instancia y proporcionar un punto de acceso global a ella. Reduce el espacio de nombres, es una mejora sobre las variables globales, además de que es usado debido a la necesidad de trabajar con el mismo objeto en distintos momentos. Ejemplo: la clase “Negocioidentificacion” que hace instancia a “RepositoryFactory”.

```
public static RepositoryFactory Instance
{
    Get
    {
        if (RepositoryFactory._instance == null)
        {
            RepositoryFactory._instance = new RepositoryFactory();
        }
        return RepositoryFactory._instance;
    }
}
```

Figura 6: Ejemplo del uso del patrón Singleton.

- Abstract Factory (fábrica abstracta): Permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí y haciendo transparente el tipo de familia concreta que se esté usando. El problema a solucionar por este patrón es el de crear diferentes familias de objetos, como por ejemplo la creación de interfaces gráficas de distintos tipos (ventana, menú, botón, etc.).
- Builder (constructor virtual): Abstrae el proceso de creación de un objeto complejo, centralizando dicho proceso en un único punto.
- Factory Method (método de fabricación): Centraliza en una clase constructora la creación de objetos de un subtipo de un tipo determinado, ocultando al usuario la casuística, es decir, la diversidad de casos particulares que se pueden prever, para elegir el subtipo que crear. Parte del principio de que las subclasses determinan la clase a implementar. A continuación se muestra un ejemplo de este patrón.

- Estructurales:

Los patrones estructurales describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos.

- Facade: Proporciona una interfaz unificada para un conjunto de interfaces de un subsistema. Define una interfaz de alto nivel que hace que el subsistema sea más fácil de usar. Ejemplo la interfaz `INegocioIdentificacion` es utilizada para otras interfaces como `IValePago`.

- Adapter o Wrapper (Adaptador o Envoltorio): Adapta una interfaz para que pueda ser utilizada por una clase que de otro modo no podría utilizarla.
- Bridge (Puente): Desacopla una abstracción de su implementación.
- Composite (Objeto compuesto): Permite tratar objetos compuestos como uno solo.
- Decorator (Decorador): Añade funcionalidad a una clase dinámicamente.
- Módulo: Agrupa varios elementos relacionados, como clases, singletons, y métodos, utilizados globalmente, en una entidad única

2.6.3. Diagrama de Paquetes

Los diagramas de paquetes muestran la descomposición jerárquica lógica de un sistema, así como las dependencias entre las partes lógicas que lo conforman. A continuación se muestra el diagrama de paquetes del sistema IDBIOACCESS, en el cual se puede apreciar que la estructura organizativa de los paquetes responde a la arquitectura definida para el módulo.

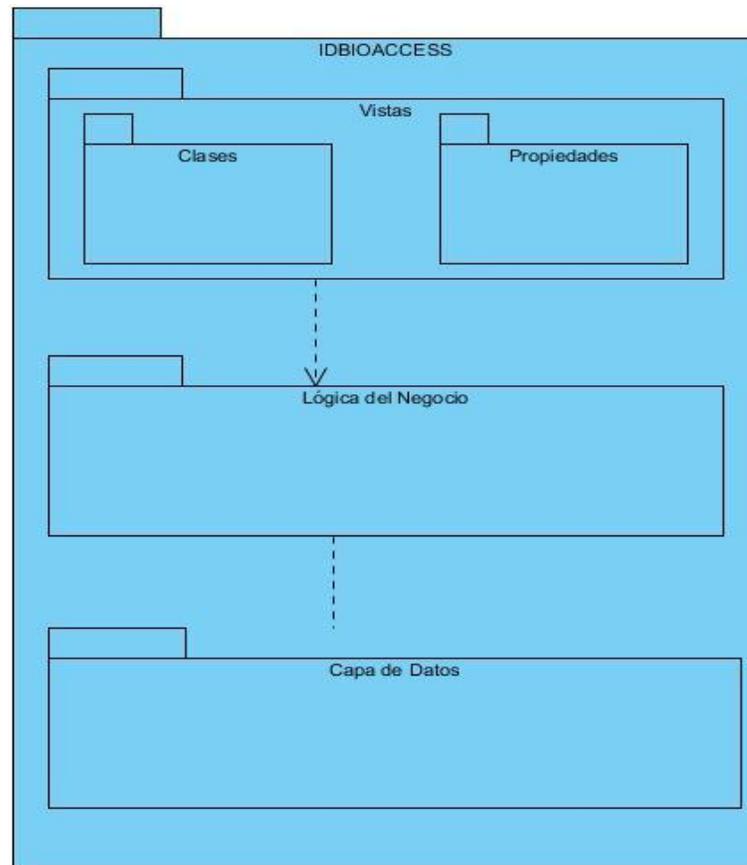


Figura 7: Diagrama de Paquetes del sistema.

2.6.4. Diagrama de Clases

Los diagramas de clases son diagramas estáticos que exponen las diferentes clases que componen un sistema y cómo se relacionan entre ellas. Sirven para mostrar tanto lo que el sistema puede hacer (análisis), como de qué manera puede ser construido (diseño). Son la base de los diagramas de componentes y despliegue. Permiten no solo visualizar, especificar y documentar modelos estructurales, sino también construir sistemas ejecutables, aplicando ingeniería directa e inversa. (Pressman, 5ta Edición). El lenguaje utilizado para especificar una clase de diseño es el mismo que el lenguaje de programación. En la figura 8 se muestra el diagrama de clases donde se observa como a través de la clase controladora *NegocioIdentificacion* implementa la clase interfaz *INegocioIdentificacion* para lo cual hace una instancia a *RepositoryFactory* que contiene todas las demás clases que se necesitan para la ejecución del programa.

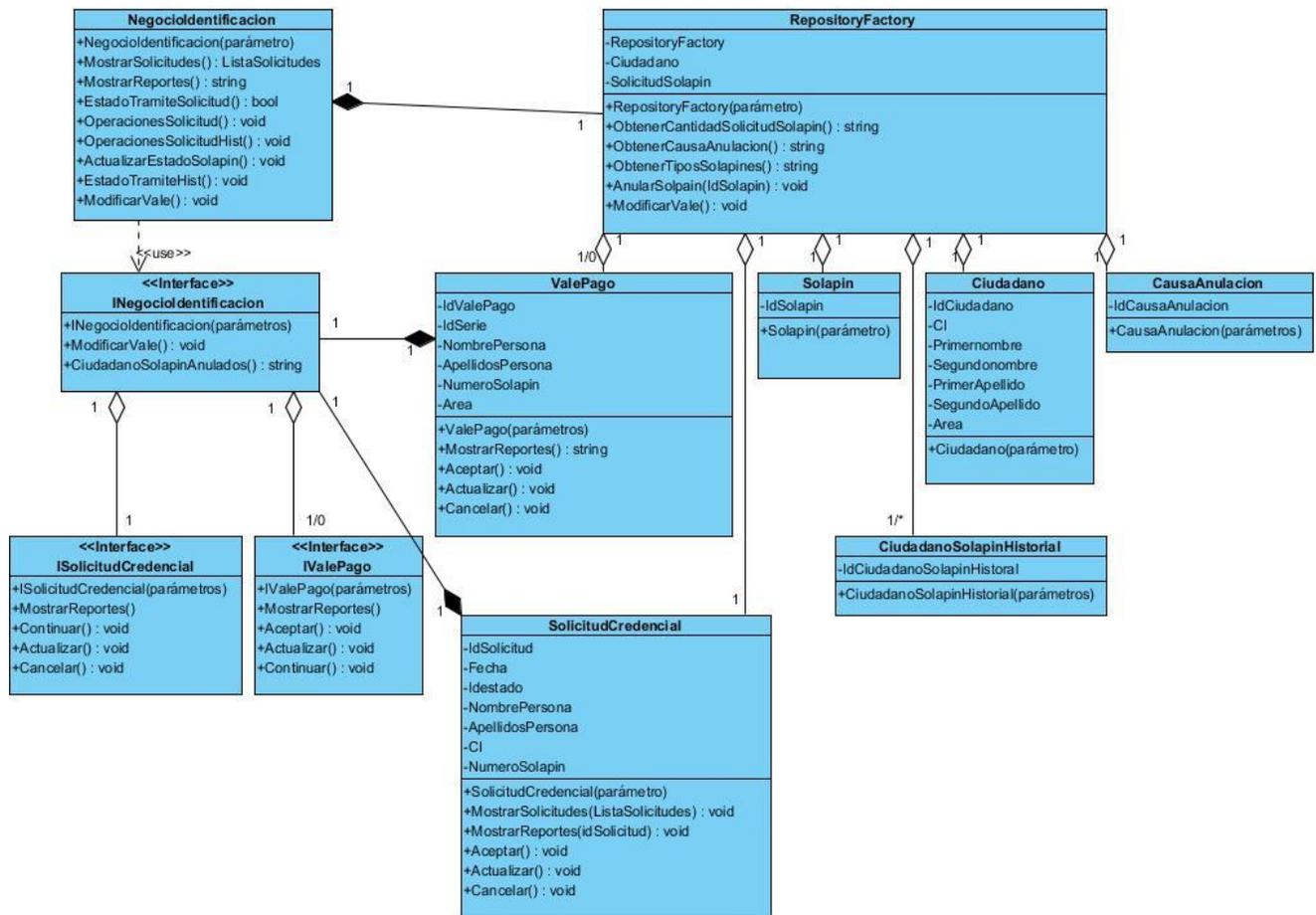


Figura 8: Diagrama de clases del diseño.

2.6.5. Modelo de datos

Como parte de un correcto diseño, es necesario definir lo más exacto posible un modelo lógico, para que sea menos engorroso llegar a un modelo físico. El modelo de clases persistente o lógico como también se le conoce, es el encargado de mostrar las clases que después de analizado el modelo de dominio se identifican que no son persistente, es decir, no tiene valor en el espacio y en el tiempo. Para el módulo gestión de proceso de solicitud de credenciales se definió un modelo físico de la Base de Datos, para verlo ir al [Anexo 5](#)

Descripción de las tablas de la base de datos

La descripción de una tabla de la base de datos posee su nombre, atributos, tipo y descripción, como se muestra a continuación, pero solo se documentan las más importantes, el resto se puede encontrar en el [Anexo 6](#)

Tabla 4: Descripción de la entidad idciudadano

Nombre de la tabla: idciudadano			
Descripción: Entidad que guarda los datos de un ciudadano.			
Atributo	Tipo de dato	Nulo	Descripción
idciudadano	integer	No	Identificador del ciudadano
primernombre	varchar	No	Primer nombre del ciudadano
segundonombre	varchar	Sí	Segundo nombre del ciudadano
primerapellido	varchar	No	Primer apellido del ciudadano
segundoapellido	varchar	Sí	Segundo apellido del ciudadano
area	varchar	Sí	Identificador del área
roluniversitario	varchar	Sí	Identificador del rol
carneidentidad	varchar	No	Número del carné de identidad del ciudadano

2.7. Conclusiones parciales

Se expuso el diseño del módulo para la gestión de solicitud de credencial. Con la utilización de la arquitectura en capas se garantizó que al realizar cambios en una de las capas, las demás no se verían afectadas, pues el flujo de datos es en una dirección. Al realizar el modelo de procesos del negocio, permitió describir detalladamente los procesos que intervienen en el Sistema de gestión de solicitud de credencial, posibilitando definir las entradas y salidas de todas las actividades.

Al definirse los requisitos que constituyen la base para la implementación del sistema, se realizaron los prototipos no funcionales de interfaz de usuario el cual visualiza la estructura de la futura aplicación. Se definió como la arquitectura cliente servidor y se generaron los artefactos diagrama de clases del diseño, el modelo de datos y el diagrama de paquetes del diseño que permitirán una mayor comprensión de la propuesta de solución. Fueron abordados los patrones de diseño empleados para proveer a la solución una imagen de profesionalidad y calidad.

Capítulo 3: Implementación y Prueba

3.1. Introducción

La implementación de un software es la fase que más esfuerzo requiere, en esta fase es donde se le da cumplimiento a cada uno de las funcionalidades que la aplicación debe cumplir teniendo bien definida la propuesta de solución, con los requisitos y descripciones hechas por el analista y la elaboración de los demás artefactos generados. La solución a desarrollar, debe ser fruto de un correcto diseño y construcción de las funcionalidades, las mismas se realizan de forma iterativa facilitando organizar el progreso del sistema e ir obteniendo pequeños fragmentos del producto. Durante la construcción se ejecuta la implementación y las pruebas de caja negra de las funcionalidades, teniendo estas listas para integrarse al sistema.

3.2. Diagrama de componentes

El diagrama de componente describe cómo queda estructurado el sistema mediante elementos físicos y sus relaciones entre componentes de software, estos pueden ser de código fuente, binarios, ejecutables entre otros. Los componentes se agrupan en paquetes según la estructura lógica del sistema que mediante capas bien definidas facilita la implementación del sistema.

En la figura 9 se muestra el diagrama de componentes correspondiente al módulo de Gestión de solicitud de credenciales, el cual se modeló teniendo en cuenta las distintas capas del sistema. Estas capas se agruparon en paquetes que contienen los distintos componentes que se utilizan para el correcto funcionamiento del sistema.

En la capa de presentación se encuentran las interfaces con las cuales interactúa el usuario. En la capa de Negocio se observan las clases encargadas de comunicarse con la capa de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. En esta capa se encuentra el ORM (NHibernate), herramienta que tiene como objetivo principal mapear los objetos desde una aplicación .Net a una base de datos relacional y en la capa de datos es donde residen todos los datos y es la encargada de acceder a los mismos.

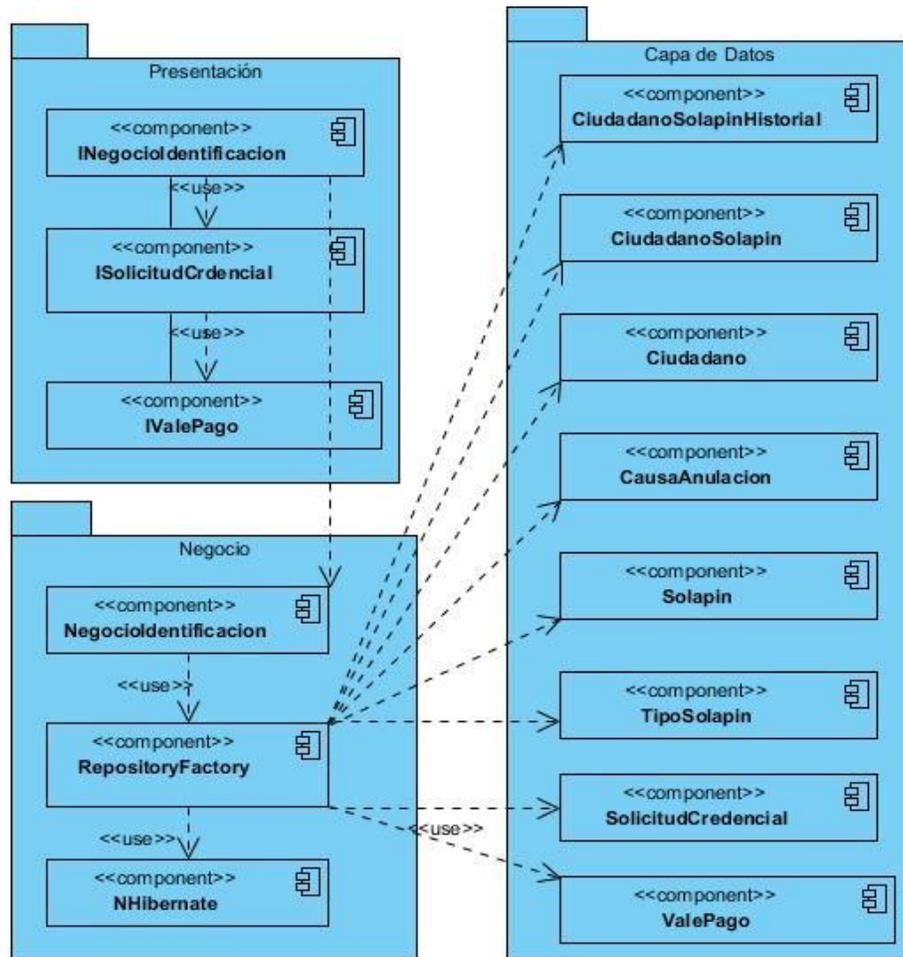


Figura 9: Diagrama de componentes

3.3. Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria.

A continuación se muestra el diagrama de despliegue para la propuesta de solución.

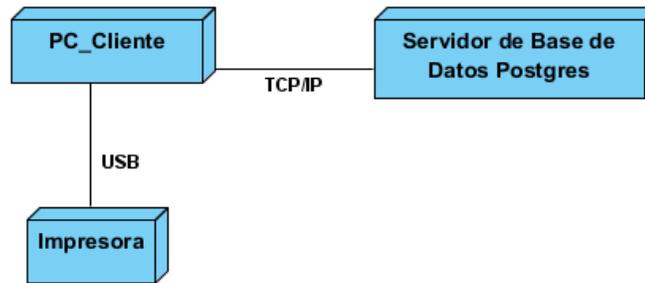


Figura 10: Diagrama de Despliegue.

- Descripción de los nodos y protocolos de comunicación entre ellos.

PC_Cliente: Representa una computadora desde la cual el usuario podrá acceder a la aplicación.

Servidor de Bases de Datos Postgre: Representa el servidor donde estará el SGBD PostgreSQL que dará respuesta a las peticiones hechas por la aplicación.

Impresora: los informes que se generan en el sistema podrán ser llevados a formato duro mediante una impresora conectada a las PC clientes.

Protocolos: TCP/IP, se utiliza en la comunicación entre el servidor y la base de datos para realizar operaciones sobre la información de las tablas.

USB: Bus Universal Serial, interfaz para la transmisión en serie de datos y distribución de energía.

3.4. Estándares de codificación

La especificación del lenguaje C# no define un estándar de codificación. Sin embargo, Microsoft utiliza las instrucciones de este tema para desarrollar ejemplos y documentación.

Las convenciones de codificación tienen los objetivos siguientes:

- Crean una apariencia coherente en el código, para que los lectores puedan centrarse en el contenido, no en el diseño.
- Permiten a los lectores comprender el código más rápidamente al hacer suposiciones basadas en la experiencia anterior.
- Facilitan la copia, el cambio y el mantenimiento del código.
- Muestran los procedimientos recomendados de C#.

- Convenciones de nomenclatura

En breves ejemplos que no incluyen directivas using, se utilizan calificaciones de espacio de nombres. Si se sabe que un espacio de nombres se importa en un proyecto de forma predeterminada, no es necesario completar los nombres de ese espacio de nombres. Los nombres completos pueden partirse después de un punto (.) si son demasiado largos para una sola línea.

- Convenciones de diseño

Un buen diseño utiliza un formato que destaque la estructura del código y haga que el código sea más fácil de leer. Las muestras y ejemplos de Microsoft cumplen las convenciones siguientes:

- Se utiliza la configuración del Editor de código predeterminada (sangría automática, sangrías de 4 caracteres, tabulaciones guardadas como espacios).
- Se escribe solo una instrucción por línea.
- Se escribe solo una declaración por línea.
- Si a las líneas de continuación no se les aplica sangría automáticamente, se hace con una tabulación (cuatro espacios).
- Se agrega al menos una línea en blanco entre las definiciones de método y las de propiedad.
- Se utiliza paréntesis para que las cláusulas de una expresión sean evidentes, como se muestra en el código siguiente:

```
3   if ((indexselec != -1) && (indexselec != 0))
4   {
5       _reportViewer.ProcessingMode = ProcessingMode.Local;.
6   }
```

Figura 11: Ejemplo de código con paréntesis.

- Criterios de Calidad

Es de vital importancia que durante el desarrollo se tengan en cuenta permanentemente los siguientes criterios de calidad:

Tabla 5: Criterios de calidad

CRITERIO	OBJETIVO
Facilidad de comunicación	Proporcionar a los usuarios y desarrolladores entradas y salidas fácilmente asimilables.
Descripción	Proporcionar y/o plasmar en el código detalles y explicaciones sobre la implementación realizada.
Simplicidad	La implementación realizada debe hacerse de la forma más comprensible posible.

3.5. Validación de la solución

Probar el software es de vital importancia ya que proporciona un alto grado de confianza y seguridad en el producto, a la vez que permite hacer estimaciones realistas sobre el comportamiento del producto una vez que se encuentre en funcionamiento. Para verificar el correcto funcionamiento del software se realizan pruebas que demuestran si las funcionalidades cumplen con las expectativas del cliente.

Las pruebas son un conjunto de actividades en las cuales un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos, donde los resultados son observados y registrados para dar una evaluación de algún aspecto del sistema o componente que es evaluado y determinar la calidad del mismo. (Magaña, 2011)

Existen varios tipos de pruebas de software, todas destinadas a encontrar la mayor cantidad de defectos para ofrecer un producto de óptima calidad.

Objetivos de las Pruebas:

- Encontrar y documentar los defectos que puedan afectar la calidad del software.
- Validar que el software trabaje como fue diseñado.
- Validar y probar los requisitos que debe cumplir el software.
- Validar que los requisitos fueron implementados correctamente.

Con tales fines, se realizan las pruebas de unidad de caja negra.

3.5.1. Pruebas de Caja negra.

Las pruebas de Caja negra se refieren a las pruebas que se llevan a cabo sobre la interfaz del software y permiten obtener un conjunto de condiciones de entrada que ejercitan completamente todos los requisitos funcionales del programa.

Objetivos: el objetivo de realizar este tipo de prueba al sistema es el de revelar el incorrecto o incompleto funcionamiento de este, así como los errores de interfaz, rendimiento y errores de inicialización y terminación.

Alcance: el proceso de pruebas de Caja negra se va a centrar principalmente en los requisitos funcionales del software para verificar el comportamiento de la unidad observable externamente y la calidad funcional.

Descripción: se llevan a cabo sobre la interfaz del software y son completamente indiferentes al comportamiento interno y a la estructura del programa. Los casos de prueba de Caja negra pretenden demostrar que:

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

La prueba de Caja negra intenta encontrar errores de las siguientes categorías:

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

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

- Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.

- Técnica del Análisis de Valores Límites: esta Técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

El proceso de pruebas a cualquier software se ejecuta a través de iteraciones, donde, en cada nueva iteración deben haberse erradicado los defectos encontrados en la anterior, para probar que al final del proceso, cuando el producto esté listo para ser entregado al cliente, se encuentra libre de la mayor cantidad de errores posible. Para la aplicación de las pruebas de caja negra se desarrollaron los casos de prueba correspondientes a todas las funcionalidades del módulo, estos constituyen un conjunto de condiciones o variables bajo las cuales se determinará si el requisito de una aplicación es parcial o completamente satisfactorio.

- Realización de los casos de prueba para las funcionalidades:

Mostrar solicitudes.

Aceptar solicitud.

Cancelar solicitud.

Imprimir Solicitud.

A continuación se muestra la prueba para estas funcionalidades, las demás pueden verse en el [Anexo 7](#).

Caso prueba: Solicitudes.

Tabla 6: Caso prueba Solicitudes

Escenario	Descripción	Respuesta del sistema	Flujo central
Mostrar Solicitudes	Se intenta cargar a través de un servicio web las solicitudes almacenadas en Base de datos.	El sistema dará la opción de ver una lista con todas las solicitudes recibidas.	1. Seleccionar la opción "Solicitud Credenciales" en el menú principal. 2. Oprimir el botón "Continuar".
Actualizar lista de Solicitudes	Se intenta a través del botón "Actualizar" refrescar y actualizar la lista de solicitudes recibidas.	El sistema debe mostrar las nuevas solicitudes recibidas a continuación de las anteriores.	1. Seleccionar la opción "Actualizar" en el menú mostrado a la derecha y en la parte inferior de la pantalla. 2. Oprimir el botón

			"Continuar".
Cancelar una Solicitud	Se intenta cancelar una de las solicitudes mostrada en la lista.	El sistema debe eliminar la solicitud de la lista y de la base de datos quedándose en la vista anterior y debe mostrar un mensaje "Solicitud cancelada".	1. Seleccionar la opción "Cancelar" en el menú mostrado a la derecha y en la parte inferior de la pantalla.
Imprimir planilla de Solicitud	Se intenta imprimir la planilla de solicitud una vez creada por el sistema.	El sistema debe imprimir la planilla en la opción seleccionada por el cliente y mostrar un mensaje "Datos salvados satisfactoriamente".	1. Seleccionar la opción "Imprimir" en el menú mostrado en la parte superior de la pantalla.

Durante la ejecución de las pruebas algunas no arrojaron los resultados esperados. A continuación se listan algunas no conformidades durante la ejecución de la primera iteración de pruebas:

- Omisión en la acción del botón "Actualizar".
- Omisión en la acción del botón "Continuar".
- Faltas de ortografía en algunos mensajes del sistema (tildes y mayúsculas incorrectas).
- Orden inapropiado de algunos elementos en el diseño de las interface.

La plantilla de no conformidades constituye un registro de los defectos y fallos encontrados en el transcurso de las pruebas, cuyo principal objetivo es verificar en un futuro que estos errores fueron erradicados en posteriores iteraciones. (Guía, 2012) La siguiente tabla muestra un resumen del registro de defectos y dificultades encontradas durante las tres iteraciones de pruebas:

Tabla 7: Resumen de no conformidades

Iteraciones	No conformidades.	
	Criticas	No criticas
1	12	5
2	7	2
3	0	0

- Resultados de las Pruebas de Caja Negra

Las pruebas de caja negra fueron aplicadas sobre una versión estable del producto, realizándose 3 iteraciones de prueba para cada uno de las funcionalidades. Las no conformidades encontradas en la primera y segunda iteración fueron resueltas satisfactoriamente. La tercera iteración de las pruebas arrojó que el sistema funciona correctamente en un 100% de sus funcionalidades.

En la Figura 12 se muestra de forma gráfica los resultados especificados anteriormente.

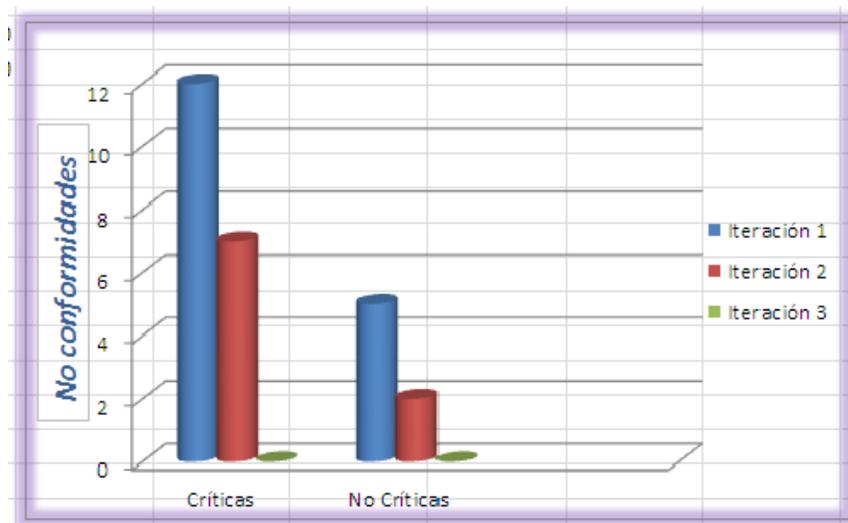


Figura 12: Resultado de las Pruebas de Caja Negra.

3.5.2. Entorno de las pruebas.

Las pruebas de software realizadas se llevaron a cabo en un entorno de pruebas con las siguientes características:

Procesador: AMD C-50 CPU 1.00GHz.

Memoria RAM: 1GB.

Sistema Operativo: Windows 7 R2 con SP2.

Una vez concluido el producto se pudo comprobar que la acción de tramitar solicitudes se muestra solamente a los usuarios que por su responsabilidad o rol dentro del negocio necesitan acceder mostrando en la vista mediante menús el acceso. Además el tiempo de respuesta del sistema a las

peticiones del usuario no excede los 3 segundos quedando así demostrado que cumple con los requerimientos definidos.

3.5.3. Validación de las variables de la investigación

Tabla 8: Comparación de los funcionamientos del sistema

Variable	Antes	Después
Gestión de solicitudes de credenciales de identificación.	El usuario debía esperar por la carta de solicitud impresa, firmada y acuñada por el máximo directivo de su área para presentarla en la Oficina de Identificación.	La solicitud llega por un servicio web con todos los datos necesarios para comenzar el proceso.
Proceso de trámites de solicitud de credenciales.	Todas las planillas de solicitud no llegaban con los datos necesarios por lo que en algunas ocasiones se perdía tiempo de hasta 5 min actualizando los datos y en otros casos había que reenviarla al directivo para su corrección que podía tardar horas. Además hasta que la persona no se presentara personalmente a la oficina, el solapín con dificultad se mantenía activo quedando expuesto a la ilegalidad.	Siempre llega toda la información necesaria, uniforme y completa para comenzar el proceso por lo que no dura más de 1 min tramitar la solicitud. Desde que el directivo envía la solicitud se desactiva el solapín que en esos momentos esté activo evitando que pueda ser utilizado por otra persona.
Control de información sobre el cobro de las credenciales.	Los vales de pago impresos se debían recoger en la dirección de economía, llenarlos a mano y transferir toda la información en un documento Excel lo que traía como consecuencia que el tiempo de atención a una persona duraba de 5 a 7 minutos y en la creación de los reportes estadísticos para los informes mensuales o cualquier información solicitada por los superiores se perdía de 1 a 3 horas, en	Los vales de pago se imprimen en la oficina de Identificación a través del sistema evitando la necesidad de perder tiempo en ir a buscarlos a otra dirección. Además toda la información queda archivada en Base de datos para los reportes estadísticos que ahora se tardan no más de 1 minuto en buscarlos, por lo que el trabajo

	dependencia del tipo de información.	del especialista se agiliza y la información es más real sin correr el riesgo de errores humanos en los cálculos.
--	--------------------------------------	---

3.5.4. Resultados.

Para que un proceso de pruebas tenga éxito se requiere de un análisis final de los resultados obtenidos o una evaluación del producto de acuerdo a todos los defectos y fallos que el sistema presentó a lo largo del proceso. Después de un análisis de los resultados obtenidos, se puede concluir que el proceso de pruebas se llevó a cabo de forma satisfactoria, pues los casos de prueba desarrollados, mostraron tres de los errores del módulo. Estos se pueden clasificar como no adversos, ya que se solucionaron de forma rápida y sin mayores problemas para la segunda iteración. El módulo para la gestión de solicitudes de credenciales ha quedado libre de todos los defectos encontrados en el transcurso de las pruebas y cumple eficientemente cada una de las funcionalidades acordadas con el cliente; por lo que se puede asegurar que cubre las expectativas de este, una vez desplegado en el entorno para el cual fue construido.

3.6. Conclusiones parciales

Se permitió incorporar a la solución una mejor estructura al tener ya definidos los componentes que habrían de ser programados, así como la relación de los nodos físicos que los componen. Además, se definieron los estándares de codificación a utilizar para el desarrollo de la aplicación con el objetivo de lograr una buena organización en el código y un mayor entendimiento de cada sentencia. Las pruebas de caja negra realizadas posibilitaron evaluar la calidad del producto, desde las respuestas de la aplicación ante las peticiones del usuario hasta el comportamiento de las clases y métodos de los procesos. La implementación del módulo cumplió con los requisitos definidos para la solución.

Conclusiones

Con el desarrollo del Módulo para la gestión de solicitudes de credenciales perteneciente al Sistema IDBIOACCESS se ha logrado dar cumplimiento a los objetivos trazados, proporcionando una solución para agilizar el proceso de trámites de solicitud de credenciales de Identificación en la Universidad de las Ciencias Informáticas. A continuación se muestran los resultados obtenidos:

- Se realizó la selección y descripción de las herramientas y metodología con las cuales se elaboraron los diversos diagramas y modelos que recogen la estructura lógica de los componentes utilizados en el desarrollo de la aplicación.
- Al identificarse el estilo arquitectónico en capas, el cual tiene como característica la definición clara de las responsabilidades, permitió reducir el acoplamiento y aumentar la reutilización de las mismas. Los patrones de diseño y los estándares de codificación empleados permitieron garantizar una mayor flexibilidad en la ejecución de la propuesta de solución, trayendo como resultado una implementación de manera organizada.
- Se le realizaron pruebas al sistema, se recolectaron y solucionaron las no conformidades. Con la realización exitosa de las pruebas se demostró que el sistema desarrollado cumple con las funcionalidades definidas por el cliente.
- El módulo desarrollado permite a los especialistas de la oficina de identificación realizar el trámite de solicitud de credenciales de una forma más ágil y efectiva al incluir al sistema la creación y emisión de la planilla de solicitud y el vale de pago para el cobro de las credenciales.
- Se asegura a través de los reportes que es capaz de brindar el sistema, una información detallada referente al cobro de las credenciales.

Recomendaciones

Teniendo en cuenta los resultados obtenidos con el desarrollo del Módulo de Gestión de Solicitud de credenciales para la Oficina de Identificación de la Universidad de las Ciencias Informáticas, se recomienda:

- Crear las condiciones necesarias en Gestión Universitaria para brindar las facilidades al directivo de gestionar las solicitudes de credenciales de identificación.
- Desarrollar un manual de usuario que ayude a interactuar con el Sistema de Gestión de Solicitud de credenciales de Identificación.

Referencias bibliográficas

CISED. 2013. *CISED_PMICA_ID_Manual de Usuario v1.0.* 2013.

Dominguez, L y Rodríguez, A. Modelo lógico y físico de la base de datos correspondiente a los módulos de investigación. [Citado el 31 de octubre del 2014].

Magaña E. 2011. *Verificación y Validación del Sistema.* 2011.

Ferrer, Y y Pérez, Y. 2008. *Análisis y Diseño del Sistema de Acreditación.* 2008 [Citado el 3 de noviembre del 2014].

Gómez, U. 2007. *Sistema de Gestión de Credenciales en la Universidad de las Ciencias.* 2007. [Citado el 5 de noviembre del 2014].

González, D y Valdés, A. 2010. *Módulo de Administración para el Sistema Único de Identificación Nacional de la República de Cuba.* 2010. [Citado el 11 de noviembre del 2014].

Literaria. 2012. *Creación Literaria y Más.* 2012.

Llanes, Y. 2011. *Módulo de administración del Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones de la República Bolivariana de Venezuela.* 2011. [Citado el 20 de noviembre del 2014].

López R, y Rodríguez, Nilberto. 2013. *Sistema de Gestión de Citas para la oficina de trámites de identificación de la Universidad de las Ciencia.* 2013. [Citado el 2 de Diciembre del 2014].

Nicolas, G. 2011. *NHibernate 3.* 2011. [Citado el 5 de Diciembre del 2014]

Pressman. 5ta Edición. *Ingeniería del Software. Un enfoque práctico.* 5ta Edición.

Valencia. 2012. *Un sistema de Gestión de solicitudes e incidencias informáticas.* 2012. [Citado el 20 de noviembre del 2014].

Bibliografía consultada.

CISED. 2013. *CISED_PMICA_ID_Manual de Usuario v1.0.* 2013.

Danny, Ramos Delis Mabel y Hernández Simons. 2013. *Módulos de Captura de datos, imágenes y Supervisión para la Plataforma Modular de Identificación y Control de Acceso.* 2013.

Dominguez, L y Rodríguez, A. Modelo lógico y físico de la base de datos correspondiente a los módulos de investigación. http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_1127_08.

Emilio Magaña. 2011. *Verificación y Validación del Sistema.* 2011.

Ferrer Martínez, Yisell. Pérez Ruio, Yadira. 2008. *Análisis y Diseño del Sistema de Acreditación.* 2008.

Gómez Urquiza. 2007. *Sistema de Gestión de Credenciales en la Universidad de las Ciencias.* 2007.

González, Diana Venero Paez y Alejandro Valdés. 2010. *Módulo de Administración para el Sistema Único de Identificación Nacional de la República de Cuba.* 2010.

Literaria., Creación. 2012. *Creación Literaria y Más.* 2012.

Llanes Embade, Yoana. 2011. *Módulo de administración del Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones de la República Bolivariana de Venezuela.* 2011.

López Rivera, Elizabeth María y Rodríguez Rojas, Nilberto. 2013. *Sistema de Gestión de Citas para la oficina de trámites de identificación de la Universidad de las Ciencia.* 2013.

Nicolas Schenker, Gabrie. 2011. *NHibernate 3.* 2011.

Pressman, Roger S. 5ta Edición. *Ingeniería del Software. Un enfoque práctico.* 5ta Edición.

Universidad Politécnica de Valencia. 2012. *Un sistema de Gestión de solicitudes e incidencias informáticas.* 2012. [En línea].

<http://www.bing.com/search?q=Conferencia%3A+Flujo+de+trabajo+Captura+de+requisitos&qs=n&form=QBRE&pq=conferencia%3A+flujo+de+trabajo+captura+de+requisitos&sc=0-0&sp>

Sistema de gestión de información de estudiantes y trabajadores en las universidades. <http://sigenu.mes.edu.cu:8080/dmmes/pages/info/aboutUsWelcome.face>.

Manual for iReport. [En línea] 2013.

<http://www.bing.com/search?q=Manual+for+iReport&pc=MOZI&form=MOZTSB>

Shores, Redwood. Oracle. [En línea].

<http://www.bing.com/search?q=Shores%2C+Redwood.+&qs=n&form=QBRE&pq=shores%2C+redwood.+&sc=0-0&sp=-1&sk=&cvid=ddc53795507545efb75e6ab5cde28862>

Sommerville 7ma edición. Biblioteca virtual de la Universidad de las Ciencias Informáticas. <http://eva.uci.cu/mod/resource/view.php?id=15881>.

Pressman 6ta edición. Biblioteca virtual de la Universidad de las Ciencias Informáticas. <http://eva.uci.cu/mod/resource/view.php?id=15881>.

García, Yanly Suárez. Diseño de la base de datos para el Grupo de Calidad de la facultad 9. Universidad de las Ciencias Informáticas. Cuba.

Repositorio Institucional de la Universidad de las Ciencias Informáticas. [En línea] 2013. http://repositorio_institucional.uci.cu/jspui

Craig Larman. UML y Patrones. 2da Edición. Biblioteca virtual de la Universidad de las Ciencias Informáticas. <http://eva.uci.cu/mod/resource/view.php?id=15881>.

Conferencia: Flujo de trabajo Captura de requisitos. [En línea]. <http://www.bing.com/search?q=Conferencia%3A+Flujo+de+trabajo+Captura+de+requisitos&qsn=&form=QBRE&pq=conferencia%3A+flujo+de+trabajo+captura+de+requisitos&sc=0-0&sp=-1&sk=&cvid=f0ff7967577941ae9d0aa842de87d084>

Ramiro Machaca Honorio. Scribd.com. [En línea] 2008. <http://www.scribd.com/doc/20966946/1095>

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

Sobre PostgreSQL-es. [En línea]. http://www.postgresql.org/es/sobre_postgresql.

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

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

Pruebas_de_caja_negra. [En línea] <http://www.ecured.cu/index.php/>

ANEXOS.

Anexo 1. Entrevistas.

Persona entrevistada	Rol Universitario	Área o dirección	Pregunta
Daniela Pita González	Estudiante	Facultad 1	¿Cuáles son los tramites que deben realizar cuando se pierde una credencial?
Florangel Román Jiménez	Trabajador.	Dirección de Seguridad y Protección.	¿Qué opina del proceso de solicitud de credenciales en la UCI?
Nayla Socarras Monzón	Especialista.	Oficina de identificación.	¿Cómo es el flujo del proceso de identificación en la oficina? ¿Cómo se tramita actualmente las solicitudes de credenciales? ¿Qué dificultades le crea el trámite de solicitud de credenciales como especialista? ¿Qué dificultades le crea el trámite del cobro de las credenciales y su información? ¿Qué tipos de información referente a las solicitudes de credenciales no obtienen actualmente a través de los reportes? ¿Cómo obtienen los vales para el cobro de las credenciales?
Katia Carmenate Socorro	Trabajador.	Oficina de identificación.	¿Qué dificultades le crea el trámite de solicitud de credenciales como trabajadora? ¿Qué dificultades le crea el trámite del cobro de las credenciales como trabajadora?
Yubisleidy Romero Romaguera	Especialista.	Dirección de informatización.	¿Cómo se pudiera mejorar el trámite de solicitud de credenciales? ¿Qué tipos de información sobre el proceso completo de identificación no genera el sistema?
Alexander Rodríguez Mompíe	Director.	Departamento de desarrollo.	¿Cómo se procesa el trámite de baja por los directivos de la UCI a través del servicio web? ¿Sería posible hacer lo mismo para las solicitudes de credenciales?

Mairelys Boera Velázquez	Especialista.	Departamento de desarrollo.	¿Cómo se ejecuta el flujo de información entre el sistema IDBIOACCESS y el SGU? ¿Cómo crear un nuevo servicio de información del SGU a la base de datos de enrolamiento?
--------------------------	---------------	-----------------------------	---

[Ir atrás](#)

Anexo 2: Requisitos Funcionales

No.	Requisito	Complejidad	Prioridad para el cliente
RF 1	Mostrar solicitudes de credenciales.	Media	Alta
RF 2	Generar planilla de solicitud.	Alta	Alta
RF 3	Aceptar solicitud de credencial.	Media	Media
RF 4	Obtener lista de solicitudes aceptadas.	Media	Media
RF 5	Imprimir solicitud.	Bajo	Bajo
RF 6	Cancelar solicitud.	Media	Media
RF 7	Generar vale de pago.	Alta	Alta
RF 8	Mostrar vale de pago.	Media	Alta
RF 9	Imprimir vale.	Bajo	Alta
RF 10	Anular solapín.	Alta	Alta
RF 11	Actualizar estado de solapín.	Alta	Alta
RF 12	Emitir reportes por tipo de solicitud.	Alta	Alta
RF 13	Emitir reportes de solicitud por fecha.	Alta	Alta
RF 14	Emitir reportes de solicitud por área.	Alta	Alta
RF 15	Emitir reportes por cantidad total de solicitudes.	Alta	Alta
RF 16	Emitir reportes por cantidad total de vales.	Alta	Alta
RF 17	Emitir reportes de vales por rol de la persona	Alta	Alta

RF 18	Emitir reportes de vales por áreas.	Alta	Alta
RF 19	Emitir reportes de vales por fecha.	Alta	Alta

Anexo 3. Descripción de los requisitos

Nº	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF1	Mostrar solicitudes de credenciales.	El requisito permite al especialista mostrar una lista de solicitudes antes de crear la planilla de solicitud. El requisito comienza cuando el técnico o especialista que labora con el sistema después de haberse autenticado, selecciona el módulo de funcionalidades Mostrar solicitudes. Se despliega la lista de solicitudes recibidas y marca la que se vaya a procesar.	Alta	Alta

Funcionalidades Tratadas: RF 3 Aceptar solicitud, RF 4 Obtener lista de solicitudes aceptadas.

Prototipo Mostrar solicitud.

The screenshot displays the 'IDBIOACCESS' web application interface. The header features the XABAL logo and the text 'IDBIOACCESS - Plataforma Modular de Identificación y Control de Acceso'. The user 'Estrella Ramos Pérez' is logged in. The left sidebar contains a menu with options like 'Captación de Imágenes', 'Recapturar Imágenes', 'Impresión de Planilla', 'Supervisión de Datos', 'Asociar Solapín', 'Aprobar Solicitud', 'Entrega de Documentos', 'Activar Solapín', 'Anular Solapín', 'Panel Control', 'Administración', 'Generar Credenciales', 'Reportes', and 'Solicitud Credencial'. The main content area is titled 'Solicitud Credencial' and contains a table with the following data:

Nombre Completo	Solapín	No.
Yunior Cañizares Ramos	E118286	9002
Juan Amaya La fuente	T125710	5706
Juan Alberto Caballero Portelles	T108885	8603
Luis Simón Suárez	E138606	9308

Below the table, it shows 'Total: 4'. At the bottom right, there are buttons for 'Actualizar', 'Continuar', and 'Cancelar'. The version 'Versión:1.02' is visible in the bottom left corner.

	Observaciones	<ol style="list-style-type: none"> 1. Para realizar la acción el usuario tiene que estar autenticado en el sistema. 2. Solo se mostrarán las solicitudes recibidas. 3. Una solicitud solo puede tener asignada una persona. 		
	Poscondiciones	Queda aceptada o denegada la solicitud por el sistema de identificación		
RF2	Generar Planilla de solicitud.	<p>El requisito permite al usuario crear una planilla de solicitud antes de acreditar a la persona.</p> <p>El requisito comienza cuando el técnico o especialista que labora con el sistema después de haberse autenticado, selecciona el módulo de funcionalidades Listar solicitudes. Se despliega la lista de solicitudes recibidas y marca la que se vaya a procesar.</p> <p>Se muestra la planilla con los datos necesarios insertados donde se incluye el tipo de solicitud, nombre y apellidos del directivo que autoriza, nombre y apellidos de la persona que solicita, número de credencial, CI, rol y área. El sistema valida los datos y si todo está correcto muestra la opción "Continuar". Cuando el usuario acepta se guardan los datos en base de datos.</p>	Alta	Alta
Funcionalidades Tratadas: RF 3 Aceptar solicitud, RF 4 Obtener lista de solicitudes aceptadas RF 5 Imprimir solicitud, RF 6 Cancelar solicitud.				
Prototipo Generar Planilla de solicitud.				

IDBIOACCESS XABAL Estrella Ramos Pérez

IDBIOACCESS Plataforma Modular de Identificación y Control de Acceso

Solicitud Credencial

Nombre Completo	Solapín	No.
Yunior Cañizares Ramos	E118286	9002
Juan Amaya La fuente	T125710	5706
Juan Alberto Caballero Portelles	T108885	8603
Luis Simón Suárez	E138606	9308

Total: 4

Vice-rectoría de Tecnología
Dirección de Informatización

A: Oficina de Identificación

Solicitud de revisión de Documento de Identificación por concepto de:

Tipo de pérdida: Deterioro de Solapín

A nombre de: Estrella Ramos Pérez

Área: Facultad 6

Número Solapín	Nombre y Apellidos	CI	Rol
E118286	Yunior Cañizares Ramos	90022741941	Estudiante

Fecha:

Actualizar Continuar Cancelar

Versión: 1.02

Observaciones

1. Para realizar la acción el usuario tiene que estar autenticado en el sistema.
2. Solo se mostrarán las solicitudes recibidas.
3. Una solicitud solo puede tener asignada una persona.

Poscondiciones

Queda aceptada o denegada la solicitud por el Sistema de identificación

Nº	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF 7	Generar Vale de pago	El requisito permite generar un vale para el cobro de la credencial. El requisito comienza cuando el técnico o especialista que labora con el sistema después de haber atendido la solicitud por motivos de pérdida, deterioro baja con pérdida o baja con deterioro de la credencial y comprobar todos los datos referentes a	Alta	Alta

la persona, da un click en "Continuar". Inmediatamente el sistema genera un vale. Se muestra el vale para el cobro de la credencial.

En este vale se muestran los datos: nombre de la entidad, fecha de creado el vale, número de folio que es único y consecutivo, nombre del área o dirección a la que pertenece la persona que solicita, nombre y apellidos, número de solapín, ID expediente y número CI, rol de la persona, tipo de pago (efectivo o descuento) y pie de firma.

El usuario debe especificar el tipo de pago que se va a efectuar marcando una de las dos opciones que se brindan. Se imprime el vale como constancia e inmediatamente el sistema anula la credencia anterior.

Funcionalidades Tratadas: RF 8 Mostrar vale, RF 9 Definir tipo de pago, RF 10 Imprimir vale.

Prototipo generar vale.

The screenshot shows the IDBIOACCESS web application interface. The header includes the XABAL logo, the text "IDBIOACCESS Plataforma Modular de Identificación y Control de Acceso", and the user name "Estrella Ramos Pérez". A sidebar on the left contains a menu with options like "Captación de Imágenes", "Recapturar Imágenes", "Impresión de Planilla", "Supervisión de Datos", "Asociar Solapín", "Aprobar Solicitud", "Entrega de Documentos", "Activar Solapín", "Anular Solapín", "Panel Control", "Administración", "Generar Credenciales", "Reportes", and "Solicitud Credencial".

The main content area displays a form titled "Solicitud Credencial" for "Cobro por Pérdida o Deterioro del Solapín". The form includes the following fields:

- UCI Universidad del Valle de las Ciencias Informáticas
- A: Dirección de Contabilidad y Finanzas
- De: Dirección de Informatización
- Folio: _____ Fecha: _____
- DATOS PERSONALES**

PRIMER NOMBRE Luis	SEGUNDO NOMBRE	PRIMER APELLIDO Simón	SEGUNDO APELLIDO Suárez
CARNET IDENTIDAD 93082618024	NRO SOLAPIN E138806	NRO SERIE EH13823	
AREA Facultad 2	ROL Estudiante	MONTO 40	
- FORMA DE PAGO: Descuento Efectivo
- Firma del Funcionario: _____ Firma del Solicitante: _____

Buttons for "Aceptar" and "Cancelar" are located at the bottom right of the form.

<p>Observaciones</p>	<ol style="list-style-type: none"> 1. Para realizar la acción el usuario tiene que estar autenticado en el sistema. 2. Solo se mostrarán las solicitudes recibidas. 3. Una persona solo puede tener asignado un solapín activo. 4. Cada vez que se imprima un vale se debe anular el solapín activo perteneciente a la persona. 5. Si el usuario presiona el botón Cancelar se muestra el mensaje de confirmación: “¿Está seguro de realizar la acción?”, si presiona Aceptar regresa al listado de solicitudes. Si presiona el botón Cancelar se mantiene en la página sin realizar ninguna operación.
<p>Poscondiciones</p>	<p>Queda impreso o denegado el vale por el Sistema de identificación.</p>

[Ir atrás](#)

Anexo 4. Modelo físico de la Base de Datos



Generated by DatabaseSpy

www.altova.com

[Ir atrás](#)

Anexo 5. Descripción de las tablas de la base de datos

Nombre de la tabla: idciudadanosolapínhist.			
Descripción: Entidad que guarda el historial de solapínes creados			
Atributo	Tipo de dato	Nulo	Descripción
idciudadanosolapínhist	integer	No	Identificador del solapín de un ciudadano
idciudadano	integer	No	Identificador del ciudadano
idsolapín	integer	No	Identificador del solapín
Identificadoranulacion	integer	Si	Identificador del tipo de anulación

Nombre de la tabla: idsolapín.			
Descripción: Entidad que guarda los datos de un solapín			
Atributo	Tipo de dato	Nulo	Descripción
idsolapín	integer	No	Identificador del solapín
numerosolapín	varchar	No	Número del solapín
codigobarra	varchar	No	Número del código de barra
serial	varchar	Si	Número de serie del solapín
estado	char	Si	Estado del solapín
idtipodesolapín	integer	si	Identificador del tipo de solapín

Nombre de la tabla: ncausaanulacion.			
Descripción: Entidad que guarda los datos de un solapín			
Atributo	Tipo de dato	Nulo	Descripción
identificadoranulacion	integer	No	Identificador de la causa de anulación
descripcion	Caracter varying	No	Caracterización de la anulación

Nombre de la tabla: ntiposolapín			
Descripción: Entidad que guarda los datos de un solapín			
Atributo	Tipo de dato	Nulo	Descripción

idtiposolapín	integer	No	Identificador del tipo de solapín
descripcion	Character varying	Si	Caracterización de la anulación
categoria	Character varying	Si	Categoría del solapín

Nombre de la tabla: nsolicitudcredencial			
Descripción: Entidad que guarda los datos de un solapín			
Atributo	Tipo de dato	Nulo	Descripción
idsolicitudcredencial	integer	No	Identificador de la solicitud
descripcion	Character varying	Si	Caracterización de la solicitud

Nombre de la tabla: nvaledepago			
Descripción: Entidad que guarda los datos de un solapín			
Atributo	Tipo de dato	Nulo	Descripción
idvaledepago	integer	No	Identificador del valed de pago
descripcion	Character varying	Si	Caracterización del vale de pago

[Ir atrás](#)

Anexo 6. Pruebas de Caja negra.

Caso prueba: Crear vale de pago.

Escenario	Descripción	Respuesta del sistema	Flujo central
Crear Vale de Pago	Se intenta a través del botón "Continuar" en la vista de Planilla de Solicitud crear el vale de pago.	El sistema creará el vale de pago con los datos de la persona ya insertados.	1. Oprimir el botón "Continuar" En la vista de Planilla de Solicitud.
Imprimir Vale de Pago	Se intenta imprimir el vale de pago para la constancia del proceso.	El sistema debe imprimir el vale en la opción seleccionada por el cliente.	1. Seleccionar la opción "Imprimir" en el menú mostrado en la parte superior de la pantalla.

[Ir atrás](#)