

Universidad de las Ciencias Informáticas

FACULTAD 3



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

Título: Diseño e implementación de una herramienta para la gestión de credenciales en el subsistema capital humano del sistema integral de gestión CedruX.

Autor: Carlos Vingut Guerrero

Tutores: Ing. Joisel Pérez Pérez
Ing. José Antonio Linares Torres

JUNIO 2011



*“Dame seis horas para talar un árbol y yo
pasaré las primeras cuatro afilando el hacha.”*

Abraham Lincoln

Declaramos que somos los únicos autores del trabajo “Diseño e implementación de una herramienta para la gestión de credenciales en el subsistema capital humano del sistema integral de gestión Cedrux.” y autorizamos a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Carlos Vingut Guerrero

Autor

Ing. José Antonio Linares Torres

Tutor

Ing. Joisel Pérez Pérez

Tutor

A mis padres...

Distinguir los individuos unos de otros, reconocer la identidad de una persona, identificarla: estas operaciones que hoy nos parecen banales son el resultado de procesos históricos complejos. Según las épocas, el desarrollo de formas específicas de identificación ha jugado un papel relevante en el ejercicio del poder y en la organización y regulación de las sociedades. La historia de la identificación arroja nueva luz sobre la génesis de la modernidad democrática (1).

A lo largo de la historia las personas comenzaron a sentir necesidad de ser identificados. Hacia los siglos XVIII y XIX, la identificación personal comienza a tomar valor para el mundo, debido a los conflictos armados (1ra y 2da Guerra Mundial) muchas personas perdían sus vidas y quedaban en el anonimato, o eran enterrados como desconocidos, además de la migración que existía hacia algunos países por parte de los civiles, por lo que se hizo necesario crear documentos (tarjetas, placas, etc...) que identificaran a las personas.

Para este proceso se desarrollaron varios métodos en el mundo, uno de ellos fue la identificación mediante el uso de credenciales de identificación personal. A raíz de este método surgieron varias dificultades a la hora de generar las credenciales, en las organizaciones cubanas la actual gestión de las mismas dificulta el proceso del control de asistencia en el sistema integral de gestión de entidades Cedrux, por lo que aquellas instituciones que utilicen este último se verán afectadas. En aras de resolver este problema se desarrolla el diseño e implementación de una herramienta para la gestión de credenciales en el subsistema Capital Humano dentro del sistema integral de gestión de entidades Cedrux.

Palabras Claves

Credenciales, solapines, identificación personal, control de asistencia, control de acceso.

Índice de contenido

Introducción	1
Capítulo 1: Fundamentación teórica.	4
1.1 Introducción	4
1.2 Tecnologías y tendencias actuales para la identificación personal.	4
1.2.1 Credenciales de identificación	4
1.2.2 Tecnologías de hardware utilizadas para la identificación.....	5
1.2.3 Aplicaciones y sistemas informáticos relacionados con la Gestión de Credenciales de identificación personal.....	8
1.2.4 Valoración del estado del arte	10
1.3 Tendencias actuales.....	11
1.3.1 Aplicación Web.....	11
1.3.2 Arquitectura Cliente/Servidor	12
1.3.3 Software libre.....	12
1.3.4 Modelo de desarrollo	13
1.5 Herramientas y Tecnologías	15
1.3.5 Herramientas CASE	15
1.3.6 Servidor WEB.....	16
1.3.7 Control de versiones.....	16
1.3.8 Sistema Gestor de Base de Batos	17
1.3.9 Navegador web Mozilla Firefox.....	18
1.3.10 NetBeans.....	18
1.3.11 Frameworks.....	19
1.3.12 Lenguajes de programación	22
1.3.13 Arquitectura Modelo- Vista- Controlador	23
1.4 Conclusiones del capitulo	24
Capítulo 2: Descripción y análisis de la solución propuesta.	25
2.1 Introducción	25
2.2 Diseño de la solución arquitectónica.....	25
2.2.1 Valoración de la especificación de requisitos.....	25
2.2.2 Principales funcionalidades	26
2.3 Estructura de la solución en términos de componentes	33
2.3.1 Componente Credenciales	¡Error! Marcador no definido.
2.4 Modelo Entidad-Relación.....	30

2.5	Diagrama de clases del Diseño	33
2.5.1	Diagrama de clases del diseño del proceso Gestionar Plantillas	28
2.5.2	Diagrama de clases del diseño del proceso Gestionar Credenciales.....	29
2.6	Patrones de diseño empleados.....	34
2.7	Implementación	36
2.7.1	Estructura física del componente Credenciales dentro del marco de trabajo ¡Error! Marcador no definido.	
2.7.2	Estándares de codificación.....	36
2.8	Conclusiones.....	37
Capítulo 3: Validación de la solución propuesta.....		38
3.1	Introducción.....	38
3.2	Métricas usadas para la evaluación del modelo del diseño propuesto.....	38
3.2.1	Métrica Tamaño operacional de clases (TOC).....	38
3.2.2	Métrica Relaciones entre clases (RC).....	41
3.3	Niveles de pruebas aplicados	45
3.4	Estrategia de pruebas.....	45
3.5	Diseño de casos de prueba para Caja Negra	46
3.6	Conclusiones	48
Conclusiones Generales.....		49
Trabajos citados		51
Bibliografía.....		54
Anexos		57

Índice de Tablas

Tabla 1	Comparativa entre sistemas informáticos de gestión de credenciales.....	11
Tabla 2	Requisitos funcionales del subproceso Gestionar Credenciales.....	26
Tabla 3	Requisitos funcionales del subproceso Gestionar Credenciales.....	26
Tabla 4	Descripción de las entidades del subproceso Gestionar Plantillas	31
Tabla 5	Descripción de las entidades del subproceso Gestionar Credenciales.....	32
Tabla 6	Descripción de las entidades del subproceso Reporte de Impresión.....	33
Tabla 7	Prefijos para los tipos de datos (14).	37
Tabla 8	Métrica Tamaño Operacional de Clase (TOC).	38
Tabla 9	Rango de valores de para la evaluación técnica de los atributos de calidad relacionados con la métrica TOC.	39
Tabla 10	Métrica Relaciones entre Clases (RC).	42

Tabla 11 Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica RC.	42
---	----

Índice de Imágenes

Figura 1 Código 39	7
Figura 2 Diagrama de Estructura de Componentes	34
Figura 3 Entidades del modelo de datos asociadas al subproceso Gestionar Plantillas	30
Figura 4 Entidades del modelo de datos asociadas al subproceso Gestionar Credenciales.....	31
Figura 5 Entidades del modelo de datos asociadas al subproceso reporte de impresión	33
Figura 6 Diagrama de clases del diseño del proceso Gestionar Plantillas.....	28
Figura 7 Diagrama de clases del diseño Gestionar Credenciales	29
Figura 8 Cantidad de procedimientos por clases.	58
Figura 9 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.	39
Figura 10 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.....	40
Figura 11 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación.....	40
Figura 12 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.	41
Figura 13 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.	42
Figura 14 Representación de la incidencia en % de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.	43
Figura 15 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento.....	43
Figura 16 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas.	44
Figura 17 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización.	44

Introducción

Los nombres, apellidos, números de identidad y entre otros elementos son utilizados a diario por las personas para identificarse dentro de la sociedad. La identidad personal es fundamental para las entidades en la gestión del capital humano, pues algunos de los procesos que lo conforman como el Control de Acceso y Control de Asistencia tienen como base la identificación y agrupación de diferentes usuarios¹ por rasgos y características comunes.

En el mundo para llevar a cabo la identificación de los usuarios se utilizaron varios métodos, elementos y herramientas que facilitaron en gran medida la gestión del capital humano, uno de los más conocidos y utilizados por las organizaciones cubanas es el uso de tarjeta de identificación o credencial de identificación.

Las credenciales de identificación suelen contener una serie de componentes que aportan las características y rasgos que diferencian a cada usuario de los demás, dentro de las que generalmente se encuentran se pueden mencionar, imágenes, números y/o textos que se asocian a la identidad del usuario. (2)

La fabricación de estas credenciales de forma manual se hace difícil, debido a la cantidad y variedad que pueden existir en una misma entidad, además del los grandes volúmenes de información que se deben manipular para asociar los datos de los usuarios correspondientes a cada una de ellas.

A pesar de que en el mundo y específicamente en la Universidad de las Ciencias Informáticas se han obtenido logros en este sentido, generando aplicaciones informáticas para la gestión de credenciales que están siendo explotadas por las entidades y organizaciones cubanas, las existentes hoy en día no pueden ser utilizadas por el proceso de Control de Asistencia del subsistema Capital Humano perteneciente al Sistema Integral de Gestión Cedrux, ya que estas están desarrolladas sobre tecnologías que dificultan la integración con Cedrux, además en su gran mayoría están sometidos a licencias privativas que afectan la soberanía tecnológica. A raíz de esto surge como **problema a resolver**: la actual gestión de las credenciales de identificación en las entidades, dificulta el control de asistencia por dispositivos² mediante el sistema integral de gestión Cedrux.

¹ Personas que intervienen en el proceso como trabajadores o clientes.

² Pieza o conjunto de piezas o elementos preparados para realizar una función determinada y que generalmente forman parte de un conjunto más complejo... (28)

Por lo que se plantea como **objeto de estudio** los sistemas informáticos de gestión de credenciales de identificación en el Control de Asistencia dentro de la Gestión del Capital Humano.

De ahí que el **campo de acción** quede enmarcado en el diseño e implementación de los sistemas informáticos de gestión de credenciales de identificación en el Control de Asistencia dentro de la Gestión del Capital Humano.

Por lo que se toma como **idea a defender** el diseño e implementación de una herramienta para la gestión de credenciales en el subsistema Capital Humano del sistema integral de gestión Cedrux, contribuirá a facilitar el control de asistencia por dispositivos en las entidades cubanas que utilicen Cedrux.

Se plantea entonces como **objetivo general** diseñar e implementar una herramienta para la gestión de credenciales que se integre con el subsistema Capital Humano del Sistema Integral de Gestión Cedrux.

Proponiéndose como **objetivos específicos**:

- Fundamentar la investigación mediante la elaboración del Marco Teórico.
- Proponer el diseño e implementación de una solución de software para la gestión de credenciales para desarrollar la aplicación.
- Validar la solución desarrollada para satisfacer las necesidades del cliente cumpliendo con todos los requerimientos.

Para darle cumplimiento a los objetivos que se trazaron se desarrollaron las siguientes **tareas**:

- Estudio de la documentación de la gestión de credenciales y sus funcionalidades en sistemas informáticos a nivel nacional e internacional.
- Estudio de tecnologías y herramientas utilizadas en la gestión de credenciales.
- Realización del diseño de la propuesta de solución.
- Implementación de los requerimientos especificados para la herramienta de gestión de credenciales.
- Realización de pruebas a la solución.

Los **métodos investigativos** utilizados fueron:

- **Métodos empíricos**

- **Experimento:** Para favorecer el desarrollo de las pruebas, con el objetivo de verificar las funcionalidades para comprobar que esté funcionando correctamente y detectar posibles errores.
- **Observación:** Para realizar una descripción detallada.
- **Métodos teóricos**
 - **Analítico – sintético:** Permitiendo procesar toda la información enfocada hacia la investigación, organizar y simplificar el análisis, la teoría, documentos y bibliografía; extrayendo los elementos más importantes relacionados con el diseño y confección de credenciales dentro de la gestión de Capital Humano.
 - **Histórico – lógico:** Para conocer los antecedentes y tendencias actuales en la gestión del Capital Humano en los ERP.

El resultado de la investigación se muestra en el documento de la siguiente forma:

- **Capítulo 1: Fundamentación Teórica:** Se expone el estado del arte donde se muestran módulos de Capital Humano vinculados al campo de acción. Al mismo tiempo se describe el objeto de estudio, se explica el funcionamiento de la herramienta para la gestión de credenciales; también se describen las tendencias y tecnologías actuales seleccionadas para el módulo y el por qué de su utilización.
- **Capítulo 2: Descripción y análisis de la solución propuesta:** Se elaboró una crítica al diseño propuesto por el analista, dado la importancia del mismo para la implementación. Se describe uno de los algoritmos más complejos con los que cuenta la aplicación, explicando además la estructura de datos usada en el mismo, se describen las clases que fueron definidas, que sirven de documentación a los que le den continuidad a la implementación del software y como se realiza la integración entre estas clases.
- **Capítulo 3: Validación de las solución propuesta:** Se valida la solución propuesta a través de la realización de pruebas; el mismo se encuentra dividido en dos partes, la explicación detallada de las pruebas de caja blanca que fueron realizadas al código del software y las pruebas de caja negra que se realizaron a las funcionalidades del mismo.

Capítulo 1: Fundamentación teórica.

1.1 Introducción

En el presente capítulo se estarán abordando una serie de cuestiones referentes a los sistemas de Gestión de Credenciales existentes en la actualidad y las tendencias en este sentido tanto nacional, como internacional, llevando a cabo un minucioso análisis sobre el tema, para identificar los aspectos que deben ser tomados en cuenta para que la herramienta que se pretende desarrollar cuente con las funcionalidades y calidad que se demanda en estos tipos de sistemas informáticos en la actualidad. Se explicará brevemente el modelo de desarrollo elaborado por la dirección del proyecto a partir del cual estará orientado el desarrollo de la solución. Se realizará un análisis de la arquitectura definida para el sistema y se especificarán las tecnologías y herramientas de desarrollo a utilizar durante la implementación, para facilitar el cumplimiento de los requisitos tanto técnicos como funcionales.

1.2 Tecnologías y tendencias actuales para la identificación personal.

La industria del software ha demostrado ser una de las áreas más dinámicas y con mayor crecimiento en los últimos años. La evolución hacia un modelo más racional para los usuarios, con menos costos de licencia y donde se intensifique la prestación de servicios, reduciendo el precio de desarrollo e incrementando la calidad, viene siendo lo más importante a la hora de desarrollar cualquier tipo de aplicación informática. A continuación se ofrecerá una valoración sobre algunas de las tendencias y tecnologías que marcan un alto nivel en el mundo del software. (3)

1.2.1 Credenciales de identificación

Las tarjetas credenciales de identificación plásticas de pvc son una manera segura para evitar falsos representantes de empresas en caso de visitas a otros lugares. Soportan datos variables y fotografías, código de barras o tecnología de proximidad, generalmente utilizadas para distinguir e identificar al personal y cargo dentro de la empresa.

Utilizadas en gran mayoría por trabajadores de contacto directo con el público tales como cajeros, vendedores y anfitriones, entre otros.

Entre sus beneficios podemos mencionar

- Puede cumplir la función de credencial y tarjeta de acceso (si se acompaña de un sistema electrónico de acceso) accionando un dispositivo de control de acceso electrónico, ya sea con código de barras o con proximidad.
- Puede utilizarse para control horario de asistencia.
- Soporta datos de texto e imágenes variables y fijas sin problema.
- Es una manera segura de acreditar a invitados, expositores y organizadores de ferias y eventos.
- Puede utilizarse para acreditación de calidad de socio en clubes y otras organizaciones.
- Puede utilizarse como entrada a eventos, fiestas, conciertos, convenciones, etc. Generalmente nuestros clientes las utilizan para entregar free-pass o entrada preferencial a sus clientes VIP. (4)

1.2.2 Tecnologías de hardware utilizadas para la identificación.

Algunas de las tecnologías de hardware utilizadas por los usuarios para la identificación son:

Claves por Teclado

Realmente esta opción es la más económica, pero la menos segura. Hace tiempo que no es muy utilizada y no se han generado hasta el momento nuevas aplicaciones donde puedan resurgir como una opción válida (5).

Tecnología de Cinta Magnética

Es la tecnología más conocida y difundida, dado que se utiliza en todos los sistemas de tarjeta de crédito y de compra. El lector identifica al usuario leyendo el código inscrito en la cinta magnética que está adherida a la tarjeta. Es más difícil generar los códigos en cintas magnéticas que en códigos de barras, pero a pesar de esto, los lectores y codificadores de datos son ampliamente generalizados y fáciles de conseguir. Sus ventajas son su difusión, popularidad y bajo costo; sin embargo, el hecho de que la banda se raye o que su código pueda borrarse al ser expuestas a campos magnéticos. Son recomendables por ejemplo para usar en ambientes industriales. Sólo se recomiendan en oficinas o establecimientos administrativos. (6)

Tarjetas de Código de Barras

El código de barras es una tecnología de identificación automática. Permite recolectar datos con precisión y rapidez. Un código de barras consiste en una serie de barras paralelas con espacios entre ellas. Los diseños predeterminados de anchura se utilizan para codificar datos en el código. Para leer información en un símbolo de código de barras, un dispositivo de lectura, actúa como un lápiz óptico que se desliza a través del símbolo de un lado al otro, la anchura de barras y los espacios son analizados por el decodificador del lector para finalmente obtener los datos originales se recuperan. La aplicación más visible en esta tecnología es en la industria de supermercado, donde ha estado en uso desde 1970 (5).

Touch Memories

Se trata de una pastilla electrónica, encapsulada en acero inoxidable de unos 16 mm, tan práctica que puede ser colgada en un llavero.

Se les denomina comúnmente llave electrónica y brindan un alto nivel de seguridad, ya que son altamente resistentes al desgaste, siendo ideales para ambientes industriales, no así para ambientes con alto grado de generación de corriente estática (Ej: oficinas con mucha alfombra y ambientes muy secos). Su tecnología de avanzada evita la posibilidad de duplicarlas. Por su costo es de destacar que son unos de los medios más caros, sin embargo, nunca se desgastan y son muy confiables (5).

Tarjetas de Proximidad o Radio Frecuencia (RF)

Estas tarjetas poseen un diseño tecnológico que las hace prácticamente imposibles de duplicar, lo cual las convierte en una de las tecnologías más modernas y efectivas. Son muy prácticas y de bajo costo de mantenimiento, idóneas en situaciones de máxima seguridad y alta tecnología, incluso pueden ser leídas dentro de una billetera, una cartera, etc.

Estas tarjetas son llamadas también de Radio Frecuencia (RF) y estos sistemas constan de dos partes; la unidad lectora y el tag, transponder o tarjeta. Se denomina proximidad activa cuando las tarjetas tienen alimentación interna, y proximidad pasiva cuando no tienen esa batería interna, esta última es la más usada, el lector emite una onda electromagnética que genera una corriente interna en la tarjeta, proporcionando su alimentación, dentro de ésta existe un microchip que tiene una memoria que guarda el código y una antena que es la encargada de recibir y transmitir, al inducirse esa corriente en la tarjeta, extrae de la memoria el dato codificado y lo devuelve,

enviándolo al lector, el mismo lo recibe, lo decodifica, lo filtra, lo amplía y lo envía a la unidad de control de accesos a la que esté conectada (5).

Sistemas Biométricos

Su funcionamiento está basado en la lectura o reconocimiento de alguna parte del cuerpo humano; de la huella dactilar, geometría de la mano, frecuencia de la voz, por la retina o reconocimiento facial; eliminando por completo el uso de las tarjetas.

Los más conocidos son los lectores de huellas dactilares, geometría de la mano e iris del ojo. Sus desventajas son la velocidad y el precio, además, deben ir acompañados de un teclado (para anteponer un código y acelerar el proceso de búsqueda), además de la poca posibilidad de ser autónomos (generalmente por su complicada lógica se ven obligados a trabajar con un software de análisis y una PC conectada directa al lector, lo cual es poco práctico y más caro), pero seguramente con el tiempo se irán superando estas dificultades y en un futuro se convertirán en una mejor opción en el mercado (5).

Tecnología utilizada en la UCI: El código de barras

La tecnología de código de barras es muy económica, y de fácil implementación, ya que con una impresora de calidad pueden ser impresos los carnés de identificación con códigos de barras. Existen varios tipos de codificación, actualmente, los más usados son:

- Código 39
- Código 39 ASCII Total
- Codabar
- Intercalado 2 de 5
- Código 128
- UPC (Código Universal de Producto)
- EAN (European Article Numbering o Sistema de Numeración Europeo)

En la UCI se utiliza el **Código 39**, que es el más utilizado porque codifica números, letras mayúsculas, y algunas marcas de puntuación (mayúsculas A-Z, Números 0-9, "espacios" y símbolos: -, +, /, \$, . , %). El Código 39 puede ser variable en longitud, permitiendo hacer códigos de cualquier cantidad de dígitos. Este formato ha pasado a ser la norma para el gobierno, la industria, la educación y las aplicaciones de negocios (5).



Figura 1 Código 39

1.2.3 Aplicaciones y sistemas informáticos relacionados con la Gestión de Credenciales de identificación personal.

A continuación se hace referencia a algunas de las herramientas utilizadas para la gestión de credenciales tanto en el ámbito nacional como en el internacional, mencionando sus características principales y las facilidades que nos brindan.

1.2.3.1 AlphaCard

El sistema AlphaCard es propiedad de la empresa Jolly inc. (Reg.C2226540 California, EEUU), está especializado en la gestión de credenciales de identificación, facilitando una serie de herramientas y posibilidades que la hace una potente herramienta en esta área, dentro de sus propiedades fundamentales se encuentran:

- Compatible con todas las impresoras de PVC³.
- Compatible solo con los sistemas operativos Windows 2000, XP, Server 2003, Vista, y Windows 7.
- Es una aplicación de tipo Escritorio (Desktop).
- Permite importar datos de base de datos existentes para ser utilizados en la confección de las credenciales.
- Dentro de su amplia gama de herramientas se encuentran campos para, textos, imágenes, códigos de barra, cintas magnéticas, chips inteligentes y otras.
- Los archivos son almacenados en archivos con extensión “.jsp” de manera local, por lo que se dificulta su uso de forma online. (7)

El uso del AlphaCard en las organizaciones cubanas, no permite la soberanía tecnológica, genera gastos ya que su licencia es privativa. Además dificulta el acoplamiento con el sistema integral de gestión Cedrux, dificultando procesos como el Control de Asistencia dentro de las organizaciones cubanas que utilicen Cedrux. También tiene como desventaja la forma en la que almacena sus plantillas, la cual hace más difícil la reutilización de las mismas de forma on-line.

1.2.3.2 Advanced ID Creator

Advanced ID Creator es una potente herramienta para la gestión de credenciales, es propiedad de XPRESS Software. Alguna de sus propiedades más significativas son:

³ Un tipo de polímero (Plástico) que se obtiene como derivado del petróleo, y es utilizado para crear todo tipo de objetos, como por ejemplo: tuberías, tarjetas de identificación, tarjetas de cuentas bancarias, entre otras cosas.

- Es una aplicación de tipo Escritorio(Desktop).
- Trabaja sobre el sistema operativo Windows.
- Cuenta con la posibilidad de usar herramientas como código de barra, cinta magnética, chips inteligentes, líneas, figuras geométricas, textos y otras.
- Cuenta con un sistema de usuarios para garantizar la seguridad de los diseños, el cual da la posibilidad de autenticarse al entrar al sistema con varios usuarios, gestionando los privilegios entre los mismos.
- Aunque posee una base de datos local, da la posibilidad de importar alguna existente de tipo Microsoft SQL, Microsoft Access, Microsoft Excel y CSV(Comma, SeparatedValues).
- Permite imprimir credenciales de forma independiente o en grupos. (8)

Funciona solo sobre el sistema operativo Windows, su utilización por parte de las organizaciones cubanas generarían gastos en patentes y soporte técnico, no es compatible con bases de datos montadas en Postgres u Oracle, por mencionar algunas de las más importantes, y dificulta la gestión de credenciales para las organizaciones que utilicen CedruX, ya que no cumple con los requisitos para acoplarse a este último.

1.2.3.3 CardFive

Es una aplicación potente y fácil de usar, con características que se necesitan para generar tarjetas, con total integración y versatilidad en la interfaz. Dentro de sus propiedades más significativas se encuentran:

- Es una aplicación de tipo Escritorio(Desktop).
- Trabaja sobre el sistema operativo Windows.
- Se integra a bases de datos existentes.
- Este programa brinda la posibilidad de codificar bandas magnéticas, procesar tarjetas inteligentes (smartcards), crear efectos y controlar el manejo de marcas de agua, hologramas y otras (9).

Al igual que los mencionados anteriormente funciona solo sobre el sistema operativo Windows y no es compatible con el sistema integral de gestión CedruX, por lo que dificulta su uso por las organizaciones cubanas que utilicen CedruX.

1.2.3.4 DataCard Id Work Id

El software de identificación de ID Works permite mejorar cada aspecto del diseño de tarjeta y reportes, así como la producción de estos. Ofrece gran poder y es extremadamente fácil de usar. Además de un diseño flexible y modular, permite

seleccionar solamente los componentes específicos que se necesitan. DataCard es una compañía con más de 30 años de experiencia en este tipo de herramientas.

- Es una aplicación Desktop.
- Trabaja sobre el sistema operativo Windows. (10)

A pesar del cumulo de funcionalidades que brinda y la calidad del servicio, no cumple con las características necesarias para acoplarse al sistema integral de gestión de entidades Cedrux, es un sistema Desktop y funciona sobre el sistema operativo Windows. Todas estas características descartan su posible uso por las organizaciones cubanas para la gestión de credenciales de identificación.

1.2.4 Valoración del estado del arte

Después de haber realizado el estudio de los sistemas y herramientas utilizados para la gestión de credenciales teniendo en cuenta no solo aspectos fundamentales desde el punto de vista del software, sino también del producto, se pone de manifiesto que ninguno de los software estudiados cumplen con los requisitos de forma cabal, atendiendo a lo que se demanda por parte de la gestión del Capital Humano en el Sistema Integral de Gestión Cedrux. Ya sea por las herramientas sobre las que están soportadas, por las licencias de dichos productos o bien porque no son capaces de cumplir con los requisitos establecidos a nivel nacional, sería entonces factible que el proyecto y el país en el proceso de lograr su soberanía tecnológica trabajaran sobre herramientas y tecnologías que no fueran propietarias, siendo una aplicación web y el trabajo con PHP lo más indicado para desarrollar un sistema de este tipo con el fin de lograr un alto grado de eficiencia, confiabilidad y rapidez en la gestión contable de las empresas.

A continuación se muestra una tabla comparativa entre las herramientas antes mencionadas y la que se propone desarrollar (ver **Tabla 1**).

	CardFive	ID Works	AlphaCard	AID Creator	CEDRUX
Software Libre					
Multiplataforma					
Independencia tecnológica					
Obtención de datos a partir de SBD					

Cumple los requerimientos del ERP					
Gestión de código de barra					
Gestión de cinta magnética					

Tabla 1 Comparativa entre sistemas informáticos de gestión de credenciales

1.3 Tendencias actuales.

La industria del software ha mostrado ser una de las áreas más dinámicas y con mayor crecimiento en los últimos años. La evolución hacia un modelo más racional para los usuarios, con menos costos de licencia, donde se intensifique la prestación de servicios, se reduzca el tiempo de desarrollo e incremente la calidad, viene siendo lo más importante a la hora de desarrollar cualquier tipo de aplicación. A continuación se ofrecerá una valoración sobre algunas de las tendencias y tecnologías que marcan un nivel alto en el mundo del software y que bien contribuyen a lo dicho anteriormente.

1.3.1 Aplicación Web

La creación de aplicaciones de este tipo ha tomado gran auge debido a las funcionalidades que ofrece; algunas de estas se presentarán a continuación:

- **Compatibilidad Multiplataforma:** Una misma versión de la aplicación puede correr sin problemas en múltiples plataformas como Windows y Linux.
- **Menos requerimientos de hardware:** Este tipo de aplicación no consume (o consume muy poco) espacio en disco y también es mínimo el consumo de memoria RAM en comparación con los programas instalados localmente. Tampoco es necesario disponer de computadoras con poderosos procesadores ya que la mayor parte del trabajo se realiza en el servidor donde reside la aplicación.
- **Actualización:** Una de las características fundamentales de las aplicaciones web es que siempre se mantienen actualizadas y no requieren que el usuario deba descargar actualizaciones ni realizar tareas de instalación.
- **Acceso inmediato y desde cualquier lugar:** Las aplicaciones basadas en tecnología web no necesitan ser descargadas, instaladas o configuradas. Además, pueden ser accedidas desde cualquier computadora conectada a la red.

- Seguridad en los datos: Los datos se alojan en servidores con sistemas de almacenamiento altamente fiables y se ven libres de problemas que comúnmente sufren los ordenadores de usuarios comunes como virus y roturas de disco.

Cuando se desarrolla una aplicación web existe una tendencia al uso de Arquitectura en capas y del patrón de diseño: Modelo-Vista-Controlador (MVC). (3)

1.3.2 Arquitectura Cliente/Servidor

La arquitectura cliente/servidor se basa en un sistema distribuido entre múltiples procesadores donde hay clientes que solicitan servicios y servidores que los proporcionan. La mayoría de las aplicaciones que se están desarrollando actualmente en la industria de software utilizan este tipo de arquitectura debido a las funcionalidades que brindan, entre ellas:

- Permite integrar y compartir información entre sistemas diferentes sin necesidad de que todos tengan que utilizar el mismo sistema operativo.
- Posibilita el mantenimiento y el desarrollo rápido de aplicaciones.

La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones. (3)

1.3.3 Software libre

El software libre permite crear soluciones cada vez más sofisticadas y más personalizadas debido a la libertad de los usuarios para ejecutar, copiar, distribuir y mejorar el software, tiene entre sus ventajas fundamentales:

- Independencia tecnológica: El acceso al código fuente permite el desarrollo de nuevos productos sin la necesidad de desarrollar todo el proceso partiendo de cero.
- Libertad de uso y redistribución: Las licencias de software libre existentes permiten la instalación del software tantas veces y en tantas máquinas como el usuario desee.

De ahí la repercusión que tiene en el ámbito informático hoy día, siendo una tendencia que va madurando cada vez más.

1.3.4 Modelo de desarrollo

La propuesta de modelo de desarrollo que a continuación aparece fue elaborada por el equipo de producción en colaboración con las Líneas de desarrollo del proyecto ERP-Cuba de acuerdo con las necesidades presentadas por cada una de ellas y donde se tuvieron en cuenta los principales riesgos con los que se cuentan en el proyecto (11).

1.3.4.1 Características

- **Centrado en la arquitectura**

La arquitectura determina la línea base y los elementos de software estructurales a partir de los elementos de la arquitectura de negocio. Interviene en la gestión de cambios y diseña la evolución e integración del producto. La arquitectura orienta las prioridades en la producción y resuelve las necesidades tecnológicas y de soporte para el desarrollo.

- **Orientado a componentes**

Las iteraciones son orientadas según la importancia arquitectónica de los componentes, los mismos son abstracciones arquitectónicas de los procesos de negocio y requisitos asociados que modelan, el componente es la unidad de medición y orden de las iteraciones.

- **Iterativo e incremental**

Las iteraciones son planificadas y coordinadas con el equipo de arquitectura, los clientes y la alta gerencia. Cada iteración constituye el desarrollo de componentes, los cuales son integrados al término de la iteración, permitiendo de esta manera la evolución incremental del producto.

- **Ágil y adaptable al cambio**

El desarrollo de las partes precisa solamente las características principales de la solución, priorizando los talleres y las comunicaciones entre las personas. Los clientes y funcionales están involucrados en el proyecto y poseen parte de la responsabilidad del

éxito del mismo. Los cambios son conciliados semanalmente, discutidos y aprobados.

1.3.4.2 Fases

- **Concepción**

Los objetivos de esta fase son:

- ◆ La evaluación de factibilidad del proyecto y con ello la elaboración y aprobación del proyecto técnico, definiendo los macro requisitos y macro componentes.
- ◆ Definir los procesos de negocios e identificar los requisitos. Definir la línea base de los requisitos del proyecto.

El hito de la fase Concepción es concebir el alcance del proyecto.

- **Elaboración**

Los objetivos de esta fase son:

- ◆ La firma del Acuerdo de Colaboración o Contrato.
- ◆ Se describen los requisitos y se realiza el diseño arquitectónico de la solución.

El hito de esta fase es establecer la línea base de la arquitectura.

- **Construcción**

El objetivo de esta fase es:

- ◆ Implementar las funcionalidades del sistema informático.

El hito de esta fase son las pruebas internas y de liberación por parte del Laboratorio Industrial de Pruebas de Software del sistema informático.

- **Cierre**

Los objetivos de esta fase son:

- ◆ Realizarle las pruebas piloto y de regresión al software.
- ◆ Garantizar la transferencia del sistema al cliente.
- ◆ Mantenimiento al software.

El hito de esta fase es la transferencia de una versión estable junto a su documentación al cliente del sistema informático.

1.5 Herramientas y Tecnologías

1.3.5 Herramientas CASE

Las Herramientas de Ingeniería de Software Asistida por Computadoras (Computer Aided Software Engineering, CASE por sus siglas en inglés) son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y dinero. Se usan para fines como, realizar el diseño del proyecto, cálculo de costo, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación y detección de errores. (12)

Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad a un menor costo. Permite crear diagramas de clases, generar código desde diagramas y documentación.

Visual Paradigm se integra al Entorno de Desarrollo Integrado (Integrated Development Environment, IDE por sus siglas en inglés) de Eclipse. Está diseñado para desarrollar software con Programación Orientada a Objetos, reduce la duración del ciclo de desarrollo brindando ayuda tanto a arquitectos, analistas, diseñadores como a desarrolladores. Busca también automatizar tareas tediosas que pueden distraer a los desarrolladores. Dentro de sus características fundamentales están (13):

- Multiplataforma
- Interoperabilidad
- Modelamiento de los Requisitos
- Colaboración de Equipo
- Generación de Documentación
- Editor de Detalles de Casos de Uso
- Ingeniería de Código
- Modelado de Procesos de Negocio

- Integración con Entornos de Desarrollo
- Modelamiento de Bases de Datos

1.3.6 Servidor WEB

Apache

Apache, es un servidor de protocolo para la transferencia de hipertextos (Hypertext Transfer Protocol, HTTP por sus siglas en inglés) de software libre para plataformas Unix, Windows, y Macintosh, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo (14).

Dentro de sus características fundamentales se encuentran:

- Multiplataforma
- Es un servidor web conforme al protocolo HTTP/1.1
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Basado en hilos de ejecución en la versión 2.0.
- Incentiva la retroalimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones para trabajar con otros lenguajes como PHP, un lenguaje de programación del lado del servidor.

1.3.7 Control de versiones

Una versión, revisión o edición de un producto, es el estado en el que se encuentra en un momento dado en su desarrollo o modificación. Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas. Un sistema de control de versiones debe proporcionar un mecanismo de almacenaje de los elementos que

deba gestionar y un registro histórico de las acciones realizadas con cada elemento o conjunto de elementos (normalmente brindando la posibilidad de volver o extraer un estado anterior del producto) entre otros aspectos. Todos los sistemas de control de versiones se basan en disponer de un repositorio, que es el conjunto de información gestionada por el sistema. Este repositorio contiene el historial de versiones de todos los elementos gestionados. Cada uno de los usuarios puede crearse una copia local duplicando el contenido del repositorio para permitir su uso. Es posible duplicar la última versión o cualquier versión almacenada en el historial.

Subversion

También conocido como SVN, es un sistema de control de versiones que se ha popularizado bastante, en especial dentro de la comunidad de desarrolladores de software libre. Está preparado para funcionar en red y se distribuye bajo licencia libre (15).

- Mantiene versiones no sólo de archivos, sino también de directorios.
- Mantiene versiones de los metadatos asociados a los directorios.
- Además de los cambios en el contenido de los documentos, se mantiene la historia de todas las operaciones de cada elemento, incluyendo la copia, cambio de directorio o de nombre.
- Atomicidad de las actualizaciones, una lista de cambios constituye una única transacción o actualización del repositorio, esta característica minimiza el riesgo de que aparezcan inconsistencias entre distintas partes del repositorio.
- Soporte tanto de ficheros de texto como de binarios.
- Mejor uso del ancho de banda, ya que en las transacciones se transmiten sólo las diferencias y no los archivos completos.

1.3.8 Sistema Gestor de Base de Datos

Se denomina Sistema Gestor de Base de Datos (siglas: SGBD) al conjunto de programas que permiten definir, construir y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad.

PostgreSQL

PostgreSQL es un servidor de base de datos relacional, libre. Se destaca en ejecutar consultas complejas, consultas sobre vistas, subconsultas y e intersecciones de gran tamaño. Permite la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. Como toda herramienta de software libre PostgreSQL tiene

entre otras ventajas las de contar con una gran comunidad de desarrollo en Internet, su código fuente está disponible sin costo alguno y algo muy importante es que es multiplataforma. Fue diseñado para ambientes de alto volumen. Escala muy bien al aumentar el número de CPUs y la cantidad de RAM. Soporta transacciones y desde la versión 7.0, claves ajenas con comprobaciones de integridad referencial. Tiene mejor soporte para vistas procedimientos almacenados en el servidor, transacciones, almacenamiento de objetos de gran tamaño y además tiene ciertas características orientadas a objetos.

Se estará haciendo uso de PostgreSQL en su versión 8.3 ó superior e inferior a 8.4

1.3.9 Navegador web Mozilla Firefox

Mozilla Firefox es un navegador de Internet, libre y de código abierto. Es usado para visualizar páginas web. Incluye corrector ortográfico, búsqueda progresiva, marcadores dinámicos y un sistema de búsqueda integrado que utiliza el motor de búsqueda que desee el usuario. Además se pueden añadir funciones a través de complementos desarrollados por terceros. Dentro de sus características fundamentales se encuentran:

- Multiplataforma.
- Cuenta con una protección antimalware e integración con el antivirus.
- navegación por pestañas.
- Bloqueador de ventanas emergentes.
- Múltiples Extensiones.
- Incluye un buscador integrado en la interfaz que hace búsquedas en Google.
- Posee gestor de descargas.
- Utiliza el sistema SSL para proteger la comunicación con los servidores web, utilizando fuerte criptografía cuando se utiliza el protocolo HTTPS.

1.3.10 NetBeans

NetBeans es un IDE, una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación como C, C++, C#, PHP, Ruby, Python, HTML, JavaScript, CSS y otros.

Además NetBeans gestiona la complejidad de SOA e integra soporte de modelado en UML incluyendo soporte bidireccional que permite implementar y sincronizar rápidamente los modelos con los cambios en el código a medida que la aplicación avanza por los ciclos de desarrollo.

Existe además un número importante de módulos para extender el IDE NetBeans y es un producto libre y gratuito sin restricciones de uso.

Ha sido desarrollado para distintas plataformas como Linux, MacOS X, Solaris y también Windows.

Integra además:

- Symfony
- Zend Framework.
- PHP UnitTesting.
- Depuración de PHP.
- MySQL.
- Sistemas de Control de Versiones (SVN, CVS, Mercurial y Git).

1.3.11 Frameworks

Un framework, en el desarrollo de software, es una estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, 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.

A continuación se brinda una breve explicación del marco de trabajo utilizado por el sistema integral de gestión de entidades Cedrux, sobre el cual se desarrolla la herramienta para la gestión de credenciales.

Sauxe

El primer elemento que debe regir una selección adecuada de alcance de un marco tecnológico radica en las restricciones de diseño que el mismo asume, basados en las tecnologías, la capacidad técnica del equipo de desarrollo, los intereses jurídicos mercantiles así como la infraestructura tanto de la organización productora como cliente de los productos que el mismo facilitara desarrollar. Sauxe cuenta con una arquitectura en capas como se muestra en la **(Figura 2)** que a su vez presenta en su capa superior un

MVC⁴. Contiene un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo. Siguiendo el paradigma de independencia tecnológica por el cual apuesta el país, reutiliza las siguientes tecnologías libres (16):

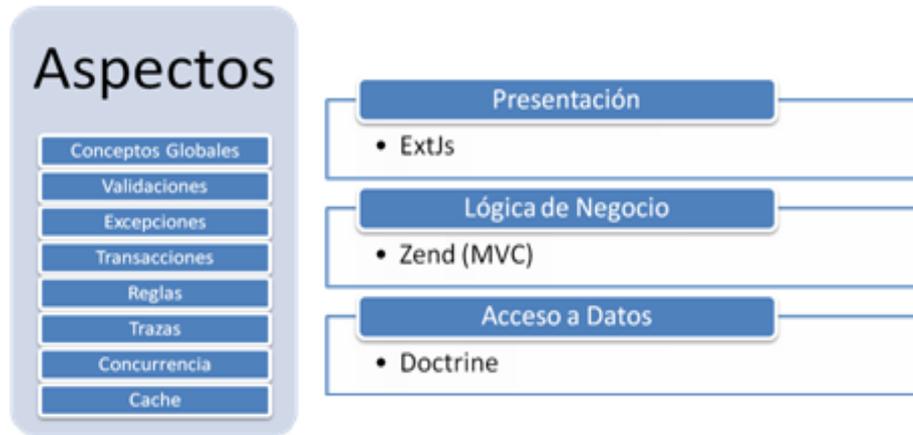


Figura 2 Arquitectura de Sauxe.

Extjs

Librería construida empleando JavaScript con el fin de desarrollar aplicaciones web interactivas usando tecnologías como AJAX y DOM. Ext es muy potente ya que contiene rica colección de componentes para el diseño de Interfaces Gráficas de Usuario (GUI's) del lado del cliente haciendo uso extensivo de AJAX. Dispone de un conjunto de componentes gráficos como:

1. Cuadros y áreas de texto.
2. Campos para fechas.
3. Campos numéricos.
4. Selectores estáticos y dinámicos.
5. Botones.
6. Editor HTML.
7. Elementos de datos (con modos de sólo lectura, datos ordenables, columnas que se pueden bloquear y arrastrar, etc.).
8. Árbol de datos.
9. Pestañas.
10. Barra de herramientas.
11. Menús al estilo de Windows.

⁴ **(MVC).Modelo Vista Controlador** es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se estará haciendo uso de Ext en su versión 2.2

Zend

Es un framework de alta calidad y de código abierto para el desarrollo de aplicaciones y servicios web con PHP.

Zend Framework brinda facilidades de uso y poderosas funcionalidades. Proporciona soluciones para construir modernos, robustos y seguros sitios web, está diseñado para php 5 y posee buenas capacidades de ampliación. Presenta entre otras, las siguientes características (17):

1. Proporciona un sistema de caché de forma que se puedan almacenar diferentes datos.
2. Proporcionan los componentes que forma la infraestructura del patrón MVC.
3. Proporciona una capa de acceso a base de datos, construida sobre PDO⁵ pero ampliándola con diferentes características.
4. Proporciona mecanismos de filtrado y validación de entradas de datos.
5. Permite convertir estructuras de datos PHP a JSON⁶ y viceversa, para su utilización en aplicaciones AJAX (especificado en el epígrafe 1.9.1).
6. Proporciona capacidades de búsqueda sobre documentos y contenidos.

Doctrine

Doctrine es un potente y completo sistema ORM⁷ (en inglés Object Relational Mapper) para PHP 5.2+ que incorpora una DBL (capa de abstracción a base de datos). Uno de sus rasgos importantes es la habilidad de escribir opcionalmente las preguntas de la base de datos orientada a objeto (18).

Esto les proporciona una alternativa poderosa a diseñadores de SQL manteniendo un máximo de flexibilidad sin requerir la duplicación del código innecesario.

Sus principales funcionalidades son:

- 1- Exporta una base de datos existente a sus clases correspondientes.
- 2- Convierte clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos.

⁵ PDO: (en inglés: PHP Data Objects) es una interface de acceso a datos que permite la conexión a diferentes bases de datos utilizando tecnología orientada a objetos.

⁶ JSON: es un formato ligero para el intercambio de datos.

⁷ ORM: Componente de software que permite trabajar con los datos persistidos como si fueran parte de una base de datos orientada a objetos.

Se estará haciendo uso de Doctrine en su versión 1.2.1.

1.3.12 Lenguajes de programación

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina.

PHP

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando distintas bibliotecas. El Pre-procesador de hipertextos (Hypertext Pre-processor, PHP por sus siglas en inglés) inicialmente se llamó PHP Tools, siendo publicada bajo licencia de software libre.

Sus principales características son:

- Es un lenguaje multiplataforma.
- Soporte para una gran cantidad de bases de datos.
- Integración con varias bibliotecas externas, permite generar documentos en PDF.
- Ofrece una solución simple y universal para las paginaciones web dinámicas y es de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Producto de código abierto.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones.

HTML

El Lenguaje de Marcas de Hipertexto (Hyper Text Markup Language, HTML por sus siglas en inglés) es el lenguaje de marcado predominante para el diseño de

construcción de páginas web. Usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. PHTML es una extensión para un tipo de páginas web que llevan código PHP y HTML para ser generadas. Cuando una página está escrita en PHP podemos encontrarla con varios tipos de extensiones como por ejemplo .php, .php4, .php3, o .phtml.

JavaScript

Javascript es un lenguaje de programación interpretado que permite a los desarrolladores crear acciones en sus páginas web. Es utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos pero mucho más complejos. Dentro de sus principales características están:

- Multiplataforma.
- Orientado a objetos.
- Se ejecuta del lado del cliente.
- Compatible con la mayoría de los navegadores.

1.3.13 Arquitectura Modelo- Vista- Controlador

El Modelo Vista Controlador (Model View Controller, MVC por sus siglas en inglés) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la interface de usuario y el código es el que provee de datos dinámicos a la página; el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Modelo: Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comprar un número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o importes en un carrito de la compra.

Vista: Esta presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y en la vista.

1.4 Conclusiones del capítulo

La Gestión de Credenciales es muy importante para el proceso de control de acceso dentro de la Gestión del Capital Humano en una entidad. En este capítulo se expuso un grupo de conceptos de vital importancia para la comprensión del software, se valoraron las tendencias y herramientas de este tipo existentes en la actualidad, tanto nacionales como internacionales. Pudiendo constatar que dentro de las encontradas no existe ninguna que cumpla con las particularidades requeridas para desarrollar el mismo con calidad, rapidez y organización, logrando la eficiencia e independencia tecnológica dentro de las entidades cubanas.

Capítulo 2: Descripción y análisis de la solución propuesta.

2.1 Introducción

Se tomará como punto de partida para este capítulo transformar los requisitos funcionales en el diseño del futuro componente para una posterior implementación a partir de la arquitectura previamente definida por la dirección del proyecto. Se obtendrán un conjunto de artefactos que serán de gran valor para las posteriores etapas del desarrollo como lo son: el modelo de componentes, el diseño de clases que comprende diagramas y descripción de las mismas y el modelo de datos. Igualmente se especificará la utilización de un conjunto de patrones dentro del diseño del componente.

El capítulo contendrá además elementos fundamentales de la implementación del componente, en ese caso se presentan: la organización del marco de trabajo, la definición de los estándares de codificación y la estructura de datos a utilizar, la descripción de clases y algoritmos implementados, la representación de la estrategia de integración y por último la descripción general del funcionamiento del componente.

2.2 Diseño de la solución arquitectónica

2.2.1 Valoración de la especificación de requisitos

El artefacto Especificación de Requisitos del Software para el Subsistema Capital Humano como resultado del previo análisis efectuado al proceso Gestión de Credenciales, constituye un elemento clave para el diseño y la posterior implementación.

Básicamente se caracteriza por presentar los requisitos de forma completa, estando definidas todas las responsabilidades del sistema respecto a los datos de entrada, válidos o no y respecto a los datos de salida. Presenta una adecuada organización y documentación; los términos, las tablas y los prototipos están correctamente descritos y referenciados. Cada requisito que comprende tiene una única interpretación evitando la ambigüedad en las definiciones y funcionalidades, están clasificados por la importancia arquitectónica y desde el punto de vista del cliente. Viabiliza las modificaciones, la comprobación y la trazabilidad (origen- implementación) de los requisitos especificados.

2.2.2 Principales funcionalidades

Tabla 2 Requisitos funcionales del subproceso Gestionar Credenciales

A continuación se presentan los requisitos funcionales de los subprocesos Gestionar

Agrupación	Requisito		Tipo	Comp
Gestionar Credenciales	Crear Credenciales	Listar personal	FUNC	ALTA
		Listar Plantillas Por roles		
		Crear paquete		
	Listar credenciales		FUNC	BAJA
	Listar paquetes		FUNC	MEDIA
	Bloquear credenciales		FUNC	BAJA
	Desbloquear credenciales		FUNC	BAJA
	Imprimir	Vista previa	FUNC	ALTA
	Listar Reporte De impresión	Guardar Reporte	FUNC	MEDIA
		Imprimir Reporte		
Eliminar credenciales		FUNC	MEDIA	

Credencial y Gestionar Plantillas respectivamente.

Tabla 3 Requisitos funcionales del subproceso Gestionar Credenciales

Agrupación	Requisito		Tipo	Comp
Gestionar Plantillas	Crear plantillas	Cargar plantilla	FUNC	ALTA
		Configurar plantilla		
		Guardar plantilla		
	Modificar plantilla		FUNC	MEDIA
	Eliminar plantilla		FUNC	BAJA
	Listar plantillas		FUNC	BAJA

2.3 Diagrama de clases del Diseño

Los diagramas de clases según la clasificación UML son diagramas de estructura estática donde la representación de los requerimientos se lleva a cabo a través de las clases del sistema y sus interrelaciones. Representan una abstracción del dominio de modo que es formalizado el análisis de conceptos y constituyen el pilar básico del modelado, mostrando en términos generales qué debe hacer el sistema.

Específicamente los diagramas de clases de diseño son muy útiles porque muestran a través de atributos y métodos la estructura de las clases que después serán escritas en algún lenguaje de programación (PHP y JavaScript en este caso).

A continuación se muestran en las figuras 6 y 7 los diagramas de diseño relacionados con los procesos que se identificaron en el análisis, mostrando en los mismos la relaciones entre las clases que conforman la aplicación, donde se da una idea de cómo sería su funcionamiento a modo general dentro del marco de trabajo y se hace énfasis en los métodos más importantes de algunas de las clases, por ejemplo, en el caso del proceso Gestionar Plantillas las clase OpGestionarPlantillaControler y OpGestionarPlantillaModel (Ver Figura 6) y en el proceso Gestionar Credencial OpGestionarCredencialControler y OpGestionarCredencialModel (Ver Figura 7). Su desarrollo estuvo sobre la base que define el marco de trabajo Sauxe, el cual brinda una arquitectura basada en capas, que implementa un MVC en las capas superiores, algo que se puede apreciar con facilidad en los diagramas que se muestran a continuación.

2.3.1 Diagrama de clases del diseño del proceso Gestionar Plantillas

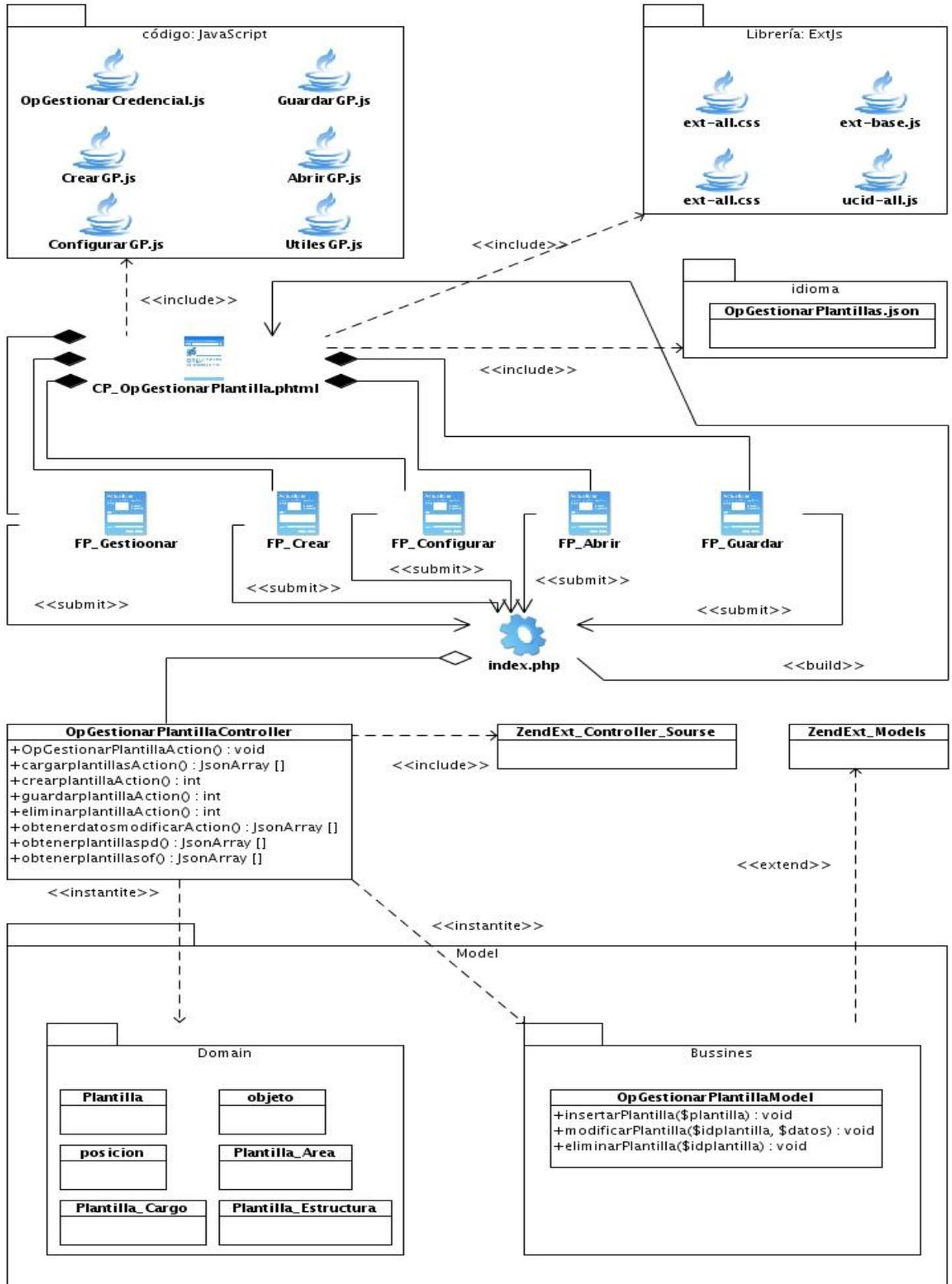


Figura 3 Diagrama de clases del diseño del proceso Gestionar Plantillas

2.3.2 Diagrama de clases del diseño del proceso Gestionar Credenciales

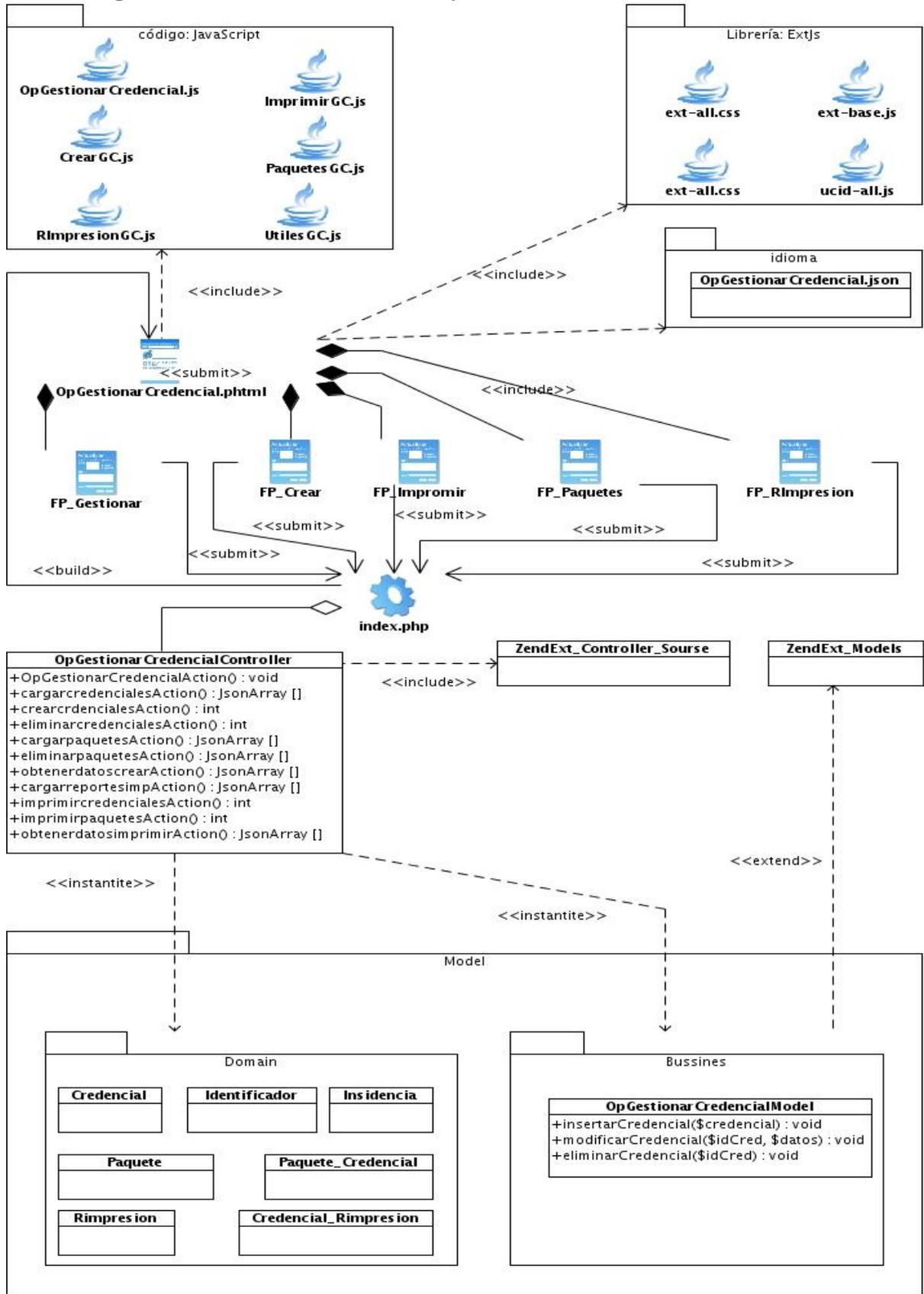


Figura 4 Diagrama de clases del diseño Gestionar Credenciales

2.4 Modelo Entidad-Relación

Un modelo de datos es una colección de conceptos bien definidos matemáticamente que ayudan a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo.

El modelo de datos propuesto para la solución, a partir de los requisitos obtenidos por los analistas generó un total de 13 entidades, utilizando 6 para la gestión de plantillas y 7 para la gestión de credenciales, para su construcción se tuvo en cuenta la reducción a la mínima expresión de los campos nulos.

Seguidamente se mostrará una explicación detallada de las entidades fundamentales utilizadas en el modelo de datos por cada uno de los subprocesos a implementar para la Gestión de las Credenciales identificados en los requisitos funcionales.

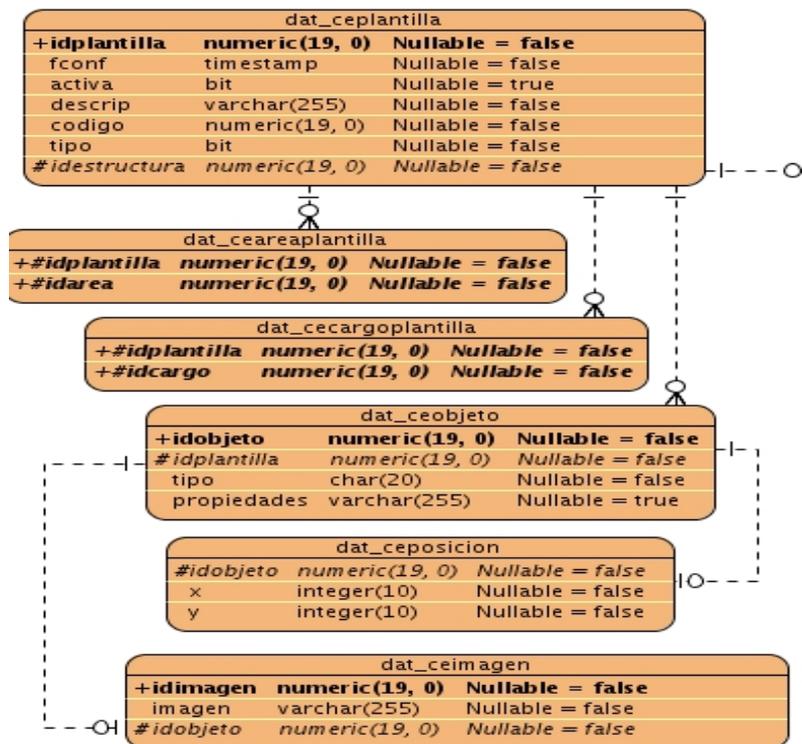


Figura 5 Entidades del modelo de datos asociadas al subproceso Gestionar Plantillas

Nombre de la entidad	Descripción	Relaciones
plantilla	Es la encargada de almacenar la relación de los datos fundamentales que forman las plantillas, los cuales serán	Se relaciona con las tablas objeto , plantilla_area , plantilla_cargo , plantilla_estructura y

	utilizados a la hora de generar las credenciales.	credencial.
objeto	En ella se almacenan las propiedades de los objetos que conforman la credencial.	Se relaciona con la tabla plantilla y posición .
posicion	Es una especialización de objeto y almacena la posición de los objetos, que cuentan con esta propiedad.	Se relaciona con objeto .

Tabla 4 Descripción de las entidades del subproceso Gestionar Plantillas

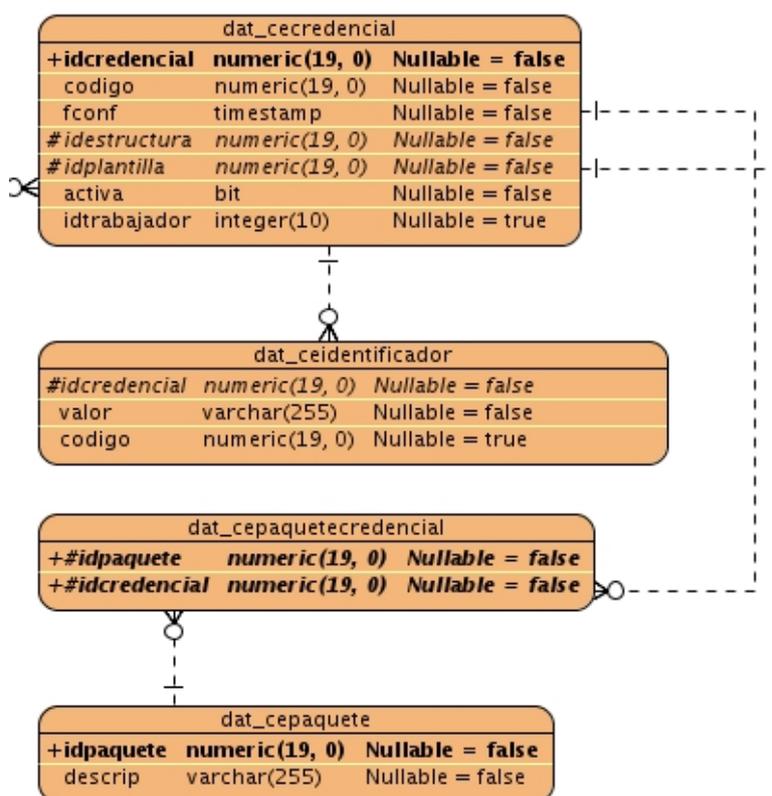


Figura 6 Entidades del modelo de datos asociadas al subproceso Gestionar Credenciales

Nombre de la entidad	Descripción	Relaciones
credencial	Es la encargada de almacenar la relación de los datos fundamentales	Se relaciona con plantilla , identificador , paquete_credencial ,

	que forman las credenciales, a partir de las plantillas diseñadas con anterioridad, haciendo coincidir sus objetos dinámicos con los datos asociados a la entidad empleado que se relaciona con la credencial.	credencial_rimpresion.
identificador	En este se guardan los identificadores que se definieron en la plantilla y que se generan de forma automática en el momento en que se crea la credencial. Código de Barras, valores alfanuméricos, etc.	Se relaciona con credencial.
paquete_credencial	Surge a partir de la relación de muchos a muchos que existe entre la tabla paquete y la tabla credencial, esto da la posibilidad de agrupar credenciales ya creadas para imprimirlas o realizar alguna operación sobre ellas de manera conjunta.	Se relaciona con credencial y paquete.

Tabla 5 Descripción de las entidades del subproceso Gestionar Credenciales

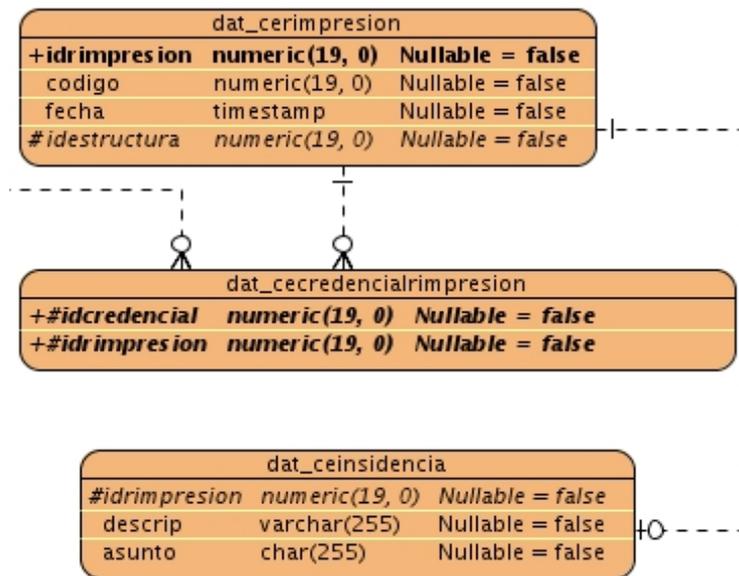


Figura 7 Entidades del modelo de datos asociadas al subproceso reporte de impresión

Nombre de la entidad	Descripción	Relaciones
rimpression	Almacena los reportes de las credenciales que se imprimen, almacenando las incidencias en la tabla incidencia en caso de que exista alguna a por la que se procede a imprimir la credencial.	Se relaciona con credencial_rimpresion e incidencia .

Tabla 6 Descripción de las entidades del subproceso Reporte de Impresión

Nota: Para comprender con mayor claridad la relación en el modelo de datos de las tablas anteriormente analizadas ver el Anexo II en la página 57.

2.5 Estructura de la solución en términos de componentes

El componente credencial debe ser diseñado e implementado. La Gestión de credenciales es la esencia del mismo, el cual necesita consumir los servicios que le proporcionan la información de las personas, trabajadores, áreas y cargos para

realizar sus funcionalidades, estos servicios se encuentran ubicados en distintos componentes que representan a su vez subcomponentes del componente Capital Humano (**Ver Figura 3**).

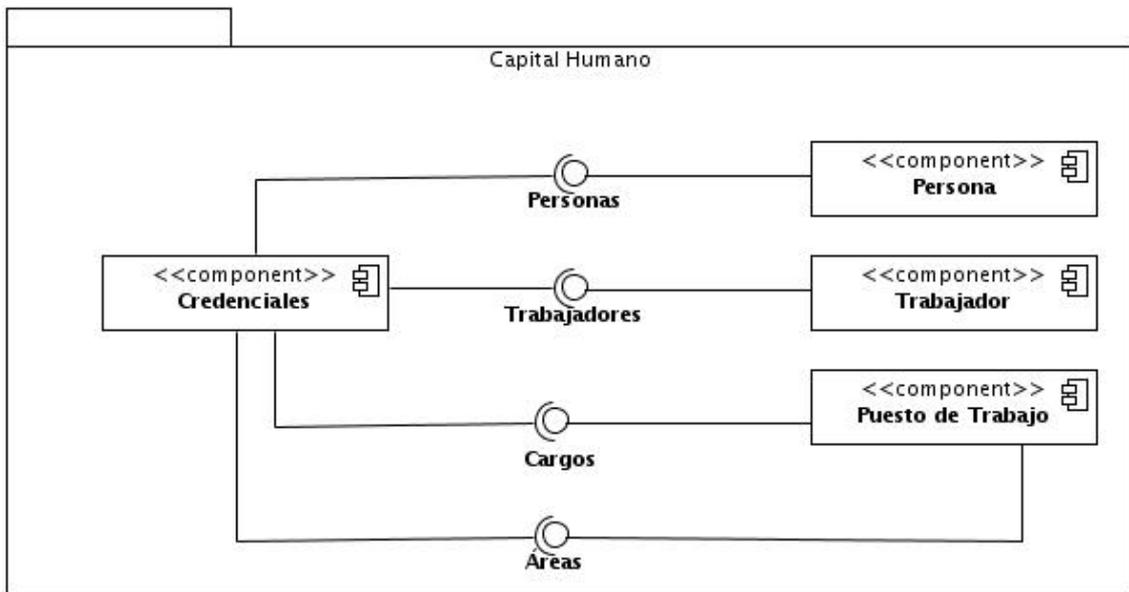


Figura 8 Diagrama de Estructura de Componentes

2.6 Patrones de diseño empleados

El diseño fue elaborado siguiendo patrones basados en la experiencia, que de manera general constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. En este caso se emplearon los patrones GRASP (en inglés General Responsibility Assignment Software Patterns), los que describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Los patrones GRASP que se utilizaron son los siguientes:

Experto: Dicho patrón es evidenciado en la definición de las clases de acuerdo a las funcionalidades que deben realizar a partir de la información manejada dentro del componente, como por ejemplo las clases controladoras y las del modelo. Específicamente: la clase OpGestionarPlantilla, será la responsable de efectuar las operaciones que conciernen a las funciones: crear, eliminar y actualizar las plantillas,

asumiendo toda la lógica para cada una de ellas. Sobre este mismo principio se realiza el diseño de las restantes funcionalidades.

Creador: Este patrón es adaptable a las clases del paquete Domain, quienes son las encargadas de crear los objetos de tipo Doctrine_Query, para permitir el acceso a la información almacenada a nivel de datos.

Alta cohesión: Este patrón fue utilizado en el diseño del componente de manera general; donde se agruparon las clases en dependencia de los requerimientos a los que se les debía dar respuesta, según la premisa de que cada clase debe implementar las operaciones que estén sobre la misma área funcional.

Bajo acoplamiento: Un ejemplo de su uso fue en el modelo de datos, donde se definieron un conjunto de clases persistentes, entre las cuales se establecieron las relaciones necesarias de manera que fueran más independientes y reutilizables para reducir el impacto de los cambios y acrecentar la oportunidad de una mayor productividad.

Controlador: Las clases controladoras definidas: OpGestionarPlantillaController y OpGestionarCredencialController son un ejemplo de la aplicación de este patrón, las mismas tendrán a cargo la responsabilidad de manejar los eventos dentro del componente.

Durante el diseño del componente se emplearon patrones GOF, específicamente:

Fachada: La aplicación de este patrón en el componente Credenciales se evidencia en la interfaz de servicios simples que se proporcionan para establecer la comunicación con otros componentes dentro y fuera del Subsistema.

Mediador: La comunicación en la base de datos se puede tornar compleja debido a las dependencias marcadas entre las tablas que la componen, esto resulta engorroso a la hora de acceder a un determinado valor. La solución a este inconveniente viene dada por la utilización de este patrón; específicamente; creando una nueva tabla entre todas las tablas unidas mediante una relación de muchos a muchos. La tabla mediadora posee una relación de uno a muchos con las vinculadas a ella. De esta forma, el comportamiento distribuido entre las clases queda adaptado a las circunstancias y necesidades del diseño.

Dicha operación se concreta en el modelo de datos antes presentado entre las tablas asociadas a **Credencial** y **Paquete** donde la tabla mediadora resultó ser **Paquete_Credencial**.

Cadena de Responsabilidad: Está concebido que ante la ocurrencia de un error al realizarse una determinada consulta a la base de datos el mismo sea manejado por el Modelo, creando una excepción de tipo ZendExt_Exception. Dicha excepción debe ser propagada al Controlador, el cual será el encargado de capturarla y enviarla a la Vista ya traducida, esta última por su parte mostrará un mensaje al usuario en un lenguaje entendible notificando el error y sin especificar detalles del mismo. De esta manera se distribuyen las responsabilidades entre las diferentes componentes, evidenciándose por lo tanto el empleo de este patrón.

La Arquitectura Base y el diseño flexible y escalable a través del correcto uso de patrones de diseño en la generación de los artefactos necesarios para el desarrollo, posibilitaron crear una entrada apropiada como punto de partida a las actividades de implementación, con la máxima de lograr una mayor calidad del producto y la satisfacción del cliente.

2.7 Implementación

Para la implementación de esta herramienta se hizo uso del marco de trabajo Sauxe, establecido por el proyecto ERP Cuba para el desarrollo del Sistema integral de gestión Cedrux. Además de los estándares de codificación PascalCasing, CamelCasing y Notación húngara, también establecidos por el proyecto.

2.7.1 Estándares de codificación

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura para facilitar la lectura, comprensión y mantenimiento del código. Debido a la complejidad del sistema Cedrux, el numeroso personal involucrado en él y el alto nivel de integración existente entre sus componentes, el grupo arquitectónico del proyecto definió normas de codificación con el fin de obtener un estándar en la implementación por el equipo de desarrollo que permitiera asegurar la calidad del software, obteniendo un código más legible y reutilizable.

Seguidamente se muestran algunos de los estándares utilizados en la implementación del componente Credenciales.

PascalCasing

El estándar PascalCasing establece que los identificadores, nombres de clases, variables, métodos o funciones están compuestos por múltiples palabras juntas, iniciando cada palabra con letra mayúscula. La nomenclatura de las clases del componente Credenciales se realizó sobre la base de este estándar, usando palabras compuestas y sugerentes acordes al propósito de la misma. (19)

CamelCasing

El estándar camelCasing es parecido al PascalCasing con la particularidad de que la letra inicial del identificador no comienza con mayúscula. Esta notación se utilizó para el nombre de funciones y atributos. (19)

Notación húngara

Esta convención, también conocida como notación: REDDICK por el nombre de su creador, se basa en definir prefijos para cada tipo de datos según el ámbito de las variables con el fin de brindar mayor información al nombre de la variable, método o función. La notación húngara fue utilizada para la definición de variables de acuerdo con los siguientes prefijos (**Tabla 7**) (19):

Tipos de datos	Prefijos
Arreglos	arr
Objetos	obj
Enteros	int
Cadena	str
Float	flt
Boolean	bool

Tabla 7 Prefijos para los tipos de datos (19).

2.8 Conclusiones

La Arquitectura Base definida y los artefactos generados como resultado de la especificación de los requisitos correspondientes al componente Credenciales, fueron la base para la realización del presente capítulo a partir de los cuales fue posible llevar a cabo el diseño y la implementación del componente, con el fin de responder a las funcionalidades claves en la Gestión de Credenciales.

Capítulo 3: Validación de la solución propuesta.

3.1 Introducción

La existencia de errores en los software durante su desarrollo e incluso después de haber concluido el mismo tiene altas posibilidades de evidenciarse. Para evitar y mitigar este problema, el desarrollo de software debe ir acompañado de alguna actividad que garantice la calidad y a pesar de que es imposible asegurar que un software está completamente libre de errores, existen formas y métodos para acercarse a un resultado óptimo; un ejemplo de ello es la prueba, que se considera un elemento crítico para garantizar la calidad del software.

La prueba y validación de los resultados deben estar presentes durante todas las etapas del desarrollo. Es de vital importancia medir la cobertura de las pruebas, es decir, determinar cuándo se han realizado las pruebas necesarias; continuando con las mismas mientras se sigan detectando errores.

En el presente capítulo se realiza la validación de la solución propuesta. Se evalúa el diseño empleando métricas de software que proporcionan una medida de la complejidad y calidad del software. Se aplican pruebas para verificar la funcionalidad y estructura de cada componente desarrollado.

3.2 Métricas usadas para la evaluación del modelo del diseño propuesto

Las métricas utilizadas durante la evaluación de la solución del diseño son las siguientes:

3.2.1 Métrica Tamaño operacional de clases (TOC)

Tamaño operacional de de clase(TOC)	
Descripción	El TOC está dado por el número de métodos asignados a una clase.
Atributos que afecta	Modo en que lo afecta.
Responsabilidad	El aumento del TOC provoca un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	El aumento del TOC provoca un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento del TOC provoca una disminución en el grado de reutilización de la clase.

Tabla 8 Métrica Tamaño Operacional de Clase (TOC).

Resultados del instrumento de evaluación de la métrica Tamaño Operacional de Clase (TOC).

Atributo	Categoría	Criterio
Responsabilidad	Baja	< =Prom.
	Media	Entre Prom. y 2* Prom.
	Alta	> 2* Prom.
Complejidad de implementación	Baja	< =Prom.
	Media	Entre Prom. y 2* Prom.
	Alta	> 2* Prom.
Reutilización	Baja	> 2*Prom.
	Media	Entre Prom. y 2* Prom.
	Alta	<= Prom.

Tabla 9 Rango de valores de para la evaluación técnica de los atributos de calidad relacionados con la métrica TOC.

Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos:

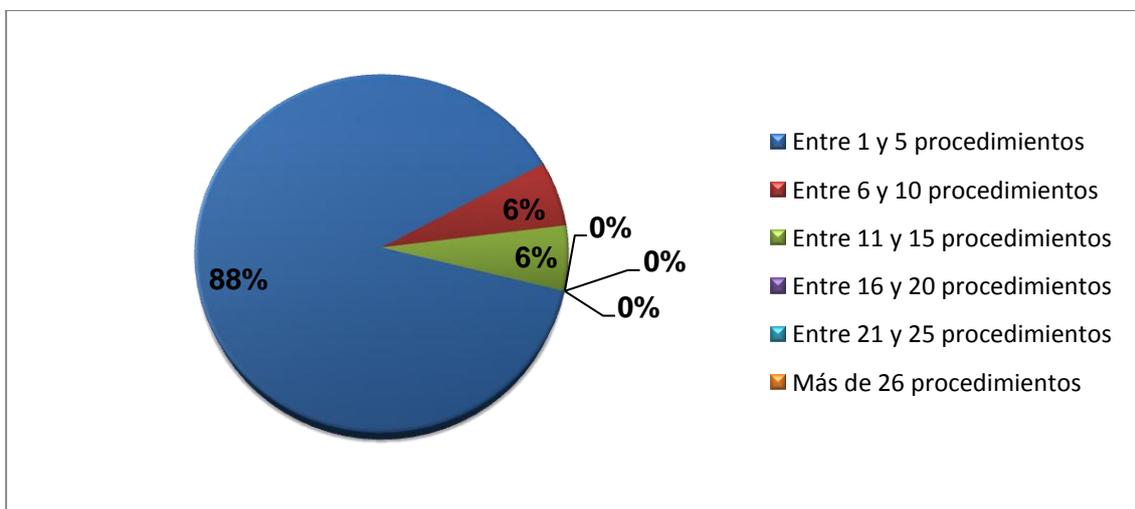


Figura 9 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad:

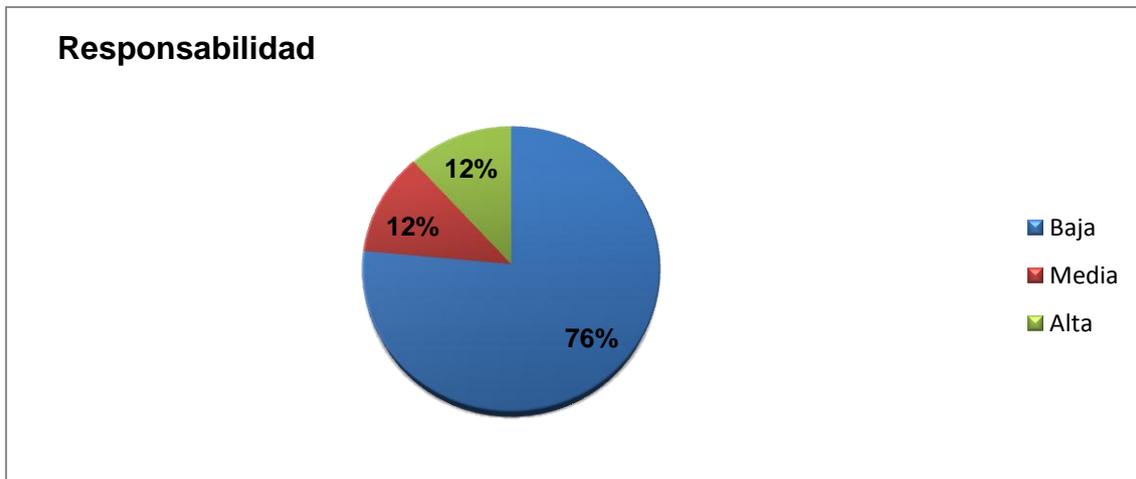


Figura 10 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.

Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación:

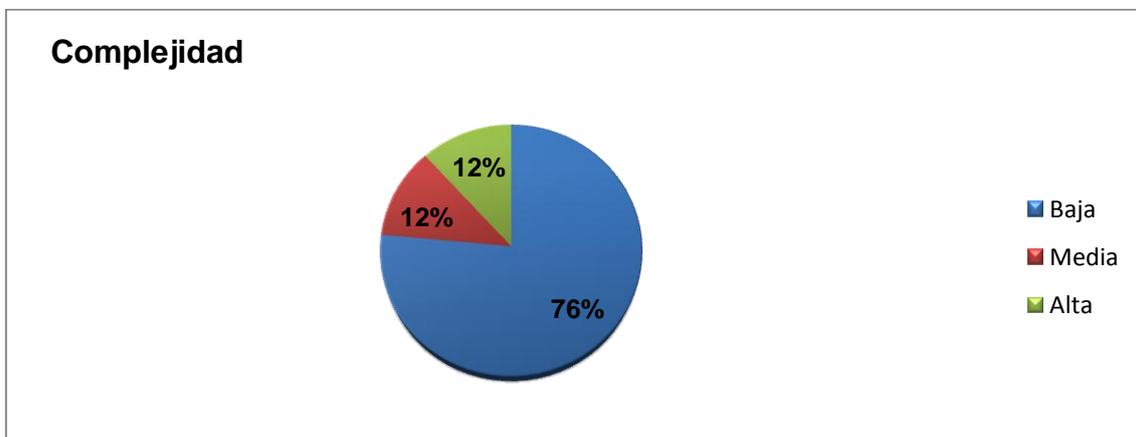


Figura 11 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación.

Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización:

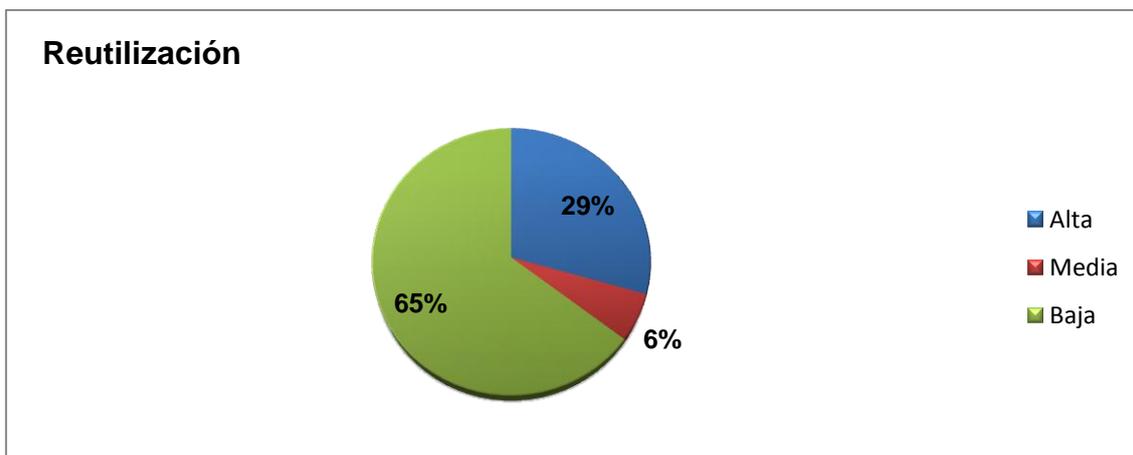


Figura 12 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.

Los resultados obtenidos durante la evaluación del instrumento de medición de la métrica TOC demuestran que el diseño propuesto para el componente Credenciales se encuentra dentro de los niveles aceptables de calidad, mostrando que más de la mitad de las clases (88%) poseen menos cantidad de operaciones que la media registrada en las mediciones. Para el 76% de las clases los atributos de calidad fueron evaluados satisfactoriamente, confirmando la elevada reutilización, baja complejidad y responsabilidad en el diseño propuesto.

3.2.2 Métrica Relaciones entre clases (RC)

Relaciones entre clases (RC)	
Descripción	Las relaciones entre clases están dadas por el número de relaciones de uso de una clase con otras.
Atributos que afecta	Modo en que lo afecta.
Acoplamiento	El aumento del RC provoca un aumento del Acoplamiento de la clase.
Complejidad del mantenimiento	El aumento del RC provoca un aumento de la complejidad del mantenimiento de la clase.
Reutilización	El aumento del RC provoca una disminución en el grado de reutilización de la clase.

Cantidad de pruebas	El aumento del RC provoca un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.
----------------------------	--

Tabla 10 Métrica Relaciones entre Clases (RC).

Resultados del instrumento de evaluación de la métrica Relaciones entre clases (RC).

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	> 2
Complejidad de Mantenimiento	Baja	< =Prom.
	Media	Entre Prom. y 2* Prom.
	Alta	> 2* Prom.
Reutilización	Baja	> 2*Prom.
	Media	Entre Prom. y 2* Prom.
	Alta	<= Prom.
Cantidad de Pruebas	Baja	< =Prom.
	Media	Entre Prom. y 2* Prom.
	Alta	> 2* Prom.

Tabla 11 Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica RC.

Representación en % de las dependencias entre clases obtenidas en los intervalos definidos:

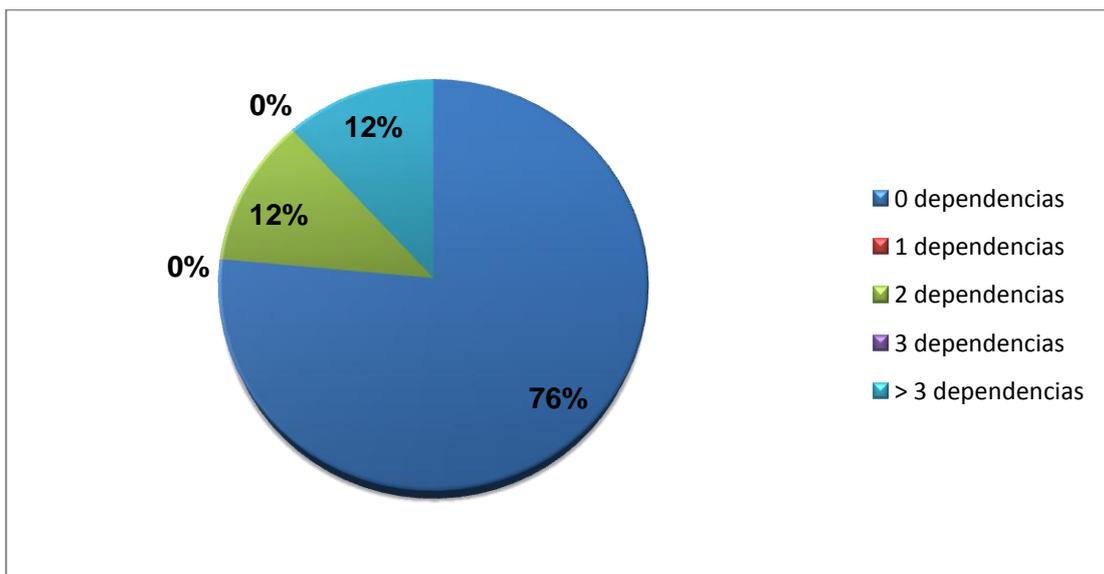


Figura 13 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento:

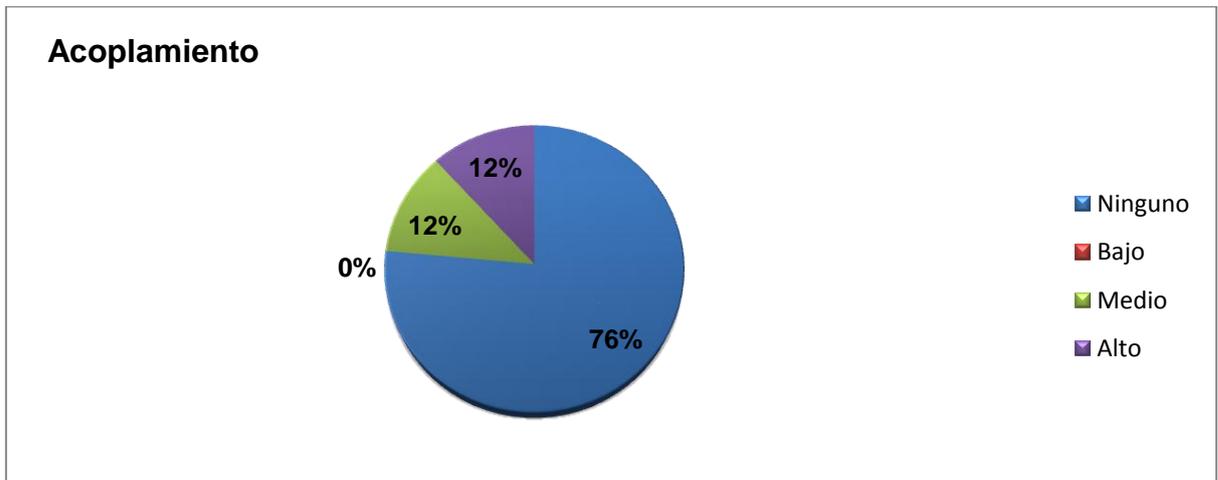


Figura 14 Representación de la incidencia en % de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.

Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento:

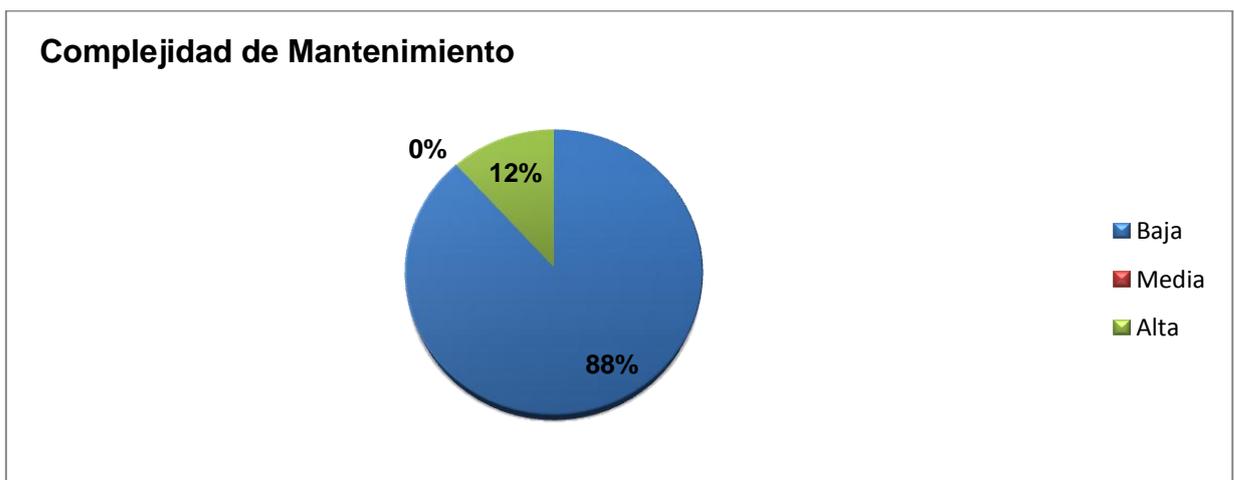


Figura 15 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento.

Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas:

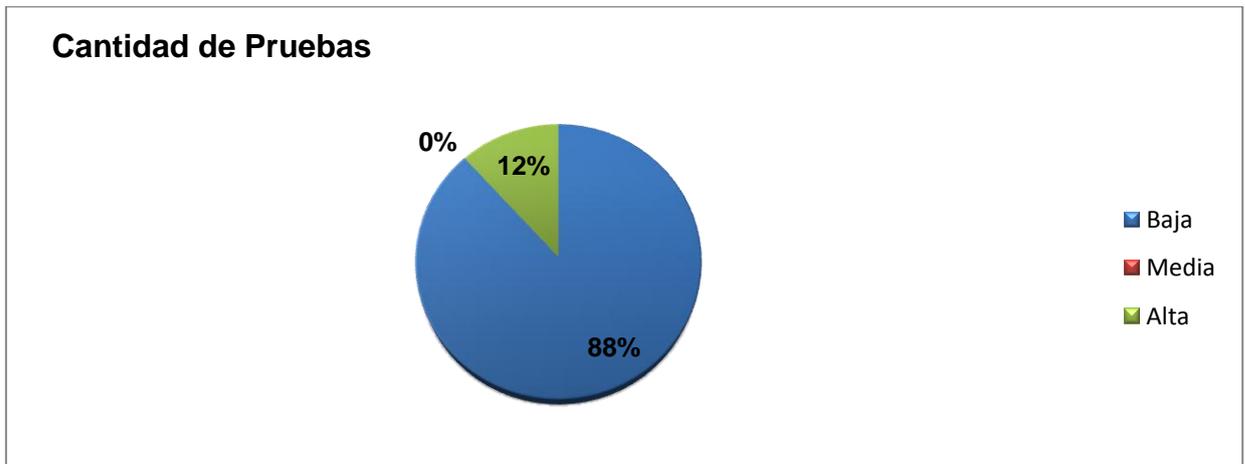


Figura 16 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas.

Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización:

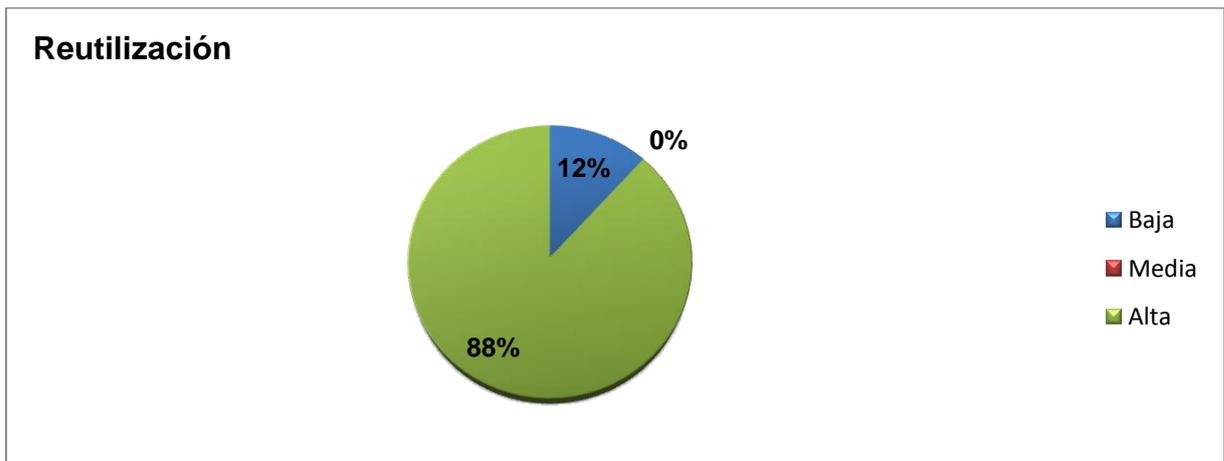


Figura 17 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización.

Los resultados obtenidos durante la evaluación del instrumento de medición de la métrica RC demuestran que el diseño propuesto para el componente Credenciales se encuentra dentro de los niveles aceptables de calidad, mostrando que el 76% de las clases poseen menos de 3 dependencias entre clases. Los atributos de calidad fueron evaluados satisfactoriamente para el 88% de las clases, confirmando la elevada reutilización y bajo acoplamiento, complejidad y cantidad de pruebas en el diseño propuesto. Las métricas de software aplicadas posibilitaron estimar la calidad de los atributos internos del producto, demostrando una aceptable calidad de diseño. La

solución propuesta contribuirá a la disminución de disturbios durante la implementación del componente, garantizando la reutilización y agilidad en el proceso de desarrollo de software.

3.3 Niveles de pruebas aplicados

En el desarrollo del componente se definieron pruebas para diferentes objetivos, escenarios o niveles de trabajo. Para aplicarlas se evalúa el sistema empezando por los procesos más simples y pequeños, avanzando progresivamente hasta probar todo el software en su conjunto.

- Pruebas de desarrollo: Diseñadas e implementadas por el equipo de desarrollo.
- Pruebas Unitarias: Comienzan con la prueba de cada componente. Verifican que los flujos de control y de datos estén cubiertos y que funciona como se espera.
- Pruebas de integración: Verifican que los componentes probados operen correctamente, cuando se combinan y descubren errores en las especificaciones de las interfaces de los paquetes.
- Pruebas de aceptación: El cliente comprueba que el software funciona según sus expectativas y se encuentra ejecutando las funciones y tareas para las cuales fue construido.

3.4 Estrategia de pruebas

Para llevar a cabo el proceso de pruebas al componente Credencial se definió una estrategia de pruebas con el propósito de garantizar la calidad del software. Se determinó aplicar, a partir del método de caja negra, las técnicas de Partición de Equivalencia y Camino Básico respectivamente, en los niveles de Desarrollador, Unidad e Integración. Este método no se trata de manera separada a los niveles; al estar estrechamente relacionados no se hace distinción específica del mismo entre uno u otro nivel. En los niveles de desarrollador y de unidad, a medida que se realice la implementación, se aplicarán pruebas comprobando que cada funcionalidad implementada se ajuste a los requerimientos. En el nivel integración se realizará la verificación y validación de las funcionalidades del componente, como un conjunto de componentes integrados y combinados según la dependencia jerárquica y la comunicación entre ellos. De igual forma, las técnicas de prueba no se hallan de forma aislada, sino como un conjunto integrado de acciones que combinadas permitirán verificar y evaluar la calidad de software. Se utiliza la técnica de Partición de Equivalencia para definir casos de prueba que descubran clases de errores,

reduciendo el número de casos de prueba a desarrollar para demostrar que las funciones del software son operativas; las entradas se aceptan de forma adecuada y se producen salidas correctas.

Independiente a la estrategia de pruebas definida, existe un equipo de calidad encargado de probar el módulo para su liberación posterior a las entidades piloto donde el cliente comprobará que el software funciona según sus expectativas, y se encuentra ejecutando las funciones y tareas para las cuales fue construido.

3.5 Diseño de casos de prueba para Caja Negra

Los casos de prueba para caja negra se basan en las diferentes entradas que puede recibir el software, y sus correspondientes valores de salida; están centrados en realizar pruebas del software a través de la funcionalidad.

Los casos de prueba demuestran:

- Que las funciones de software son operativas.
- Las entradas se aceptan de la forma adecuada produciendo el resultado correcto.
- La integridad de la información interna (Ejemplo: archivos de datos) se mantiene.

Para verificar que la aplicación se comporta según los requerimientos establecidos por el cliente, se diseñan casos de pruebas usando el método de caja negra. A continuación se especifica el caso de prueba para el requisito "Crear Credencial" el cual a partir de la selección de una plantilla prediseñada y un trabajador genera una credencial o un paquete de las mismas en caso que se seleccione un grupo de trabajadores, para ser impresa en ese momento o posteriormente.

Condiciones de ejecución:

- Se debe identificar y autenticar ante el sistema y además debe tener los permisos para ejecutar esta acción.
- Se debe seleccionar el subsistema Capital Humano.
- Se debe seleccionar la opción Administración => Credenciales => Gestión de Credencial.
- Debe existir al menos una plantilla y un trabajador registrados en el sistema.

En las tablas que se muestran a continuación se el caso de prueba para el requisito crear credencial.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo de escenarios
Anexo I. Crear Credencial.	El sistema genera una credencial a partir de una plantilla prediseñada y los datos de un trabajador.	EP 1.1: Crear credenciales considerando los datos válidos.	<ul style="list-style-type: none"> Se especifican los datos de la credencial. El sistema crea la credencial.
		EP 1.2: Crear credenciales sin seleccionar una plantilla, un trabajador o ambos.	<ul style="list-style-type: none"> Se especifican los datos de la credencial sin haber seleccionado una plantilla, un trabajador o ambos.

No.	Nombre del campo	Tipo	Válido	Inválido
1	Plantilla prediseñada	Estructura Arbórea	NA	NA
2	Trabajador	Lista	NA	NA

ID. del escenario	Escenario	Plantilla	Trabajador	Respuesta del sistema	Resultado de la prueba
EP 1.1	Seleccionar una plantilla y un trabajador para crear la credencial.	V(87-C Trabajadores)	V(87042523846)	El sistema crea una credencial con el diseño de la plantilla seleccionada y los datos del	

				trabajador.	
EP 1.2	Crear credencial sin haber seleccionado un trabajador y una plantilla.	I(Vacio)	I(Vacio)	El sistema muestra un mensaje de Error.	

3.6 Conclusiones

Durante el desarrollo de este capítulo se ejecutaron métricas de diseño para caracterizar numéricamente los distintos aspectos del desarrollo y se realizaron pruebas que permitieron evaluar todos los elementos del software. Fueron validadas las funcionalidades implementadas, a través de diferentes pruebas de caja negra, mostrando cómo respondían adecuadamente a los requisitos funcionales y garantizando la satisfacción plena de las necesidades reales de los usuarios y demandas del cliente. A partir de los resultados de las métricas y las pruebas aplicadas al sistema, se obtuvo un código de mayor calidad, funcionalmente probado, y se evaluaron satisfactoriamente los atributos relacionados con el desarrollo del software.

Conclusiones Generales

A partir del análisis de los procesos de la gestión de credenciales y sistemas informáticos vinculados a esta actividad, se demostró la necesidad e importancia de desarrollar un sistema informático para desarrollar una mejor gestión de credenciales en las entidades cubanas facilitando la gestión de control de asistencia en el subsistema Capital Humano perteneciente al sistema integral de gestión Cedrux.

Para ello se realizó un estudio de las herramientas y tecnologías que posibilitó entender y avalar la selección de técnicas utilizadas. Efectuándose a su vez el diseño e implementación del componente credenciales para el sistema integral de gestión de entidades Cedrux, probándose y valorándose mediante métricas y pruebas de software.

De acuerdo con el objetivo general propuesto se obtuvo el componente “Credenciales” como producto confiable y totalmente funcional que cumple con los requerimientos descritos, permitiendo el diseño y confección de credenciales de manera automática a partir de datos registrados de trabajadores en el sistema integral de gestión de entidades Cedrux, además de generar reportes de impresión de las credenciales e inhabilitar las mismas por algún tipo de incidencia que lo requiera. Este sistema nos brinda también la posibilidad de integrarse a otros subsistemas y módulos de los mismos, como es el caso del módulo Control de Asistencia, del subsistema Capital Humano del sistema integral de gestión de entidades Cedrux.

La investigación y desarrollo realizados constituye un aporte a la actual gestión de credenciales para las entidades cubanas, facilitando este proceso en aquellas que utilicen Cedrux como sistema integral de gestión de entidades.

Recomendaciones

Concluido el siguiente trabajo de diploma, considerando cumplidos los objetivos trazados en el mismo se recomienda:

- Continuar realizando pruebas de calidad al componente.
- Realizar el despliegue de la aplicación.

Trabajos citados

1. **ABOUT, ILSEN y DENIS, VINCENT.** *HISTORIA DE LA IDENTIFICACION DE LAS PERSONAS.* 2011.
2. Seragro. <http://seragro.cl/?a=329>. [En línea] 2011.
3. **Fernández González, Mairelys y Zorrilla Rivera, Osley.** *Diseño e implementación del componente Ajuste al Costo del Subsistema Costos y Procesos del.* 2010.
4. Elipse.
<http://www.elipse.cl/productos/tarjetas%20credenciales/credenciales%20de%20identificacion%20codigo%20de%20barras%20proximidad%20hid%20pvc%20fotografia.html>. [En línea] 2011.
5. **Gómez Urquiza, Guillermo y Gil Martín, Ing. Manuel Alejandro.** *SISTEMA DE GESTION DE CREDENCIALES.* 2007.
6. **GORDÓN DÍAZ, NATHALI YESSENIA.** *CONTROL DE ACCESO EN LA ENTRADA DEL INSTITUTO.* Ecuador : s.n., 2009.
7. **Alphacard.** Alphacard. <http://store.alphacard.com/alpha-card-professional-id-card-software>. [En línea] 2011.
8. **XPRESS.** Advanced ID Creator. <http://www.advancedidcreator.com/>. [En línea] 2011.
9. ALTATEC. *CardFive - Software de diseño de Credenciales.* [En línea] NFIVE, 2011. <http://www.altatec.com.mx/html/index.php?module=Quotation&func=product&pid=8&product=CardFive+++Software+de+dise%C3%B1o+de+Credenciales>.
10. CARD DATA SYSTEMS. [En línea] 2011. <http://carddatasystems.com/idworks.html>.
11. **Vega, Yanet.** *Definición del ciclo de vida del proyecto.* 2009.
12. **SCRIBD.** Herramientas CASE. [En línea] http://es.scribd.com/doc/36908565/Herramientas-Case#ad_unit=Doc_Sideboard_MediumRectangle_BTF_300x250&url=http%3A/es.scribd.com/doc/36908565/Herramientas-Case%23ad_unit%3DDoc_Sideboard_MediumRectangle_BTF_300x250%26url%3Dhttp%253A/es.scribd.com/doc/3690.
13. visual-paradigm. *visual-paradigm.* [En línea] 2005. [Citado el: 15 de 04 de 2011.] <http://www.visual-paradigm.com>.
14. Documentación del Servidor HTTP Apache 2.0. [En línea] 2009. [Citado el: 12 de 04 de 2011.] <http://httpd.apache.org/docs/2.0/es/>.
15. Subversion. [En línea] 2010. [Citado el: 23 de 04 de 2011.] <http://subversion.apache.org/>.

16. **Gómez Baryolo, Ing. Oiner, Morejón Borbón, Ing. Yoandry y García Tejo, Ing. Darien.** *ARQUITECTURA TECNOLÓGICA PARA EL DESARROLLO DE SOFTWARE*. 2011.
17. Zend Framework. [En línea] [Citado el: 10 de 02 de 2011.] <http://framework.zend.com/manual/en/>.
18. Doctrine. *Doctrine*. [En línea] [Citado el: 10 de 04 de 2011.] <http://www.doctrine-project.org>.
19. **Ramos Arias, Taimé y Torres Salas, Pedro Antonio.** *Diseño e implementación del módulo Banco del Sistema Integral de Gestión CEDRUX*. 2011.
20. **Arencibia Ramírez, Lic. Joel y Otero Dartayet, Dasiel.** *Subsistema para el manejo y reportes de credenciales*. 2007.
21. **Delgado Kios, Lic. Yeneit y Quevedo Montero, Yasel Rafael.** *Propuesta de Arquitectura para el Sistema de Gestión de Credenciales*. 2008.
22. **Fuentes, Lidia, Troya, Jose M. y Vallecillo, Antonio.** *Desarrollo de Software Basado en Componentes*. Malaga, Spain : s.n., 2007.
23. **Fernandez Leyet, Osmar.** *Documento Línea Base de proyecto-CEDRUX-1.0*. Ciudad de la Habana : s.n., 2010.
24. **Unidad de Compatibilización Integración y Desarrollo De Software para la Defensa.** *Proceso de Desarrollo y Gestión de Proyectos de Software (1ra Versión)*. 2009.
25. Avicard. <http://www.avicard.com.uy/productos/identificacion>. [En línea] 2011.
26. Archivert. <http://www.archivert.cl/tarjetas-y-credenciales-colgante.htm>. [En línea] 2011.
27. **Quevedo Montero, Yasel Rafael.** *Propuesta de Arquitectura para el Sistema de Gestión de Credenciales*. 2008.
28. Diccionarios.com. [En línea] 2011. http://www.diccionarios.com/detalle.php?palabra=dispositivo&modo=4&dicc_51=on.
29. **Vallecillo, Antonio, Fuentes, Lidia y Troya, José M.** *Desarrollo de Software Basado en Componentes*. Malaga, España : s.n., 2007.
30. **Prieto, Félix.** *Patrones de diseño*. 2009.
31. **Pérez Pérez, Joisel, Chaviano Gómez, Enrique y Vázquez Zambrano, Donel.** *Diseño de solución informática para la gestión y control de los costos en las entidades*. Ciudad de la Habana : s.n., 2009.
32. **Kenzie, Williams.** Ezinearticles. *La Historia de las Tarjetas de Identificación*. [En línea] 2009. http://ezinearticles.com/?The-History-of-Identification-Cards&id=3765530&usg=ALkJrhgUajBrayjllkNwxlePvHWF7L6g_Q.

33. **Demarchi, Rogelio.** Lavo. *La historia de la identificación.* [En línea] <http://www.lavoz.com.ar/columna/historia-identificacion>.
34. **territoriopc.** [En línea] 2001. [Citado el: 30 de 04 de 2011.] http://www.territoriopc.com/javascript/tutorial_javascript_introduccion.php.
35. **Policia Nacional del Perú.** SCRIBD. *Sistemas de identificación Humana.* [En línea] 2011. <http://es.scribd.com/doc/35212816/Sistemas-de-Identificacion-Humana>.
36. **Pruebas de Software.** [En línea] 2010. [Citado el: 12 de 04 de 2011.] <http://lsi.ugr.es/~ig1/docis/pruso.pdf>.
37. **Lenguaje HTML.** [En línea] [Citado el: 12 de 03 de 2011.] <http://www.desarrolloweb.com/articulos/711.php>.
38. **GNU Operating System.** [En línea] 2009. [Citado el: 07 de 03 de 2011.] <http://www.gnu.org/philosophy/free-sw.html>.
39. **Extensible Markup Language (XML).** [En línea] 2003. <http://www.w3.org/XML/>.
40. **El mundo informatico.** [En línea] [Citado el: 05 de 05 de 2010.] <http://jorgesaavedra.wordpress.com/2006/08/17/patrones-grasp-craig-larman>.
41. **desarrolloweb.com.** [En línea] 2006. [Citado el: 20 de 04 de 2011.] <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.
42. **debug_mode=on.** [En línea] 2009. [Citado el: 03 de 05 de 2011.] <http://es.debugmodeon.com/articulo/el-patron-mvc>.

Bibliografía

1. **ABOUT, ILSEN y DENIS, VINCENT.** *HISTORIA DE LA IDENTIFICACION DE LAS PERSONAS.* 2011.
2. Seragro. <http://seragro.cl/?a=329>. [En línea] 2011.
3. **Fernández González, Mairelys y Zorrilla Rivera, Osley.** *Diseño e implementación del componente Ajuste al Costo del Subsistema Costos y Procesos del.* 2010.
4. Elipse.
<http://www.elipse.cl/productos/tarjetas%20credenciales/credenciales%20de%20identificacion%20codigo%20de%20barras%20proximidad%20hid%20pvc%20fotografia.html>. [En línea] 2011.
5. **Gómez Urquiza, Guillermo y Gil Martín, Ing. Manuel Alejandro.** *SISTEMA DE GESTION DE CREDENCIALES.* 2007.
6. **GORDÓN DÍAZ, NATHALI YESSENIA.** *CONTROL DE ACCESO EN LA ENTRADA DEL INSTITUTO.* Ecuador : s.n., 2009.
7. **Alphacard.** Alphacard. <http://store.alphacard.com/alpha-card-professional-id-card-software>. [En línea] 2011.
8. **XPRESS.** Advanced ID Creator. <http://www.advancedidcreator.com/>. [En línea] 2011.
9. ALTATEC. *CardFive - Software de diseño de Credenciales.* [En línea] NFIVE, 2011. <http://www.altatec.com.mx/html/index.php?module=Quotation&func=product&pid=8&product=CardFive+-+Software+de+dise%C3%B1o+de+Credenciales>.
10. CARD DATA SYSTEMS. [En línea] 2011. <http://carddatasystems.com/idworks.html>.
11. **Vega, Yanet.** *Definición del ciclo de vida del proyecto.* 2009.
12. **SCRIBD.** Herramientas CASE. [En línea] http://es.scribd.com/doc/36908565/Herramientas-Case#ad_unit=Doc_Sideboard_MediumRectangle_BTF_300x250&url=http%3A/es.scribd.com/doc/36908565/Herramientas-Case%23ad_unit%3DDoc_Sideboard_MediumRectangle_BTF_300x250%26url%3Dhttp%253A/es.scribd.com/doc/3690.
13. visual-paradigm. *visual-paradigm.* [En línea] 2005. [Citado el: 15 de 04 de 2011.] <http://www.visual-paradigm.com>.
14. Documentación del Servidor HTTP Apache 2.0. [En línea] 2009. [Citado el: 12 de 04 de 2011.] <http://httpd.apache.org/docs/2.0/es/>.

15. Subversion. [En línea] 2010. [Citado el: 23 de 04 de 2011.] <http://subversion.apache.org/>.
16. **Gómez Baryolo, Ing. Oiner, Morejón Borbón, Ing. Yoandry y García Tejo, Ing. Darien.** *ARQUITECTURA TECNOLÓGICA PARA EL DESARROLLO DE SOFTWARE.* 2011.
17. Zend Framework. [En línea] [Citado el: 10 de 02 de 2011.] <http://framework.zend.com/manual/en/>.
18. Doctrine. *Doctrine.* [En línea] [Citado el: 10 de 04 de 2011.] <http://www.doctrine-project.org>.
19. **Ramos Arias, Taimé y Torres Salas, Pedro Antonio.** *Diseño e implementación del módulo Banco del Sistema Integral de Gestión CEDRUX.* 2011.
20. **Arencibia Ramírez, Lic. Joel y Otero Dartayet, Dasiel.** *Subsistema para el manejo y reportes de credenciales.* 2007.
21. **Delgado Kios, Lic. Yeneit y Quevedo Montero, Yasel Rafael.** *Propuesta de Arquitectura para el Sistema de Gestión de Credenciales.* 2008.
22. **Fuentes, Lidia, Troya, Jose M. y Vallecillo, Antonio.** *Desarrollo de Software Basado en Componentes.* Malaga, Spain : s.n., 2007.
23. **Fernandez Leyet, Osmar.** *Documento Línea Base de proyecto-CEDRUX-1.0.* Ciudad de la Habana : s.n., 2010.
24. **Unidad de Compatibilización Integración y Desarrollo De Software para la Defensa.** *Proceso de Desarrollo y Gestión de Proyectos de Software (1ra Versión).* 2009.
25. Avicard. <http://www.avicard.com.uy/productos/identificacion>. [En línea] 2011.
26. Archivert. <http://www.archivert.cl/tarjetas-y-credenciales-colgante.htm>. [En línea] 2011.
27. **Quevedo Montero, Yasel Rafael.** *Propuesta de Arquitectura para el Sistema de Gestión de Credenciales.* 2008.
28. Diccionarios.com. [En línea] 2011. http://www.diccionarios.com/detalle.php?palabra=dispositivo&modo=4&dicc_51=on.
29. **Vallecillo, Antonio, Fuentes, Lidia y Troya, José M.** *Desarrollo de Software Basado en Componentes.* Malaga, España : s.n., 2007.
30. **Prieto, Félix.** *Patrones de diseño.* 2009.
31. **Pérez Pérez, Joisel, Chaviano Gómez, Enrique y Vázquez Zambrano, Donel.** *Diseño de solución informática para la gestión y control de los costos en las entidades.* Ciudad de la Habana : s.n., 2009.

32. **Kenzie, Williams.** Ezinearticles. *La Historia de las Tarjetas de Identificación.* [En línea] 2009. http://ezinearticles.com/?The-History-of-Identification-Cards&id=3765530&usg=ALkJrhgUajBrayjllkNwxlePvHWF7L6g_Q.
33. **Demarchi, Rogelio.** LavoZ. *La historia de la identificación.* [En línea] <http://www.lavoz.com.ar/columna/historia-identificacion>.
34. territoriopc. [En línea] 2001. [Citado el: 30 de 04 de 2011.] http://www.territoriopc.com/javascript/tutorial_javascript_introduccion.php.
35. **Policia Nacional del Perú.** SCRIBD. *Sistemas de identificación Humana.* [En línea] 2011. <http://es.scribd.com/doc/35212816/Sistemas-de-Identificacion-Humana>.
36. Pruebas de Software. [En línea] 2010. [Citado el: 12 de 04 de 2011.] <http://lsi.ugr.es/~ig1/docis/pruso.pdf>.
37. Lenguaje HTML. [En línea] [Citado el: 12 de 03 de 2011.] <http://www.desarrolloweb.com/articulos/711.php>.
38. GNU Operating System. [En línea] 2009. [Citado el: 07 de 03 de 2011.] <http://www.gnu.org/philosophy/free-sw.html>.
39. Extensible Markup Language (XML). [En línea] 2003. <http://www.w3.org/XML/>.
40. El mundo informatico. [En línea] [Citado el: 05 de 05 de 2010.] <http://jorgesaavedra.wordpress.com/2006/08/17/patrones-grasp-craig-larman>.
41. desarrolloweb.com. [En línea] 2006. [Citado el: 20 de 04 de 2011.] <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.
42. debug_mode=on. [En línea] 2009. [Citado el: 03 de 05 de 2011.] <http://es.debugmodeon.com/articulo/el-patron-mvc>.

Anexos

Anexo I. Taxonomía Arquitectónica del marco de trabajo Sauxe.

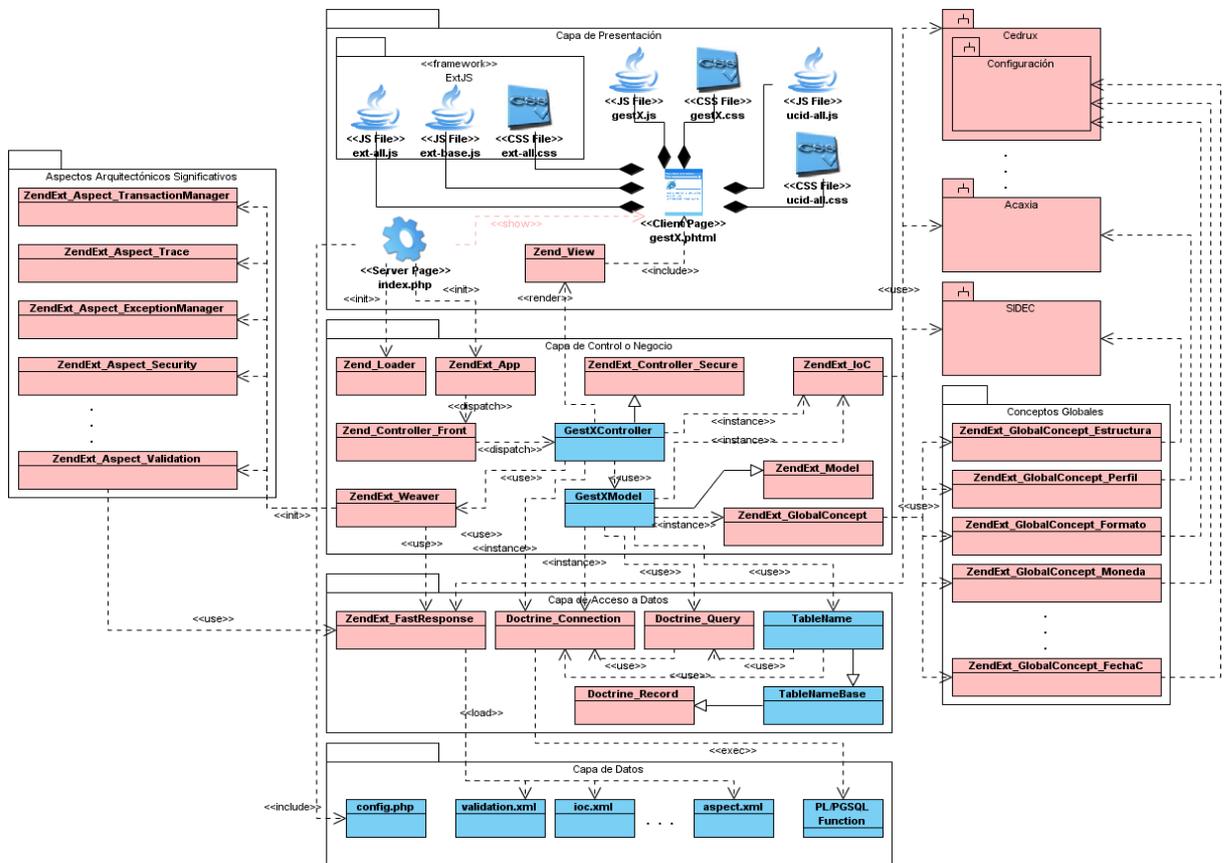


Figura 18 Taxonomía Arquitectónica

Anexo II. Modelo de datos del componente credenciales

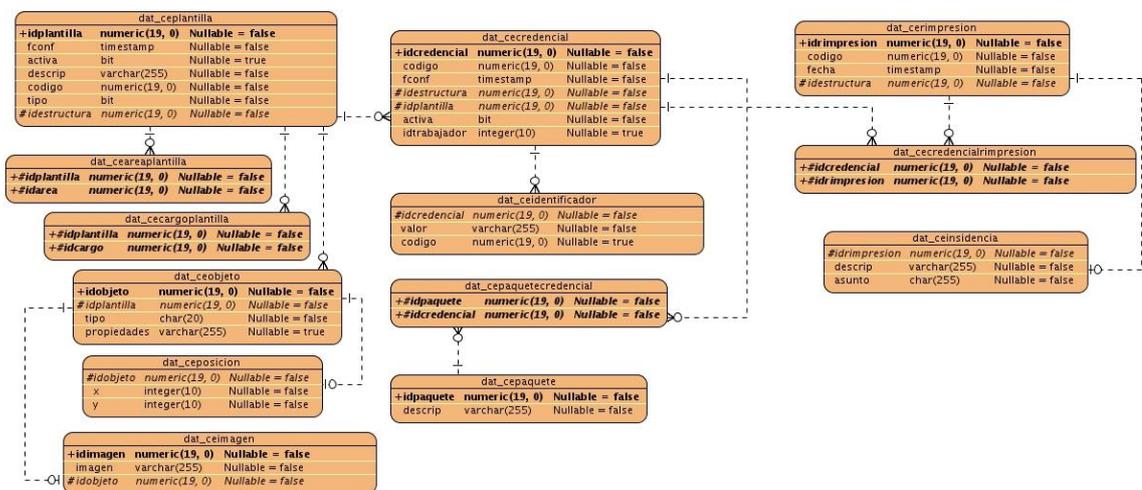


Figura 19 Modelo de datos del componente credenciales

Anexo III. Cantidad de procedimientos por clases.

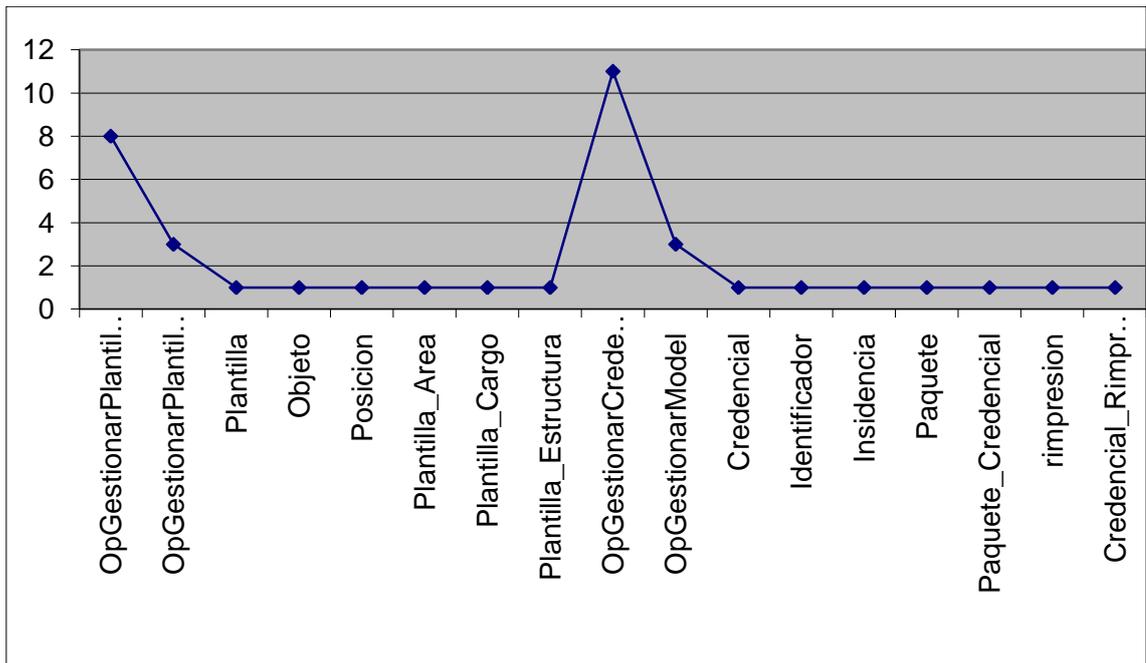


Figura 20 Cantidad de procedimientos por clases.