UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 3

MÓDULO DE AUTENTICACIÓN PARA EL SISTEMA DE CONTROL DE ACCESO ACAXIA VERSIÓN 1.1

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

Autor:

Geonel Alejandro Rama Alemán

Tutor:

Ing. René Rodrigo Bauta Camejo

Co-Tutora:

Ing. Martha Acosta Alvarez

La Habana, junio del 2015

"Año 57 de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ___ días del mes de _____ del año ____.

Autor: Geonel Alejandro Rama Alemán Tutor: Ing. René Rodrigo Bauta Camejo

_____ _____

DATOS DE CONTACTO

Autor

Nombre: Geonel Alejandro

Apellidos: Rama Alemán

Correo electrónico: garama@estudiantes.uci.cu

Tutor

Nombre: René Rodrigo

Apellidos: Bauta Camejo

Correo electrónico: rrbauta@uci.cu

Co-Tutora

Nombre: Martha

Apellidos: Acosta Alvarez

Correo electrónico: macostaa@uci.cu

AGRADECIMIENTOS

Quisiera agradecer a mi familia, la vida entera no me alcanza para terminar de agradecerles.

A Marthica, por quererme tanto y mantenerse a mi lado todos estos años pasados y los que vendrán en el futuro.

A Rubiel, a Elizabeth y a Yonnier, por convertirse en fieles compañeros de esas noches en las que se prefiere una buena taza de café y algo de literatura.

A Migue, Alién, Martha y Miguel, que ahora forman parte de mi familia y yo formo parte de la de ellos.

A mi tutor, René, por brindarme su apoyo en todo momento.

A los miembros del tribunal, por darme su voto de confianza y brindarme su apoyo.

A todo aquel que hizo posible que pudiera realizar este sueño.

RESUMEN

En la Universidad de las Ciencias Informáticas existe un sistema de control de acceso denominado ACAXIA que se encarga de gestionar la seguridad de diferentes sistemas suscritos al mismo. Este sistema solo cuenta con dos niveles de autenticación, lo que compromete la confidencialidad, la integridad y la disponibilidad de la información que protege. El presente trabajo surge con el objetivo de desarrollar la versión 1.1 del módulo de autenticación para el sistema de control de acceso ACAXIA, integrando un factor de autenticación basado en código de barras con los existentes. Para ello se realizan un conjunto de objetivos específicos. Se construye el marco teórico conceptual de la investigación. Se seleccionan y caracterizan las herramientas y tecnologías a utilizar en el proceso de desarrollo. Se definen los requisitos funcionales y no funcionales del sistema. Se realiza el análisis y diseño de la nueva versión del módulo de autenticación. Se implementa un factor de autenticación basado en código de barras y se integra a los factores ya existentes en el módulo de autenticación. Se valida el cumplimiento de las funcionalidades definidas a través de pruebas de caja blanca y caja negra. Por último, se valida el cumplimiento del objetivo mediante la realización de un experimento, el cual se combina con una encuesta a un grupo de especialistas. Finalmente, se obtiene como resultado final una nueva versión del módulo de autenticación para el sistema ACAXIA que integra un factor de autenticación basado en código de barras con los existentes.

Palabras claves: ACAXIA, autenticación, código de barras, factor de autenticación.

ABSTRACT

At the University of Information Sciences there is an access control system called ACAXIA that manages the security of different systems subscribed to it. This system has only two levels of authentication, thus compromising the confidentiality, integrity and availability of information protected. The present study was created with the objective of developing the version 1.1 of the authentication module for ACAXIA system access control, integrating a barcode based authentication factor with the existing ones. This requires a set of specific objectives to be realized. The conceptual framework of the research is built. The tools and technologies used in the development process are selected and characterized. Functional and non-functional requirements of the system are defined. The analysis and design of the new version of the authentication module is performed. A barcode based factor authentication is implemented and integrated into the existing factors in the authentication module. The compliance with the defined functionalities is validated using white-box and black box testing. Finally, achievement of the objective is validated by conducting an experiment, which is combined with a survey of a group of specialists. As a final result a new version of the authentication module for ACAXIA system that integrates a barcode based authentication factor with the existing ones is obtained.

***Keywords**: ACAXIA, authentication, barcode, factor authentication.*

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

En los últimos años las aplicaciones informáticas han alcanzado gran relevancia en el mundo empresarial, entre ellas las aplicaciones web, las cuales permiten manejar información e interactuar con ella desde cualquier parte del mundo. La información que manejan estas aplicaciones necesita ser protegida. Una de las formas en que se protege dicha información es a través del proceso de autenticación, mediante el cual se comprueba la identidad de los usuarios que intentan acceder a la misma. Para realizar esta autenticación se utilizan diferentes elementos que ayudan a confirmar su identidad. Estos elementos son conocidos como factores de autenticación, los cuales se pueden clasificar de acuerdo a sus características.

"Los principales factores de autenticación utilizados en la actualidad para comprobar la identidad de un usuario están basados en demostrar que este usuario conoce determinados datos" (Pérez San-José, 2012). Durante el proceso de autenticación, la mayoría de los sistemas informáticos solicitan cierta información que solo un usuario válido debería conocer. *"Otros factores que se pueden utilizar en los procesos de autenticación están basados en la posesión de algún dispositivo o token[1] de seguridad"* (Pérez San-José, 2012). Mediante esta forma, el sistema reconoce al dispositivo e identifica al usuario como válido. Por otra parte, algunos factores se basan en los rasgos biológicos que caracterizan al usuario. Otros factores de autenticación están basados en la forma de actuar del usuario.

Durante el proceso de autenticación se suelen combinar diferentes factores con el fin de aumentar el nivel de seguridad de la información. "*A mayor cantidad de factores involucrados en el proceso de autenticación, mayor será la seguridad de la información que se desea proteger"* (Pfleeger, 2006). *"Se recomienda utilizar al menos 3 factores de autenticación para garantizar la confiabilidad, integridad y disponibilidad de la información"* (Fernández López, 2007). En los sistemas actuales se suelen combinar hasta 5 de ellos, en dependencia de la relevancia que tenga la información que se desea proteger.

Con el avance de las tecnologías de la informática y las comunicaciones (TIC), se han desarrollado sistemas informáticos, conocidos como sistemas de control de acceso, que tienen como objetivo

[1] Término que se refiere a los dispositivos de hardware que los usuarios cargan consigo para autorizar el acceso a un servicio.

gestionar la seguridad de otras aplicaciones. Los sistemas de control de acceso permiten no solo gestionar la autenticación, sino también la autorización de los usuarios y las auditorías. Mediante la autorización, se le asignan a los usuarios ciertos permisos de acceso a la información, de acuerdo al rol que tengan definido. Mediante la auditoría se le realizan revisiones periódicas a los registros[2] de seguridad del usuario y a la información manejada por el mismo.

En la Universidad de las Ciencias Informáticas existe un sistema de control de acceso denominado ACAXIA, desarrollado por el Centro de Informatización de Entidades (CEIGE), el cual se encarga de gestionar la seguridad de los sistemas suscritos al mismo. *"En este sistema se manejan los tres procesos fundamentales del control de acceso: autenticación, autorización y auditoría"* (Gómez Baryolo, et al., 2011). El proceso de control de acceso en ACAXIA está regido por el estándar internacional SAML[3], ofreciendo la posibilidad de que se usen certificados digitales durante la gestión de la autenticación y de implementar una arquitectura *Single Sig-On*[4]*,* donde los usuarios acceden a los diferentes sistemas suscritos a ACAXIA a través de una única validación de acceso.

Actualmente, ACAXIA cuenta con la versión 1.0 del módulo de autenticación. Este módulo de autenticación permite combinar un total de 2 factores con el fin de aumentar el nivel de seguridad del proceso en sí. Estos factores están basados en el uso de usuario y contraseña y en el reconocimiento facial. "*Los factores de autenticación basados en el uso de usuario y contraseña son ampliamente aceptados por los usuarios y desarrolladores debido a su usabilidad y su bajo costo de infraestructura"* (Villalón Huerta, 2002). Sin embargo, ofrecen poca seguridad a causa de su vulnerabilidad a disímiles ataques informáticos, entre los que se pueden encontrar los ataques de fuerza bruta, la ingeniería social y la monitorización. Por otra parte, "*los factores de autenticación basados en reconocimiento facial le brindan robustez al proceso de autenticación, dotándolo de un mayor nivel de seguridad en comparación con los factores basados en usuario y contraseña"* (Suhendra, 2011). Sin embargo, el

[2] Ficheros que contienen información sobre las acciones realizadas por los usuarios y que detallan la fecha y hora en que se realizaron las mismas.

[3] Estándar abierto que define un esquema XML para el intercambio de datos de autenticación y autorización.

[4] Término en inglés que se refiere a la arquitectura de sistemas que le permite al usuario acceder a diferentes aplicaciones con una sola validación de acceso.

reconocimiento facial es susceptible a falsos positivos, lo que provoca que se acepten como usuarios legítimos otros que no lo son.

Como se mencionaba anteriormente, se considera relativamente seguro un proceso de autenticación en el que se combinen al menos 3 factores de autenticación. Luego de realizar un previo análisis de la opinión de algunos de los expertos en la materia a nivel mundial y teniendo en cuenta que ACAXIA solo cuenta con 2 factores de autenticación, se puede decir que este sistema no cumple con los estándares de seguridad establecidos para los sistemas de control de acceso. Todo esto provoca que dicho sistema no cuente con el nivel de seguridad necesario para garantizar la confidencialidad[5], integridad[6] y disponibilidad[7] de la información que se desea proteger.

Teniendo en cuenta lo expuesto anteriormente, el **problema a resolver** queda formulado por la siguiente interrogante: ¿Cómo aumentar el nivel de seguridad en el proceso de autenticación del sistema de control de acceso ACAXIA, para garantizar la confiabilidad, la integridad y la disponibilidad?

El **objeto de estudio** se encuentra enmarcado en los factores de autenticación basados en código de barras para el proceso de autenticación en sistemas web, identificándose como **campo de acción** los factores de autenticación basados en código de barras para el proceso de autenticación asociado al sistema de control de acceso ACAXIA.

Para dar solución a la problemática descrita, se plantea como **objetivo general**: Desarrollar la versión 1.1 del módulo de autenticación para el sistema de control de acceso ACAXIA, integrando un factor de autenticación basado en código de barras con los existentes. Para lograr el cumplimiento del objetivo general se plantean los siguientes objetivos específicos:

1. Construir el marco teórico conceptual de la investigación, relacionado con los métodos aplicados para el desarrollo de los factores de autenticación basados en código de barras.
2. Seleccionar y caracterizar las herramientas y tecnologías a utilizar en el proceso de desarrollo de la nueva versión del módulo de autenticación para el sistema ACAXIA.

[5] La información o los activos informáticos son accedidos solo por las personas autorizadas para hacerlo.

[6] Los activos o la información solo pueden ser modificados por las personas autorizadas y de la forma autorizada.

[7] Los activos informáticos son accedidos por las personas autorizadas en el momento requerido.

3. Definir los requisitos funcionales y no funcionales necesarios para el desarrollo de la solución.
4. Realizar el análisis y diseño de la nueva versión del módulo de autenticación para el sistema ACAXIA.
5. Implementar un factor de autenticación basado en código de barras para la versión 1.1 del módulo de autenticación del sistema ACAXIA.
6. Validar el cumplimiento de los requisitos asociados a la versión 1.1 del módulo de autenticación para el sistema ACAXIA a través de pruebas de caja blanca y caja negra.
7. Validar el cumplimiento del objetivo propuesto a través de la realización de un experimento, realizando además una encuesta a un grupo de especialistas que participan en el mismo.

Para guiar la investigación se plantea la siguiente **idea a defender**: Si se desarrolla una versión del módulo de autenticación para el sistema de control de acceso ACAXIA, en la que se integre un nuevo factor de autenticación basado en código de barras con los ya existentes, entonces se logrará aumentar el nivel de seguridad en el proceso de autenticación del sistema de control de acceso ACAXIA.

A lo largo del desarrollo de la investigación se evidencia el uso de los siguientes métodos científicos:

Métodos teóricos:

- Analítico-Sintético, para realizar un estudio en profundidad de la bibliografía especializada en cuanto a los factores de autenticación para el proceso de autenticación en sistemas web y en otros aspectos esenciales, con el fin de identificar elementos claves que contribuyen a la solución del problema planteado.
- Histórico-Lógico, el cual se emplea en la investigación para identificar el surgimiento y evolución de los factores de autenticación para el proceso de autenticación en sistemas web.
- Modelación, el cual se pone en práctica a través de la realización del análisis y diseño de la solución a través de diagramas y modelos.

Métodos empíricos:

- Entrevista, la cual se realiza al jefe del departamento de desarrollo de componentes del centro CEIGE, con el fin de recopilar toda la información necesaria respecto a su estructura y funcionamiento.
- Encuesta, la cual se le realiza a un grupo de especialistas del departamento de desarrollo de componentes del centro CEIGE para de validar el cumplimiento de objetivo planteado.

La investigación tiene como **posibles resultados** una nueva versión del módulo de autenticación para el sistema ACAXIA que integre un factor de autenticación basado en código de barras con los existentes, así como los artefactos ingenieriles asociados al desarrollo de la solución.

La investigación está estructurada en 3 capítulos, los cuales se describen a continuación:

Capítulo 1: Aborda el fundamento teórico de la presente investigación. En él se establecen conceptos asociados al dominio del problema, se analizan las soluciones existentes hasta el momento ante problemas similares y se precisan las tecnologías a utilizar para dar solución al problema.

Capítulo 2: Se realiza la descripción de la solución propuesta, para ello se efectúa el modelo conceptual, se especifican los requisitos, se describe el sistema, se establecen los patrones arquitectónicos y se realiza el modelo de diseño.

Capítulo 3: Se realiza la construcción de la solución propuesta, motivo por el cual se realiza el modelo de despliegue y el diagrama de componentes. Finalmente se realiza la validación de la solución.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DEL MÓDULO DE AUTENTICACIÓN PARA EL SISTEMA EL SISTEMA ACAXIA VERSIÓN 1.1

1.1 Introducción

En este capítulo se analizan los principales factores de autenticación y los conceptos asociados a los mismos, haciendo énfasis en los factores basados en algo poseído. Dentro de los factores basados en algo poseído se trata, de manera específica, basados en código de barras, definiendo sus principales características. Se realiza un estudio sobre soluciones similares, tanto internacionales como nacionales, identificando sus principales características, a fin de que estas puedan ser usadas como referencia en la presente investigación. Se analizan las principales características del sistema de control de acceso ACAXIA, enfatizando en cómo se realiza el proceso de autenticación dentro del mismo. También se realiza un estudio de las principales herramientas y tecnologías asociadas a dicho sistema, así como las herramientas necesarias para la realización del factor de autenticación basado en código de barras, con el fin de conformar una propuesta de solución.

1.2 Conceptos fundamentales

*"El término **seguridad informática** define el conjunto de métodos y herramientas destinados a proteger los bienes o activos informáticos de una institución"* (Pfleeger, 2006). "*Es una generalización para un conjunto de herramientas que ejecutan ciertas tareas relativas a la seguridad de los datos"* (Microsoft Corporation, 2009). Se puede decir que la seguridad informática es el conjunto de elementos que enfocan su objetivo hacia la protección de la infraestructura tecnológica de una organización determinada, centrándose principalmente en la información contenida o circulante.

*"Un **sistema de control de acceso** es una aplicación destinada a implementar factores de autenticación, que definen responsabilidades y reglas a seguir, con el fin de minimizar los efectos que puedan traer consigo las amenazas e intentar prevenir posibles ataques"* (Villalón Huerta, 2002). "*Un sistema de control de acceso maneja 3 procesos fundamentales: autenticación, autorización y auditoría"* (Suhendra, 2011). Es decir, un sistema de control de acceso constituye un conjunto de factores que se encargan de gestionar la autenticación, autorización y auditoría de usuarios en sistemas

informáticos, así como determinar los diferentes niveles de acceso de estos usuarios a la información que se maneja.

*"La **autenticación** es el proceso mediante el cual una identidad se identifica en el sistema obteniendo unas credenciales (usuario, grupo), las cuales determinan los permisos de acceso a los recursos"* (Fernández López, 2007). "*La autenticación de un objeto puede significar la confirmación de su procedencia, mientras que la autenticación de una persona a menudo consiste en verificar su identidad"* (Microsoft Corporation, 2009). Es decir, la autenticación en términos informáticos se refiere al proceso de verificación de la identidad de un usuario en un sistema determinado. Esto se realiza a través de una petición de conexión, la cual arroja como resultado si el usuario tiene o no acceso al sistema.

*"Los **factores de autenticación** son elementos que ayudan a comprobar la identidad de los usuarios durante el proceso de autenticación"* (Pérez San-José, 2012)*. "Los factores de autenticación constituyen requisitos fundamentales para los sistemas informáticos según el nivel de seguridad con que se desea proteger la información"* (Suhendra, 2011). Es decir, los factores de autenticación están asociados al control de acceso en sistemas informáticos, los cuáles contienen un conjunto de estos factores integrados, que contribuyen a la protección de la información.

1.3 Factores de autenticación

Actualmente existen 4 categorías en las que se pueden englobar los factores de autenticación, de acuerdo a lo que usan para el chequeo de la identidad:

- Por lo que se conoce (contraseña, PIN).
- Por lo que se posee (tarjeta inteligente, solapín, carnet de identidad).
- Por lo que es y que lo identifica (huellas dactilares, retina, voz).
- Por lo que es capaz de hacer (firma, patrones de escritura).

La composición de varios factores permite aumentar el nivel de seguridad en el proceso de autenticación. Por eso, cuando la información es muy importante, al punto de resultar crítica para la organización, se combinan al menos 3 factores con el fin de asegurar la confiabilidad, la integridad y la disponibilidad de la misma. Entre los factores que suelen combinarse se encuentran los basados en algo poseído.

1.3.1 Herramientas de autenticación basadas en algo poseído

"Los sistemas basados en algo poseído están compuestos generalmente por objetos llamados token de seguridad" (Pelaez González, 2011). "*Distinto a una contraseña, un token de seguridad es un objeto físico, o sea, un pequeño dispositivo de hardware que los usuarios cargan consigo para autorizar el acceso a un servicio"* (Pfleeger, 2006). El dispositivo puede ser en forma de una tarjeta inteligente o puede estar incorporado en un objeto utilizado comúnmente, como un llavero.

Los token son modelos ampliamente aceptados entre los usuarios, que pueden incorporar hardware de alta seguridad. Pueden integrarse fácilmente con otros factores de autenticación (sistemas basados en algo conocido, sistemas biométricos). Además, resulta muy fácil negarle el acceso a un usuario determinado, solo basta con retirarle su token o marcarlo como no válido para el lector. La mayoría de estos token son en forma de tarjeta debido a su facilidad de transportación y su usabilidad, aunque también se encuentran incorporados a otros objetos. Entre los dispositivos que tienen forma de tarjeta, se pueden encontrar las tarjetas inteligentes, las tarjetas de banda magnética y las tarjetas de código de barras.

1.3.2 Tarjetas inteligentes

"Las tarjetas inteligentes son tarjetas de plástico que contienen un hardware de alta tecnología, generalmente un circuito integrado, el cual puede ser de solo memoria o incluso contener todo un sistema operativo, almacenado en un microprocesador, que le permita almacenar información acerca del usuario" (Hernández Díaz, et al., 2013). La información que contienen las tarjetas inteligentes está encriptada en el sistema. Para realizar su modificación es necesario hacer uso de otro sistema operativo, el cual debe estar instalado en una computadora convencional.

"Las tarjetas inteligentes aseguran que los usuarios apropiados accedan a los datos personales que contienen. Además, unido a su fácil uso y transportación, aseguran la portabilidad y confiabilidad de la información contenida" (Rankl, et al., 2010). Estas tarjetas pueden contener información de una o varias aplicaciones al mismo tiempo gracias a que utilizan una jerarquía de almacenamiento de la información similar a la utilizada por los sistemas de ficheros convencionales.

"Debido al hardware de alta complejidad que poseen, las tarjetas inteligentes presentan un elevado costo de fabricación" (Microsoft Corporation, 2009). Esto puede ser un inconveniente para algunas empresas que necesiten adquirirlas y que no tengan suficientes recursos para ello. Además, las tarjetas

inteligentes utilizan interfaces de programación de aplicaciones específicas para los sistemas en que hacen uso de las mismas, lo que puede ser un inconveniente a la hora de desarrollar un factor de autenticación basado en su uso debido a problemas de compatibilidad de estas interfaces con el sistema ACAXIA.

1.3.3 Tarjetas magnéticas

"La tarjeta magnética es una tarjeta estándar que tiene una banda magnética, la cual se usa para almacenar una determinada cantidad de información. La tecnología usada en estas tarjetas es sumamente desarrollada y difundida a nivel mundial ya que los costes de infraestructura son bajos" (Silveiro Leyva, et al., 2013). Su uso, a pesar de ser extendido, ha quedado reducido a transferencias bancarias, ya sean operaciones de pago o extracción de dinero. Esto se debe, en gran medida, a la poca capacidad de almacenamiento que posee la banda magnética.

"A pesar de su fácil usabilidad e implementación, las tarjetas magnéticas no garantizan la confidencialidad y tampoco la integridad de los datos que contienen las mismas" (Pelaez González, 2011). La información que contienen dichas bandas puede ser leída y modificada fácilmente haciendo uso de grabadores magnéticos, lo que lo hace un factor inefectivo para integrarlo en el proceso de autenticación del sistema de control de acceso ACAXIA.

1.3.4 Tarjetas de código de barras

"Un código de barra es un sistema de codificación creado con el objetivo de identificar objetos y facilitar la obtención de información y de esta forma eliminar la posibilidad de error en la captura. La utilización de este sistema de codificación es exitosa debido a la fiabilidad que presenta en la recolección automática de datos, reduciendo los posibles errores humanos que se pueden producir en el caso de una introducción errónea de información" (Castelló Martínez, 2005). *"El código de barras presenta beneficios en cuanto a su usabilidad: es fácil producirlo y crear una infraestructura para un sistema que lo utilice, los lectores de reconocimiento de dichos códigos son muy fáciles de instalar y usar"* (Carro Paz, et al., 2010) Es por eso que los usuarios que hacen uso del mismo, se sienten cómodos usándolos. También cabe mencionar que el código de barras ofrece bajas tasas de errores en el proceso de recolección de datos, lo que permite mayor fiabilidad e integridad en la información manejada durante el proceso de autenticación.

El código de barras puede ser usado en casi cualquier actividad humana que requiera de un sistema de autenticación o identificación, ya sea en la industria, o en el comercio, e incluso en instituciones educativas, como las bibliotecas, librerías, entre otras. Se usa también para llevar el control de diversos objetos que se desee inventariar: documentos, productos del mercado, libros, entre otros. También para realizar la autenticación en sistemas informáticos, donde se suele combinar con otros factores de autenticación con el fin de lograr un proceso de autenticación fortalecido. Estas características permiten que sean las tarjetas de códigos de barras unos factores prácticos para integrar en el proceso de autenticación del sistema ACAXIA.

1.4 Funcionamiento del código de barras

"El código de barras es un arreglo que contiene información codificada en las barras y espacios del símbolo. Esta información puede ser leída por medio de dispositivos ópticos" (Carro Paz, et al., 2010). "*Es una técnica de entrada de datos, con combinaciones de barras y espacios paralelos que forman una imagen"* (Pelaez González, 2011). Esta imagen representa un número, el cual se puede leer haciendo uso de un lector óptico y luego ser desencriptado mediante un algoritmo específico. Este algoritmo convierte la información obtenida por el lector en el código asociado al usuario. Para lograr esto se utiliza un dígito de control, mediante el cual se verifica la corrección de la información. A continuación se muestra la función utilizada para el cálculo del dígito de control haciendo uso del lenguaje de programación PHP.

```php
function barcode_checksum ($codebin) {
  $checksum = 0;
  foreach (str_split(strrev($codebin)) as $pos => $val) {
    $checksum += $val * (3 - 2 * ($pos % 2));
  }
  return ((10 - ($checksum % 10)) % 10);
}
```

Figura 1 Función utilizada para el cálculo del dígito de control

Por otra parte, un código de barras está formado por 4 elementos fundamentales: módulo, barra, espacio y carácter, como se muestra en la figura 2. El módulo es la unidad básica, un conjunto de módulos forma una barra o un espacio. La barra es la parte oscura dentro del código, la cual se llama así ya que tiene forma rectangular y alargada. El espacio, sin embargo, es la parte clara, generalmente de color blanco, o del mismo color que la tarjeta que contiene al código. A la combinación de las barras

y los espacios se le llama carácter, el cual se corresponde normalmente con un carácter alfanumérico. La representación de la información en un código de barras se hace de forma binaria, por lo que a las barras se les da valor 1 y a los espacios valor 0.

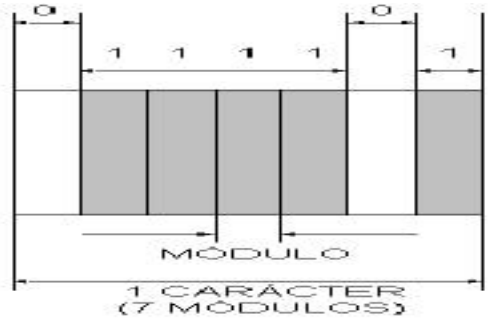


Figura 2 Representación de los elementos de un código de barras

1.5 Análisis de soluciones existentes

La autenticación por código de barras es de gran importancia para el fortalecimiento de la seguridad de sistemas informáticos ya que permite ser combinada con otros factores de autenticación y fortalecer el proceso de autenticación en sí. A raíz de esto, se han desarrollado diferentes soluciones a nivel internacional y nacional similares a la solución que se desea realizar. La mayoría de las empresas no declararan el modo en que realizan la autenticación ya que esto podría disminuir la seguridad de sus sistemas. Es por eso que solo se pueden analizar algunas de las existentes. A continuación se muestra un breve estudio de las soluciones analizadas, haciéndose énfasis en sus principales características.

1.5.1 Authy

Authy es una aplicación basada en la autenticación en 2 pasos. "*La autenticación en 2 pasos añade una capa de seguridad extra a la clásica combinación de usuario y contraseña, con el propósito de evitar que cualquier persona que adivine el usuario y contraseña pueda acceder a una cuenta de un servicio sin el código extra de autenticación"* (Gutiérrez, 2014). "*Authy utiliza la técnica: contraseña basada en el tiempo de un único uso. Mediante esta técnica se le brinda al usuario un token de seguridad que contiene un código específico y único (en el caso de Authy usa código QR[8]), el cual es leído y reconocido posteriormente por el sistema"* (M'Raihi, et al., 2011). Authy cuenta con 4

[8] Término que se refiere al código de respuesta rápida, módulo para almacenar información en una matriz de puntos o en un código de barras bidimensional.

funcionalidades básicas: adición de nuevos usuarios, verificación de los códigos de los usuarios, envío de SMS y llamadas como método alternativo a la aplicación.

Para realizar todo el proceso de autenticación con este sistema se necesita, básicamente, de una línea fija de teléfono móvil que esté registrada internacionalmente y que dicho teléfono sea compatible con el sistema operativo Android OS, además de conexión directa a Internet, ya que se necesita verificar una serie de parámetros mientras se usa la aplicación. Authy, aunque parezca del todo gratuito, no lo es. Si se necesita tener una cierta cantidad de usuarios, la cual excede un límite que impone el sistema, hay que desembolsar una suma por cada usuario que se agregue, lo que hace que la aplicación solo funcione para bases de datos pequeñas.

Esta herramienta no da solución a la problemática planteada, debido principalmente a la incompatibilidad del sistema ACAXIA con la tecnología Android y a la condición de pago que presenta la misma por agregar una cierta cantidad de usuarios. Por otra parte, se toma como referencia para el desarrollo de la solución la técnica contraseña basada en el tiempo de un único uso, donde se asigna un código de barras único por cada usuario. Esta técnica contribuye al fortalecimiento del proceso de autenticación del ACAXIA, así como al fortalecimiento de la seguridad de los sistemas suscritos al mismo.

1.5.2 TeknoTAG

"TeknoTAG es un sistema de pago que permite que una entidad gestione el control de acceso a diferentes eventos que decida crear la propia entidad" (Axesor, 2015). "*Este sistema está integrado con un control de acceso móvil mediante PDAs[9] con lector de código de Barras, código QR o RFID[10] que permiten identificar a las personas en cualquier parte del evento"* (TeknoTAG, 2015). "*El sistema TeknoTAG permite que se puedan definir roles de acceso por cada asistente, de tal forma que puedan aparecer en pantalla los datos del asistente y si tiene acceso a determinadas conferencias, talleres o comidas"* (Oryarzabal Arocena, 2015). Al definir estos roles, el sistema TeknoTAG evita el acceso de

[9] Término que se refiere a la computadora de mano, diseñada como agenda electrónica, con un sistema de reconocimiento de escritura integrado.

[10] Término que se refiere a los sistemas de identificación por radiofrecuencia que usan dispositivos denominados etiquetas, tarjetas, tags RFID.

personas no autorizadas o que no han contratado el servicio, lo que fortalece el proceso de autenticación a través del control de acceso de las personas.

Para usar este sistema es necesario instalar una red local y varios servidores de datos. La instalación de esta red se realiza únicamente a través del personal de mantenimiento de la empresa desarrolladora del sistema con el fin de supervisar el correcto funcionamiento del software y del hardware y de evitar la evasión del pago por parte de los clientes, así como la piratería. Por otra parte, a pesar de que los propios desarrolladores del sistema brindan el soporte para la instalación de dicha red, es la entidad contratante quien tiene que correr con todos los gastos de infraestructura.

Esta herramienta no da solución a la problemática planteada, debido principalmente a que es una herramienta privativa, la cual necesita de la adquisición de toda una infraestructura de instalación y la contratación del servicio técnico de la propia empresa desarrolladora. Por otra parte, se toma como referencia para el desarrollo de la solución el uso de roles de acceso, donde se asignan roles de acceso específicos a cada usuario. Esto contribuye al fortalecimiento del proceso de autenticación del ACAXIA evitando el acceso de personas no autorizadas a los sistemas suscritos al mismo.

1.5.3 Subsistema de Control de Acceso a los Comedores de la Universidad de las Ciencias Informáticas

"El Subsistema de Control de Acceso a los Comedores de la Universidad de las Ciencias Informáticas (SCAC). Tiene como objetivo principal controlar el acceso a los comedores por parte de los trabajadores y estudiantes de la universidad" (Martínez Romero, et al., 2012). Está basado en tecnologías libres, usando Java como lenguaje de programación y PostgreSQL como gestor de bases de datos. En este sistema, la autenticación se realiza a través del uso de código de barras.

Este sistema es robusto, pero tiene como principal inconveniente que está desarrollado completamente sobre el lenguaje Java, por lo que resultaría costoso adaptarlo al sistema ACAXIA. Además de esto, el sistema utiliza el mapeador de bases de datos y objetos relacionales Hibernate, el cual es incompatible con la tecnología usada para realizar el proceso de acceso a datos del sistema ACAXIA, que usa Doctrine para realizar ese proceso. Por otra parte, se utiliza la investigación realizada por los investigadores de dicha solución, así como la bibliografía utilizada, como material de apoyo a la presente investigación.

1.5.4 Resultados del análisis

Luego de haber realizado el análisis de las soluciones existentes a nivel nacional e internacional, se tiene como resultado:

- Ninguna de las soluciones analizadas brinda una solución factible al problema planteado, principalmente por las condiciones de pago de algunas de ellas, y por la incompatibilidad con las tecnologías del sistema ACAXIA.
- Se decide tomar algunas de las características de estos sistemas, entre las que se encuentran la técnica contraseña basada en el tiempo de un único uso y uso de roles de acceso.
- Se decide utilizar la investigación previa de algunas de las soluciones estudiadas, así como la bibliografía consultada para su estudio, como material de apoyo a la presente investigación.

1.6 Sistema de control de acceso ACAXIA

"ACAXIA controla la seguridad y para ello monitoriza las acciones que se realizan. Para facilitar este proceso, se crean roles a partir de la estructura previamente creada de los sistemas, asignándoles los permisos a los diferentes usuarios. Estos permisos se realizan a diferentes niveles de la estructura" (Pelaez González, 2011). ACAXIA beneficia en gran medida al proceso de autenticación, dotándolo de una mayor fortaleza a la hora de gestionar la seguridad de dichas funcionalidades. También brinda la posibilidad de agregarle el uso de roles de acceso a la solución que se desea realizar.

"ACAXIA controla el acceso a los servicios web que se gestionan en los sistemas suscritos a él, asegurando la comunicación que existe entre los principales componentes" (Gómez Baryolo, et al., 2011). Uno de los aspectos más importantes con que cuenta ACAXIA es el control de acceso a las conexiones que se establecen entre los sistemas y los servidores, bases de datos y otras estructuras, logrando así que no se almacenen las configuraciones de las conexiones en ficheros que puedan ser objeto de ataques, lo que se traduce en un proceso de autenticación fortalecido. Esta característica posibilita una mayor abstracción a la hora de desarrollar la solución, permitiendo el enfoque de la investigación solo en el proceso de autenticación.

1.6.1 Módulo de autenticación

ACAXIA, como se mencionaba con anterioridad, es un sistema de control de acceso que se encarga de gestionar la seguridad de los sistemas suscritos al mismo a través de 3 procesos fundamentales: autenticación, autorización y auditoría. Para ello utiliza 3 módulos diferentes, cada uno asociado a uno de los procesos. Es por eso que el proceso de autenticación en ACAXIA se controla a partir del módulo

de autenticación que tiene implementado este sistema de control de acceso. La versión existente del módulo cuenta con 2 factores de autenticación. Según los expertos en el tema, la combinación de estos factores no garantiza la confiabilidad, la integridad y la disponibilidad de la información que se maneja en los diferentes sistemas suscritos a ACAXIA.

1.6.2 Nivel de seguridad del módulo

Para garantizar la seguridad de la información que se maneja en los diferentes sistemas suscritos a ACAXIA, se hace necesario garantizar a su vez la confiabilidad, la integridad y la disponibilidad de esta información. Es por ello que se pretende aumentar el nivel de seguridad del proceso de autenticación. Este nivel de seguridad está dado por 3 variables fundamentales: nivel de confiabilidad de la información, nivel de integridad de la información y nivel de disponibilidad de la información. Estas variables se comprueban posteriormente durante el proceso de validación de la solución.

1.7 Metodología

"El Proceso Unificado Ágil (AUP, por sus siglas en inglés) es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP" (Rodríguez Sánchez, 2014). Esta metodología aplica las técnicas más comunes del desarrollo ágil:

- Desarrollo dirigido por pruebas.
- Modelado ágil.
- Gestión de cambios ágil.
- Refactorización de la base de datos.

El AUP, al estar basado en RUP, establece 4 fases fundamentales:

- Inicio, durante la cual se obtienen los nexos comunes entre el cliente y el equipo de desarrollo.
- Elaboración, durante la cual el equipo de desarrollo define los requisitos que va a tener el sistema.
- Construcción, durante la cual se desarrolla el sistema y se le realizan pruebas a nivel de equipo de desarrollo.

- Transición, en esta se le aplican las pruebas de validación y aceptación en un entorno de preproducción y finalmente se despliega el sistema.

"En la Universidad de las Ciencias informáticas se ha desarrollado una variación de la metodología AUP, apoyándose en el modelo CMMI-DEV en su versión 1.3, con el fin de que la misma se utilice en todos los centros productivos de la universidad" (Rodríguez Sánchez, 2014). Esta variación propone que, a partir de las 4 fases que propone la metodología AUP, se mantenga la fase de Inicio, aunque a la misma se le realizan variaciones en su objetivo. El resto de las fases se agrupan en una sola, llamada Ejecución. Por último, se agrega una nueva fase a la metodología llamada Cierre:

- Inicio, se llevan a cabo las actividades relacionadas con la planeación del proyecto (estudios acerca del cliente, estimaciones de tiempo, esfuerzo y costo).
- Ejecución, se ejecutan las actividades requeridas para desarrollar el software (modelado del negocio, captura de requisitos, arquitectura y modelo de diseño, implementación y liberación).
- Cierre, se analizan los resultados del proyecto y su ejecución, realizando a su vez las actividades relacionadas con el cierre del proyecto.

En el departamento de tecnología del centro CEIGE han decidido utilizar esta metodología, propuesta por la universidad para los centros productivos, como metodología de desarrollo. Es por eso que, para dar solución a la presente investigación y con el fin de apoyar el proceso de desarrollo, se decide utilizar dicha variación como guía de desarrollo de la presente investigación. Por otra parte, no es necesario el desarrollo de las 3 fases que propone la variación de la metodología, por lo que se decide centralizar el proceso de desarrollo en la fase de Ejecución.

1.8 Herramientas y tecnologías

En la actualidad, existen diferentes herramientas y tecnologías para desarrollar las aplicaciones informáticas, las cuales han evolucionado con el tiempo, siendo cada vez más potentes. Ellas se han especializado en diferentes plataformas, brindándole versatilidad al proceso de desarrollo de software. A continuación se describen las herramientas y tecnologías que se usan en el desarrollo de la solución.

1.8.1 Tecnologías asociadas al sistema ACAXIA

El sistema ACAXIA cuenta con un conjunto de tecnologías ya definidas por el departamento de tecnologías del centro CEIGE. El equipo de desarrollo, siguiendo las políticas de desarrollo de dicho

departamento, decide usarlas, por lo que no se hace necesario un análisis profundo de las tecnologías para el desarrollo de la solución, aunque se realiza una caracterización de ellas. Estas características quedan reflejadas a continuación.

Para el desarrollo de las clases de negocio se utiliza el marco de trabajo **Zend Framework,** en su versión 1.11. "*Zend Framework es una plataforma de desarrollo orientada a objetos basada en el lenguaje PHP, la cual está integrada por diferentes módulos y componentes que facilitan el desarrollo de aplicaciones basadas en tecnología web"* (Pelaez González, 2011). El módulo que se encarga de la gestión de la seguridad, el cual está integrado dentro de este marco de trabajo, se denomina Zend_Auth. El mismo provee una interfaz que permite configurar y personalizar los factores de autenticación.

Para el desarrollo de las clases de acceso a datos y objetos relacionales se utiliza el marco de trabajo **Doctrine** en su versión 2.0. "*Doctrine es un sistema mapeador de bases de objetos relacionales (ORM, por sus siglas en inglés) para el lenguaje PHP. Brinda la posibilidad de exportar las tablas de una base de datos a sus clases correspondientes, así como hacer el proceso inverso"* (Pelaez González, 2011). "*Para trabajar haciendo uso del mismo, es necesario crear el modelo en la sintaxis específica, lo que permite generar luego toda la base de datos"* (Fumero González, et al., 2013). Doctrine permite que se le realicen consultas a la base de datos, sirviendo como enlace entre las clases de negocio y la base de datos de objetos relacionales asociada a esas clases. Doctrine se utiliza en la solución específicamente para el trabajo con las clases de acceso a datos relacionadas con los usuarios y los roles.

Para el trabajo con las interfaces gráficas de usuario se utiliza el marco de trabajo **ExtJs** en su versión 2.2. "*ExtJs es un marco de trabajo basado en el lenguaje JavaScript para el desarrollo, del lado del cliente, de aplicaciones basadas en tecnologías web. Permite la creación, de forma fácil y rápida de interfaces gráficas para el usuario. ExtJs es adaptable a múltiples plataformas de desarrollo y su código puede ser reutilizado en varios sistemas"* (Fumero González, et al., 2013). "*Este marco de trabajo incluye licencias de código abierto (Open Source), además de licencias comerciales"* (Pelaez González, 2011). Para la conexión de las interfaces de usuario creadas con este marco de trabajo y las clases de negocio se utiliza el módulo Zend_Ext, el cual se encuentra integrado al marco de trabajo Zend Framework.

Por otra parte, se utiliza el lenguaje de programación **PHP** en su versión 5.2 para el desarrollo con los marcos de trabajo Zend Framework y Doctrine. *"PHP es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo Web y que puede ser incrustado en código HTML"* (PHP, 2009). "*Su código se ejecuta en el servidor, aunque el cliente solo recibe, resultante de esa ejecución, una página web codificada en el lenguaje HTML"* (Acosta Alvarez, 2014). Esto permite que haciendo uso de este lenguaje en los diferentes marcos de trabajos asociados al mismo, se pueda acceder a la información existente en la base de datos, así como gestionar los protocolos de acceso y servicios de autenticación.

Para el desarrollo con el marco de trabajo ExtJs se utiliza el lenguaje de programación **JavaScript** en su versión 1.5. *"JavaScript es un lenguaje interpretado. Su código no se compila, se ejecuta a través de un intérprete. JavaScript es utilizado mayormente para el desarrollo de páginas Web, ya que facilita el trabajo con la programación del lado del cliente. Los navegadores modernos interpretan el código JavaScript"* (Mozilla Developer Network, 2015). JavaScript permite, además de desarrollar las interfaces de usuario, manejar las excepciones que se generan en las mismas.

1.8.2 Herramientas de desarrollo asociadas al sistema ACAXIA

El sistema ACAXIA cuenta con un conjunto de herramientas ya definidas por el departamento de tecnologías del centro CEIGE. El equipo de desarrollo, siguiendo las políticas de desarrollo de dicho departamento, decide usarlas, por lo que no se hace necesario un análisis profundo en cuanto a herramientas para el desarrollo de la solución. Sin embargo, se realiza una caracterización de ellas. Estas características quedan reflejadas a continuación.

Para el diseño de los diferentes diagramas, generados en la etapa de modelado de negocio, se utiliza la herramienta **Visual Paradigm For UML** en su versión 8.0. *"Visual Paradigm For UML es una herramienta CASE que utiliza UML como lenguaje de modelado. Está diseñada para una amplia gama de usuarios interesados en construir sistemas fiables con el uso del paradigma orientado a objetos, incluyendo actividades como ingeniería de software, análisis de sistemas y análisis de negocios"* (Jacobson, et al.). La utilización de esta herramienta ofrece un conjunto de ventajas que hace más factible el trabajo al equipo de desarrollo: generación de código y de base de datos, transformación de diagramas de entidad-relación en tablas de base de datos. Por otra parte, la Universidad de las Ciencias Informáticas cuenta con la licencia para el uso de este software por lo cual está dentro de las

tecnologías definidas por la dirección de producción. Además, el equipo de desarrollo posee experiencia en el uso de esta herramienta.

Para la gestión de la base de datos asociada a la solución se utiliza el sistema gestor de bases de datos **PostgreSQL** en su versión 9.1. "*PostgreSQL es un sistema de gestión de bases de datos relacionales que tiene su código disponible libremente, por lo que se le pueden hacer modificaciones al mismo con tal de mejorarlo o adaptarlo a las necesidades de cualquier organización"* (PosgreSQL, 2009). PostgreSQL usa multiprocesos en vez de multihilos, lo que garantiza una mayor estabilidad en la solución ya que si ocurre un fallo durante el proceso de autenticación, este no afecta al resto de los procesos que se estén ejecutando.

Para el trabajo con el código fuente de la solución se utiliza la herramienta **Netbeans** en su versión 8.0. "*Netbeans es un entorno de desarrollo integrado que se utiliza para desarrollar aplicaciones basadas en diferentes lenguajes de programación. Netbeans cuenta con un repositorio de bibliotecas, herramientas y tecnologías, mediante las cuales se puede configurar esta herramienta para facilitar el trabajo de cualquier desarrollador"* (IBM, 2015)*. "Netbeans cuenta con soporte para múltiples lenguajes, entre ellos destacan Java, C++, PHP, HTML5, JavaScript, Python y Ruby"* (Acosta Alvarez, 2014). Una de las características más importantes que tiene esta herramienta es que para desarrollar en un proyecto no se necesita pagar por una licencia. Además de esto, Netbeans es de código abierto, permitiendo que pueda ser modificado, por lo que los desarrolladores pueden realizar las mejoras que consideren necesarias para el desarrollo de la solución.

Como servidor web se utiliza la herramienta **Apache** en su versión 2.0. "*Apache es un servidor HTTP multiplataforma de código abierto que se utiliza para ejecutar aplicaciones basadas en tecnologías web y colgarlas en la red. El mismo brinda un alto nivel de seguridad a los sistemas que se ejecutan sobre el mismo"* (Apache Software Foundation, 2014). Apache permite la virtualización del sistema ACAXIA a través de protocolos de puertos virtuales, lo que permite que se pueda acceder a dicho sistema haciendo uso de una red local o de la propia Internet.

1.9 Conclusiones parciales

Al término del presente capítulo se completa la etapa de investigación, arrojando como resultado la necesidad de desarrollar una versión del módulo de autenticación, en la cual se debe implementar un nuevo factor de autenticación basado en código de barras con el fin de integrarlo con los existentes.

Para ello se construye el marco teórico conceptual de la investigación, relacionado con los métodos aplicados al desarrollo de los factores de autenticación basados en código de barras. Por último se seleccionan y caracterizan las herramientas y tecnologías a utilizar en el proceso de desarrollo de la nueva versión del módulo de autenticación para el sistema ACAXIA, lo que permite tener claro cuales se van a utilizar.

CAPÍTULO 2. DESCRIPCIÓN DEL MÓDULO DE AUTENTICACIÓN PARA EL SISTEMA ACAXIA VERSIÓN 1.1

2.1 Introducción

En el presente capítulo se abordan aspectos que inciden directamente en el desarrollo de una solución al problema que se desea resolver. Para resolver dicho problema, se generan los artefactos que indica la metodología AUP, en su variación para la Universidad de las Ciencias Informáticas, la cual fue escogida en el capítulo anterior para guiar el proceso de desarrollo. Se elabora un modelo de la solución apoyado en la captura y posterior especificación de los requisitos funcionales y no funcionales. Se define la arquitectura, así como los patrones de diseño. Se realiza el modelo de diseño del sistema.

2.2 Modelo conceptual

"El modelo conceptual muestra (a los modeladores) clases conceptuales significativas en un dominio del problema; es el artefacto más importante que se crea durante el análisis orientado a objetos (…) es una representación de las clases conceptuales del mundo real, no de componentes de software. No se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades" (Larman, 1999). Se puede decir que el modelo conceptual representa visualmente conceptos de interés, los cuales están relacionados a objetos que son de gran importancia para el negocio. Este modelo es fundamental para comprender el dominio del problema, así como para establecer nexos comunes entre los conceptos asociados al mismo.

2.2.1 Glosario de términos del modelo conceptual

ACAXIA: Es el sistema de control de acceso. Se encarga de controlar la seguridad y el acceso de los usuarios a los diferentes sistemas suscritos al mismo.

Acción: Es la actividad que realiza el usuario para interactuar con un sistema, la cual devuelve un resultado determinado.

Funcionalidad: Es la representación de un conjunto de acciones asociadas a un sistema determinado.

Rol: Un rol representa a un grupo determinado de permisos y privilegios que puede tener un usuario al interactuar con un sistema.

Sistema: Producto suscrito al sistema de control de acceso ACAXIA para que este le brinde seguridad.

Usuario: Es la persona que interactúa directamente con uno o varios sistemas y desempeña uno o varios roles dentro de estos.

2.2.2 Diagrama de clases del modelo conceptual

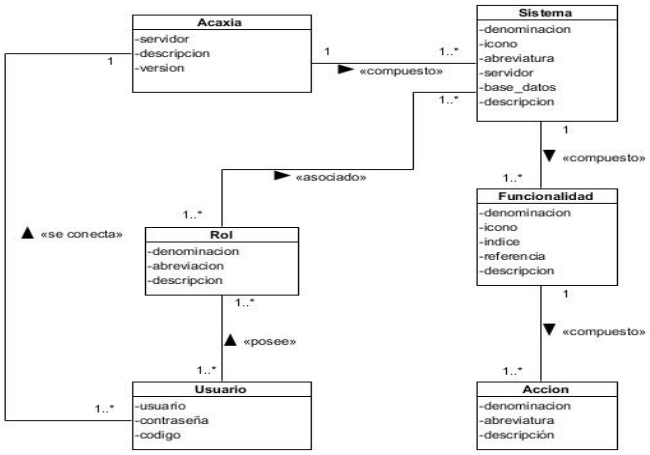


Figura 3 Diagrama de clases del modelo conceptual

2.2.3 Análisis del modelo conceptual

Un usuario se conecta a través de ACAXIA a los diferentes sistemas suscritos al mismo. Este usuario posee uno o varios roles, los cuales están asociados a los diferentes sistemas a los que ACAXIA gestiona la seguridad. El rol le permite al usuario acceder a las diferentes funcionalidades que componen a dichos sistemas y a las acciones que componen a dichas funcionalidades. Esto permite que un usuario solo pueda realizar las acciones correspondientes a los roles que tiene asignado.

2.3 Requisitos del sistema

"Los requisitos son capacidades y condiciones con las cuales debe ser conforme el sistema" (Larman, 1999). "*Los requisitos ayudan a comprender mejor el problema gracias a que poseen un conjunto de tareas orientadas a comprender lo que desea el cliente, así como la interacción entre los usuarios y el software"* (Pressman, 2010). "*Los requisitos del sistema permiten que se puedan negociar soluciones razonables, a partir de una previa comprensión de las necesidades reales del cliente, así como de un*

previo análisis de la factibilidad de los mismos" (Sommerville, 2009). Por otra parte, para una correcta obtención de los requisitos del sistema, es necesario el uso de métodos que tienen la finalidad de facilitar la comunicación con los clientes, usuarios y demás personas que puedan ayudar a comprender el funcionamiento del sistema. En, la presente investigación se emplea el método entrevista para la obtención de los requisitos necesarios para dar solución al problema.

"La entrevista es una conversación planificada entre el investigador y el entrevistado para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado" (León Hernández, et al., 2011). La entrevista suele ser empleada cuando el problema de estudio no se puede observar o es muy difícil hacerlo. Para la realización del proceso de obtención de requisitos, se decide realizar una entrevista abierta al jefe de departamento de tecnologías del centro de desarrollo CEIGE, el cual está directamente relacionado con el sistema de control de acceso ACAXIA y cuenta con la preparación necesaria para responder de forma acertada todas las preguntas.

2.3.1 Requisitos funcionales de la aplicación

"Los requisitos funcionales (RF) son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar ante entradas particulares y de cómo se debe comportar en situaciones particulares" (Sommerville, 2009). Los mismos pueden ser definidos como un conjunto de condiciones que debe cumplir la solución a desarrollar con el fin de satisfacer las necesidades de los clientes y usuarios finales. De todo el módulo de autenticación, los únicos requisitos que se modifican o agregan, son los referentes al factor de autenticación por código de barras. A continuación se muestra el listado de los requisitos funcionales obtenidos durante el proceso de captura de requisitos. Para un mayor entendimiento, estos se dividieron en 3 grupos:

- Gestionar autenticación:
  - RF 1.1. Autenticar por código de barras.
- Gestionar usuario:
  - RF 2.1. Insertar usuario.
  - RF 2.2. Modificar usuario.
  - RF 2.3. Listar usuario.
- Gestionar token:
  - RF 3.1. Generar token.

RF 3.2. Exportar token.

De los 6 requisitos identificados, existen 1 que tiene prioridad alta para el cliente y 5 que tiene prioridad media. A continuación se muestra la descripción de los requisitos mencionados anteriormente. Esta descripción se encuentra reflejada en el expediente de proyecto de la presente investigación.

Tabla 1 Descripción del requisito Autenticar por código de barras

AUTENTICAR POR CÓDIGO DE BARRAS	
Número: 1.1	**Nombre del requisito:** Autenticar por código de barras
Programador: Geonel Alejandro Rama Alemán	**Iteración Asignada:** 1
Prioridad: Alta	**Tiempo Estimado:** 72 horas
Riesgo en Desarrollo: N/A	**Tiempo Real:** 2 semanas
Descripción: Permite al usuario autenticarse en el sistema haciendo uso de un código de barras previamente asignado. Este requisito tiene como condición previa que el usuario, antes de autenticarse por código de barras, debe estar registrado en el sistema y tener asignado un código. Además de esto, debe estar seleccionado este tipo de autenticación y el usuario tiene que haberse autenticado previamente, de forma satisfactoria, haciendo uso de usuario y contraseña. Una vez autenticado mediante código de barras, el sistema permite, en caso de estar activada también la autenticación por reconocimiento facial, que el usuario se autentique haciendo uso del mismo y en caso de que no esté activada, permite la entrada a dicho sistema, pudiendo interactuar con las diferentes aplicaciones suscritas al mismo.	
Observaciones: N/A	
Prototipo de interfaz:	

Figura 4 Prototipo de interfaz de usuario del requisito Autenticar por código de barras

Tabla 2 Descripción del requisito Insertar usuario

INSERTAR USUARIO	
Número: 2.1	**Nombre del requisito:** Insertar usuario
Programador: Geonel Alejandro Rama Alemán	**Iteración Asignada:** 2
Prioridad: Media	**Tiempo Estimado:** 12 horas
Riesgo en Desarrollo: N/A	**Tiempo Real:** 1 semana

Descripción:

Permite al usuario administrador insertar un usuario determinado en el sistema y para ello debe rellenar los siguientes campos:

- Rango IP
- Tipo de escritorio (*)
- Idioma (*)
- Tema (*)
- Dominio (*)
- Entidad (*)
- Área
- Cargo

- Servidores
- Usuario (*)
- Contraseña (*)
- Código de barras

Observaciones:

- En el caso de los campos marcados con (*), son obligatorios para lograr la inserción del usuario de forma satisfactoria.

Prototipo de interfaz:

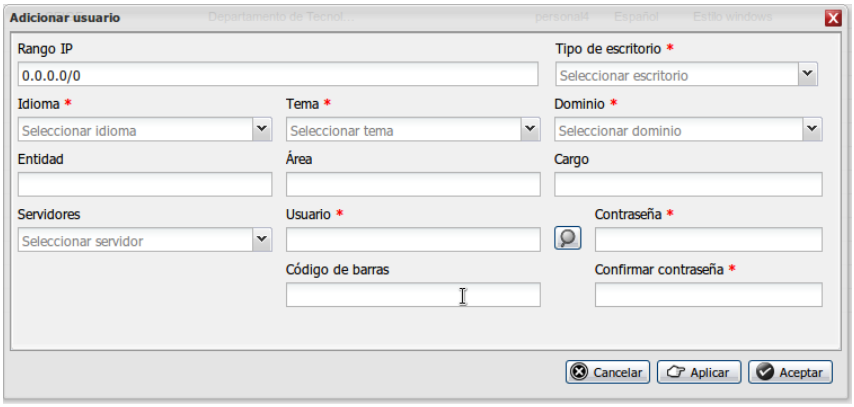


Figura 5 Prototipo de interfaz de usuario del requisito Insertar usuario

Tabla 3 Descripción del requisito de software Modificar usuario

MODIFICAR USUARIO	
Número: 2.2	**Nombre del requisito:** Modificar usuario
Programador: Geonel Alejandro Rama Alemán	**Iteración Asignada:** 2
Prioridad: Media	**Tiempo Estimado:** 12 horas
Riesgo en Desarrollo: N/A	**Tiempo Real:** 1 semana

Descripción:

Permite al usuario administrador modificar los datos de un usuario, previamente registrado en el sistema, para ello debe de rellenar los siguientes campos:

- Rango IP
- Tipo de escritorio (*)

- Idioma (*)
- Tema (*)
- Dominio (*)
- Entidad (*)
- Área
- Cargo
- Servidores
- Usuario (*)
- Contraseña (*)
- Código de barras

Observaciones:

- En el caso de los campos marcados con (*), son obligatorios para lograr la modificación del usuario de forma satisfactoria.

Prototipo de interfaz:

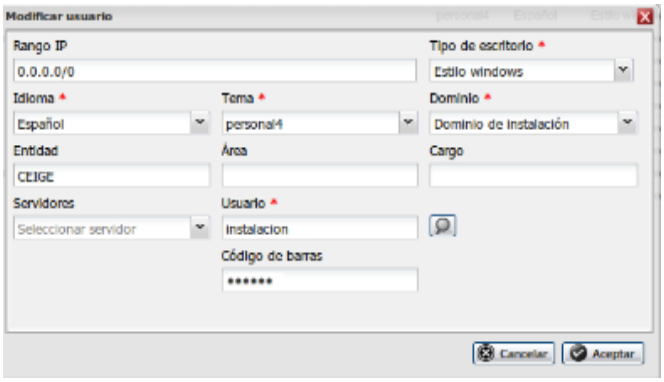


Figura 6 Prototipo de interfaz de usuario del requisito Modificar usuario

Tabla 4 Descripción del requisito Listar usuario

LISTAR USUARIO	
Número: 2.3	**Nombre del requisito:** Listar usuario
Programador: Geonel Alejandro Rama Alemán	**Iteración Asignada:** 2
Prioridad: Media	**Tiempo Estimado:** 12 horas

Riesgo en Desarrollo: N/A	**Tiempo Real:** 1 semana

Descripción:

Permite al usuario administrador listar los datos de los usuarios, previamente registrados en el sistema, mostrando en pantallas los siguientes campos:

- Usuario
- Dominio
- Entidad
- Área
- Cargo
- Tema
- Idioma
- Escritorio
- Código (*)
- Activo (**)

Observaciones:

- En el caso del campo marcado con (*), se refiere al código de barras.
- En el caso del campo marcado con (**), se refiere a si el usuario está activo o no.

Prototipo de interfaz:

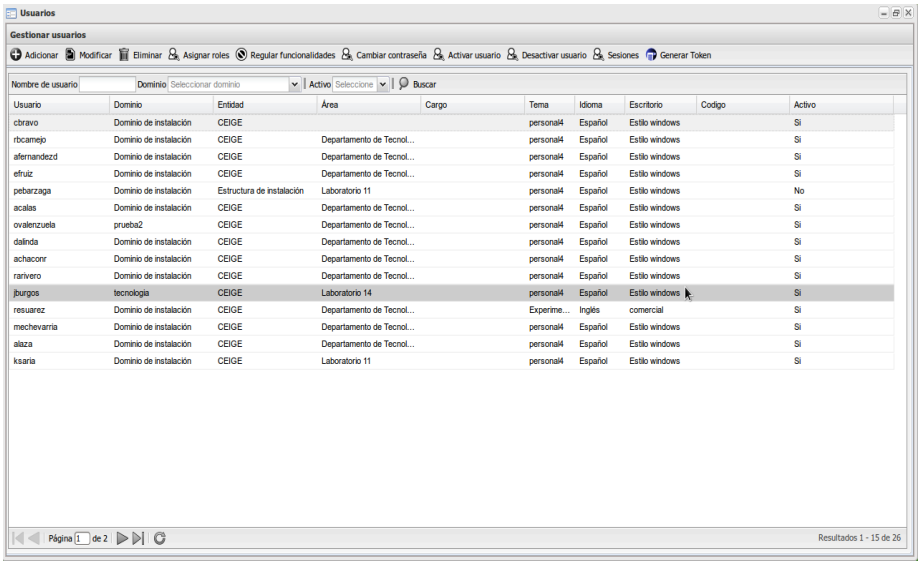


Figura 7 Prototipo de interfaz de usuario del requisito Listar usuario

Tabla 5 Descripción del requisito Generar token

GENERAR TOKEN	
Número: 3.1	**Nombre del requisito:** Generar token
Programador: Geonel Alejandro Rama Alemán	**Iteración Asignada:** 3
Prioridad: Media	**Tiempo Estimado:** 12 horas
Riesgo en Desarrollo: N/A	**Tiempo Real:** 1 semana

Descripción:

El sistema genera un token de usuario y lo guarda en un directorio predeterminado (*) a partir de diferentes datos de un usuario determinado, el cual contiene los siguientes datos del usuario:

- Usuario
- Entidad
- Área
- Cargo
- Código de barras

El usuario debe estar previamente seleccionado en el formulario Listar usuario.

Observaciones:

- (*) El directorio donde se guardan los tokens de usuario es «*web/seguridad/tokens/*».

Prototipo de interfaz:

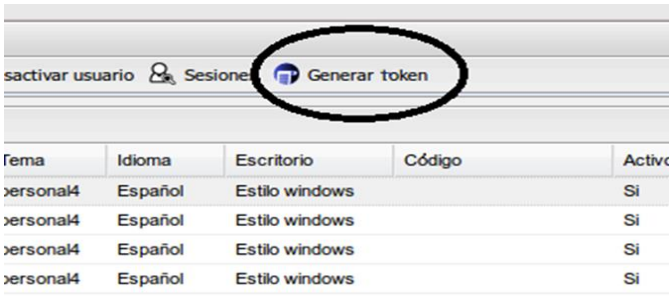


Figura 8 Prototipo de interfaz de usuario del requisito Generar token

Tabla 6 Descripción del requisito Exportar token

EXPORTAR TOKEN	
Número: 3.2	**Nombre del requisito:** Exportar token
Programador: Geonel Alejandro Rama Alemán	**Iteración Asignada:** 3
Prioridad: Media	**Tiempo Estimado:** 12 horas
Riesgo en Desarrollo: N/A	**Tiempo Real:** 1 semana

Descripción:

El sistema muestra en una ventana el token previamente generado en el requisito Generar token (*), permitiendo que se realicen las siguientes operaciones sobre el mismo:

- Guardarlo en formato .pdf
- Imprimirlo en formato papel

Observaciones:

- (*) El directorio donde se guardan previamente los tokens de usuario es «*web/seguridad/tokens/*».

Prototipo de interfaz:

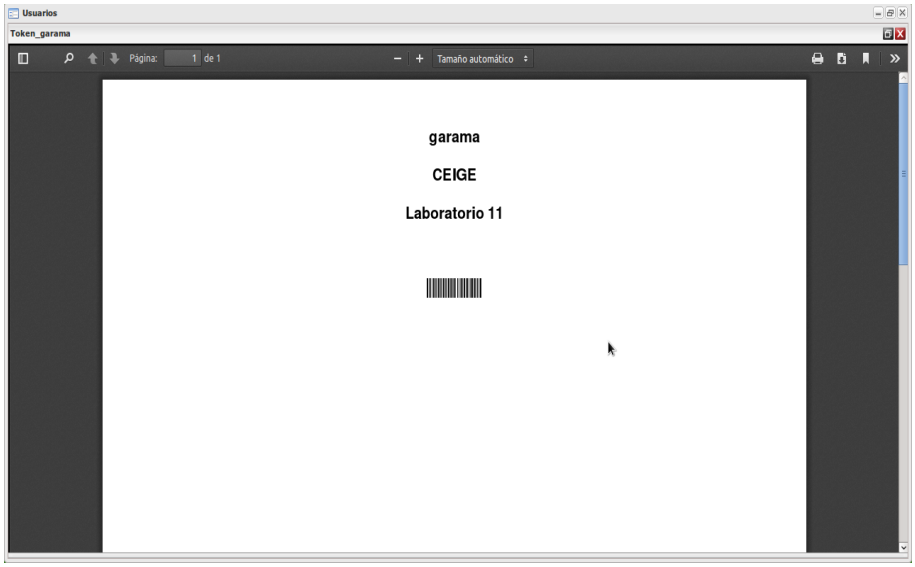


Figura 9 Prototipo de interfaz de usuario del requisito Imprimir token

2.3.2 Requisitos no funcionales de la aplicación

Los requisitos no funcionales (RNF) "*son aquellos requerimientos que no se refieren a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento"* (Sommerville, 2009). Se pueden definir como un conjunto de normativas elaboradas con el fin de lograr la usabilidad, fiabilidad y eficiencia del producto que se desea desarrollar. A continuación se describen los RNF que debe cumplir la nueva versión del módulo de autenticación:

RNF 1. Usabilidad

El módulo de autenticación podrá ser usado por personas con conocimientos mínimos en el manejo de computadoras.

RNF 2. Fiabilidad

El módulo de autenticación permitirá que solo accedan al sistema y sus funcionalidades las personas autorizadas.

RNF 3. Eficiencia

El tiempo de respuesta del módulo estará dado por la cantidad de información a procesar para lograr el proceso de autenticación. A mayor cantidad de información en la base de datos, mayor será el tiempo de procesamiento.

RNF 4. Restricciones de diseño

El módulo de autenticación debe diseñarse sobre la base de una arquitectura modelo-vista-controlador, usada en el sistema de control de acceso ACAXIA.

RNF 5. Interfaces de usuario

El módulo debe tener un diseño gráfico de las interfaces basado, fundamentalmente, en iconos.

RNF 6. Interfaces de software

El módulo debe funcionar sobre:

- Navegador web Firefox 4.0 o superior.
- Sistema operativo GNU/Linux.
- Servidor web Apache 2.0 o superior.

RNF 7. Interfaces de hardware

Las computadoras desde donde los usuarios van a acceder al sistema deben tener como mínimo una tarjeta de red, al menos 512 MB de memoria RAM y un procesador Pentium IV o

equivalente con velocidad de 1 GHz; además de los periféricos (teclado, mouse, monitor, impresora) necesarios para interactuar con el sistema. La computadora usada como servidor debe de contar como mínimo con una tarjeta de red, al menos 2 GB de memoria RAM y un procesador Dual Core o equivalente con velocidad de 1.5 GHz en cada núcleo.

2.3.3 Técnicas de validación de requisitos

"Los requisitos previamente definidos necesitan ser validados, asegurando con esto que todo el análisis realizado y los resultados obtenidos durante la definición de los mismos, resultan correctos" (Sommerville, 2009). Este proceso debe lograrse con el fin de evitar la implementación de una mala especificación. Para evitar esto, se utilizan las técnicas de validación que se describen a continuación.

Revisiones de requisitos: Mediante esta técnica se analizan los requisitos de forma sistemática (Sommerville, 2009). Esta actividad es realizada por un equipo de revisores. Para aplicar esta técnica se le realizan revisiones técnicas formales a las descripciones de los requisitos elaboradas.

Construcción de prototipos: Mediante esta técnica se construyen modelos (o prototipos) de la solución y se le muestran a los usuarios finales y a los clientes, los cuales interactúan con estos modelos con el fin de comprobar que se cumpla con sus necesidades reales (Sommerville, 2009). Para aplicar esta técnica, a medida que se fue desarrolla la solución, se muestran los prototipos resultantes al cliente.

Generación de casos de pruebas: Mediante esta técnica se diseñan pruebas orientadas a comprobar la calidad de los requisitos (Sommerville, 2009). Si una prueba es difícil o imposible de diseñar, esto puede significar que el requisito asociado a esa prueba es complejo y necesita ser rediseñado para lograr una menor complejidad. Estas pruebas se desarrollan una vez terminado el desarrollo de la solución. Su diseño se encuentra en el expediente del proyecto.

2.4 Actores del sistema

El actor de un sistema es un agente externo que interactúa con dicho sistema y que obtiene, como resultado, una respuesta del mismo. A continuación se identifican los actores que interactúan con el factor de autenticación basado en la autenticación por código de barras de la presente investigación:

Tabla 7 Descripción de los actores del sistema

ACTOR DEL SISTEMA	DESCRIPCIÓN
Usuario	Es el actor que puede autenticarse en el sistema. Una vez logrado este proceso de forma satisfactoria se especializa como *Especialista* o *Administrador*, según el rol que le corresponda.
Administrador	Es el usuario que puede interactuar con las funcionalidades relacionadas con la gestión de los usuarios en los diferentes sistemas suscritos a ACAXIA.
Especialista	Es el usuario que puede interactuar con las funcionalidades de uno o varios de los sistemas suscritos a ACAXIA, pero que no interactúa con las funcionalidades relacionadas con la gestión de usuarios.

2.5 Diseño arquitectónico de la solución

"El sistema de control de acceso ACAXIA presenta una arquitectura basada en N capas" (Treto Portal, 2013). El nuevo módulo de autenticación se desarrolla en las capas de negocio y acceso a datos. En la capa de acceso a datos se implementan las funcionalidades asociadas a la gestión de la autenticación. En la capa de negocio se implementan las funcionalidades asociadas a la gestión de usuarios y a la gestión de token. En ambas capas el sistema ACAXIA utiliza el patrón arquitectónico Modelo-Vista-Controlador (MVC), el cual divide la solución en 3 partes fundamentales, separando los datos de la aplicación, la interfaz de usuario y la lógica de negocio. A continuación se describe brevemente cada una de estas partes con el fin de lograr un mayor entendimiento:

Tabla 8 Descripción de las 3 partes del patrón Modelo-Vista-Controlador

PARTE	DESCRIPCIÓN
Modelo	Esta parte está compuesta por los datos, las reglas de negocio y las funcionalidades relacionadas con el acceso a los datos persistentes, lo cual se hace a través del marco de trabajo Doctrine. Es independiente de cualquier representación de salida y de entrada.
Controlador	En esta parte se gestionan las entradas que le da el usuario al sistema y las respuestas, asociadas a esas entradas, que el sistema le brinda al usuario. Las

	entradas usualmente se reciben como eventos asociados al movimiento y las pulsaciones del ratón, a la entrada de datos por teclado, entre otras. Las respuestas del sistema generalmente están asociadas a solicitudes de servicio que provienen de la Vista, aunque en ocasiones estas peticiones provienen del mismo Controlador.
Vista	En esta parte es donde el usuario puede interactuar con las interfaces gráficas asociadas al modelo. En el caso del proceso de autenticación, las interfaces son de tipos HTML, manejándose los eventos en las mismas a través del marco de trabajo Zend Framework. Estos eventos, o peticiones, son enviados al Controlador. Para el caso específico de los requisitos asociados a los grupos *Gestionar usuario* y *Gestionar datos*, las interfaces son de tipo ExtJs, manejándose los eventos de las mismas con la extensión Zend_Ext que posee el marco de trabajo Zend Framework. Estas peticiones son enviadas, al igual que en el proceso de autenticación, hacia el Controlador.

2.6 Patrones de diseño

"El diseño basado en patrones es una técnica que reutiliza elementos de diseño que han probado ser exitosos en el pasado" (Pressman, 2010). *"Los patrones de diseño permiten identificar clases, instancias, roles, colaboraciones y distribución de responsabilidades"* (Sommerville, 2009). *"Se clasifican, de forma general, en dos grandes grupos: los Patrones de Principios Generales para Asignar Responsabilidades (GRASP, por sus siglas en inglés) y los Patrones de Diseño Gound-of-Four (GoF, por sus siglas en inglés)"* (Larman, 1999). Ambos grupos de patrones van de la mano, siendo muy importantes en el proceso de desarrollo de software ya que ofrecen soluciones probadas y documentadas ante problemas comunes. A continuación se enuncian los diferentes patrones de diseño utilizados durante el proceso de desarrollo de la solución.

2.6.1 Patrones GRASP

Para el desarrollo de la solución se utilizaran varios de los patrones GRASP, considerándolos, más que patrones propiamente dichos, una conjunto de buenas prácticas de desarrollo. A continuación se caracterizan los patrones GRASP utilizados.

Experto: "*Este patrón se usa para establecer una responsabilidad a la clase que posee la información necesaria para cumplir con dicha responsabilidad"* (Larman, 1999). Dentro del sistema se utiliza este patrón en clases que deben ser las encargadas de proporcionar los datos cargados de una fuente de datos así como modificar los mismos. Por ejemplo, en la clase *SeguridadProxyService.*

Creador: "*Es usado para conceder a la clase B la responsabilidad de crear una instancia de clase A"* (Larman, 1999). Dentro del sistema se evidencia su uso en los casos de la instanciación de las clases dentro de la interfaz, los controladores y otros que contienen objetos como *SegUsuario*. Por ejemplo, en la clase *SeguridadProxyService.*

Controlador: "*Es usado para asignar el control de todos los eventos relacionados con el negocio a un grupo determinado de clases, usualmente llamadas controladoras"* (Larman, 1999). Un ejemplo del uso de este patrón sería en la clase *GestusuarioController*.

Alta Cohesión: "*Se emplea para asignar las responsabilidades de modo que se mantenga una alta cohesión"* (Larman, 1999). Se da una alta cohesión funcional cuando los elementos de un componente colaboran para producir algún comportamiento bien definido. Un ejemplo del uso de este patrón es en la clase *GestusuarioController*.

Bajo Acoplamiento: "*Se utiliza para asignar las responsabilidades de modo que se mantenga bajo acoplamiento"* (Larman, 1999). Este patrón se utiliza en la asignación de responsabilidades a clases de manera que un cambio en una de estas genere poco cambio en otras. Un ejemplo del uso de este patrón está dado en el uso de la clase *SeguridadProxyService*.

2.6.2 Patrones GoF

"Según el análisis realizado a diferentes autores, existen un total de 23 patrones de diseño de tipo GoF" (Larman, 1999). Estos patrones se agrupan en 3 grupos: patrones creacionales (creación de objetos), patrones estructurales (composición de clases y objetos) y patrones de comportamiento (caracterización de la forma en que interactúan y reparten responsabilidades las clases y objetos). A continuación se caracterizan los patrones GoF utilizados para el desarrollo de la solución.

Patrón creacional **Singleton**: "*Este patrón se aplica para garantizar el acceso único a una clase mediante una única instancia"* (Reynoso, et al., 2004). Para lograr esto, proporciona un punto de acceso global a todas las clases asociadas a la solución. Un ejemplo del uso de este patrón es a través de la clase *ZendExt_IoC* y las demás clases asociadas a la misma. En ellas se garantiza que cada clase tenga una instancia única, a la cual puede accederse desde cualquier parte del sistema.

Patrón estructural **Fachada**: "*Crea una única clase que permite acceder a un conjunto numeroso y complicado de clases"* (Reynoso, et al., 2004). En el desarrollo de la solución se utiliza en el uso de dicho patrón en la clase *SeguridadProxyService*, la cual se utiliza como interfaz de acceso a las demás interfaces y componentes del módulo y garantiza la comunicación entre ellos.

2.7 Modelo de diseño

"El modelo de diseño está basado en la descripción de la estructura y las funcionalidades de un sistema con un nivel de detalle elevado" (Acosta Alvarez, 2014). "*Mediante este modelo se representan los principales componentes de la solución, determinando su ubicación exacta en la arquitectura general"* (IBM, 2015). En el modelo de diseño, están contenidos elementos del sistema (clases, interfaces y subsistemas) dentro de paquetes. El modelo de diseño ayuda a entender el funcionamiento del software a partir de la visualización del software en general mediante sus principales elementos. Esto permite, a su vez, una mejor comprensión de los requisitos del sistema a construir.

2.7.1 Diagrama de paquetes

"Los diagramas de paquetes son usados, comúnmente, para esconder los elementos específicos del modelo de diseño, permitiendo la abstracción del sistema a solo paquetes o capas" (Larman, 1999). Haciendo uso de ellos se puede facilitar la comprensión del diseño, lo que implica, a su vez, que se facilite también la comprensión de los requisitos identificados. En el caso de la presente solución, se hace uso del diagrama de paquetes asociado al proceso de autenticación, ya existente. Este diagrama de paquetes se encuentra ubicado en el expediente de proyecto asociado al desarrollo del mismo, ubicado en el repositorio de desarrollo del Departamento de Tecnología del centro CEIGE, por lo que no se hace necesario realizar un nuevo diagrama de paquetes específico para la solución.

2.7.2 Diagramas de clases de diseño

"Los diagramas de clases especifican las diferentes clases que serán utilizadas en el sistema y las relaciones que existen entre ellas" (IBM, 2015). En el caso de la solución, se utilizó el modelo de clases de diseño del grupo de requisitos *Gestionar usuario*, ya existente. Este diagrama de clases de diseño se encuentra ubicado en el expediente de proyecto asociado al desarrollo de ACAXIA, situado en el repositorio de desarrollo del Departamento de Tecnología del centro CEIGE, por lo que no se hace necesario realizar un nuevo diagrama de clases específico para los requisitos pertenecientes a los grupos *Gestionar usuario* y *Gestionar token*. Se trabaja en el diagrama de clases de diseño asociado a los requisitos de grupo *Gestionar autenticación,* lo que permite una mayor claridad para entender el funcionamiento del proceso de autenticación.

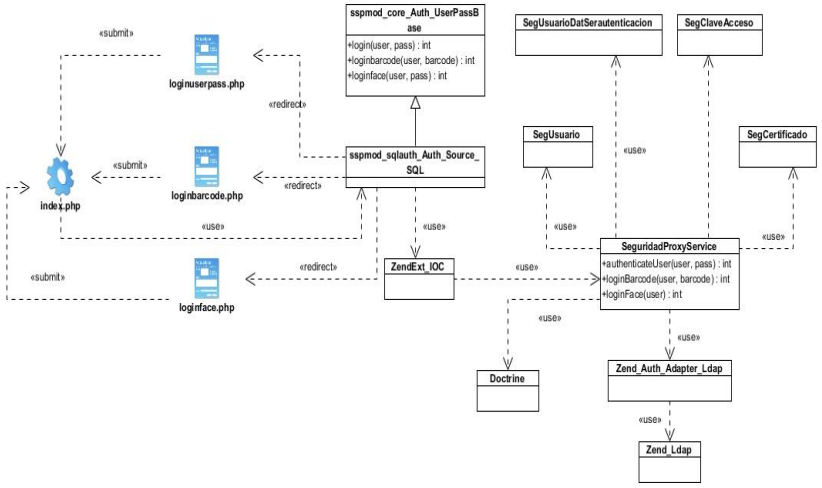


Figura 10 Diagrama de clases del diseño asociado al proceso de autenticación

En este diagrama queda reflejado cómo se envía una petición de autenticación desde cualquiera de las interfaces de usuario (**loginuserpass.php**, **loginbarcode.php**, **loginface.php**) hacia la clase controladora (**index.php**). Luego desde la clase controladora se envían los datos hacia la clase que brinda los servicios de conexión (**SeguridadProxyService)**, habiendo procesado los datos, primeramente, en las interfaces de conexión (**sspmod_sqlauth_Auth_Source_SQL**, **ZendExt_IOC**). Seguidamente, se realiza la petición de los datos a través de las clases de acceso a datos asociadas a los usuarios (**SegUsuario**, **SegUsuarioDatSerautenticacion**, **SegClaveAcceso**, **SegCertificado**). Desde las clases de acceso a datos se realiza la petición a la base de datos haciendo uso de la clase principal del marco de trabajo Doctrine (**Doctrine**) y de las clases asociadas a la conexión mediante el protocolo de acceso LDAP (**Zend_Auth_Adapter_Ldap**, **Zend_Ldap**). Una vez obtenidos los datos,

se comprueba en la clase **SeguridadProxyService** que el usuario que se desea autenticar es correcto y desde allí se envía los resultados hacia la clase controladora. A continuación se muestra una breve descripción de las clases.

Tabla 9 Descripción de las clases de diseño asociadas al proceso de autenticación

CLASES	DESCRIPCIÓN
index.php	Es la página servidora. Recibe la información enviada por los formularios de autenticación y crea un objeto **sspmod_sqlauth_Auth_Source_SQL** para que este se encargue de procesar los datos.
loginuserpass.php	Es el formulario asociado al factor de autenticación basado en usuario y contraseña.
loginbarcode.php	Es el formulario de autenticación asociado al factor de autenticación basado en código de barras.
loginface.php	Es el formulario de autenticación asociado al factor de autenticación basado en reconocimiento facial.
sspmod_sqlauth_Auth_Source_SQL	Clase interfaz del proveedor de identidad que hereda sus métodos de la clase **sspmod_core_Auth_UserPassBase**. Es donde se gestiona la autenticación del usuario, creando un objeto **ZendExt_IOC** con el fin de comprobar la identidad del usuario haciendo uso de servicios web. Luego redirecciona hacia el tipo de autenticación solicitado y se envía desde este los datos hasta la página servidora.
sspmod_core_Auth_UserPassBase	Clase abstracta que contiene la definición de los métodos y atributos que se van a usar en la clase **sspmod_sqlauth_Auth_Source_SQL**.
ZendExt_IOC	Integrador de servicios entre los módulos del sistema.
SeguridadProxyService	Proxy utilizado por ACAXIA para brindar los servicios de autenticación, autorización, auditoria y administración de perfiles. Es utilizada para

	autenticar el usuario con la base de datos de ACAXIA o con servidores LDAP.
SegUsuario	Clase entidad donde se ejecutan las consultas de Doctrine para obtener datos referentes a los usuarios.
SegUsuarioDatSerautenticacion	Clase entidad donde se ejecutan las consultas de Doctrine para obtener datos referentes a los usuarios que utilizan servidores de autenticación LDAP.
SegClaveAcceso	Clase entidad donde se ejecutan las consultas de Doctrine para obtener datos referentes a las claves de acceso.
SegCertificado	Clase entidad donde se ejecutan las consultas de Doctrine para obtener datos referentes a los certificados.
Doctrine	Clase principal del marco de trabajo Doctrine, que permite realizar consultas a la base de datos del sistema.
Zend_Auth_Adapter_Ldap	Clase interfaz de Zend_Ldap
Zend_Ldap	Permite la autenticación de usuarios con cualquier servidor LDAP.

2.8 Modelo de datos

"El modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí" (Larman, 1999). Por tal motivo, se le considera un aspecto fundamental para el desarrollo de cualquier aplicación que necesite almacenar datos. En el caso de la presente solución, se hace uso del modelo de datos ya existente, por lo que no se hace necesario realizar un nuevo modelo de datos específico para la solución. Se modifica la base de datos existente, agregándole un campo denominado *Código* a la tabla *SegUsuario*. El modelo de datos perteneciente al módulo *Seguridad* del sistema ACAXIA se encuentra recogido en el expediente de proyecto asociado al desarrollo del mismo, ubicado en el repositorio de desarrollo del Departamento de Tecnología del centro CEIGE.

2.9 Conclusiones parciales

Al término de este capítulo se completa la etapa de descripción de la solución, la cual arroja como resultado un conjunto de artefactos que permiten organizar y entender los principales conceptos que se manejan en el sistema y las relaciones entre ellos. Para ello se definen los requisitos funcionales y no funcionales, los cuales sirven de garantizar el correcto desarrollo de la solución. Por último se realiza el análisis y diseño de la nueva la nueva versión del módulo de autenticación para el sistema ACAXIA con el objetivo de facilitar la comprensión de las funcionalidades a desarrollar.

CAPÍTULO 3. CONSTRUCCIÓN Y VALIDACIÓN DEL MÓDULO DE AUTENTICACIÓN PARA EL SISTEMA ACAXIA VERSIÓN 1.1

3.1 Introducción

En el capítulo anterior se realiza la descripción de la solución propuesta, a partir de la cual, en el presente capítulo se confecciona el modelo de despliegue y el modelo de implementación. Se realizan pruebas al sistema con el fin de detectar y corregir las no conformidades y conseguir que el producto tenga mayor calidad y aceptación por parte del cliente. Finalmente se valida el objetivo propuesto a través de la realización de un experimento, el cual se complementa con una encuesta a especialistas que participan del proceso.

3.2 Modelo de despliegue

"*El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo*" (Hernández Couce, 2012). El mismo se puede representar como un conjunto de nodos, unidos por conexiones de comunicación que describe la relación existente entre los diferentes nodos que componen al sistema, así como la repartición de los componentes en dichos nodos. A continuación se muestra el diagrama correspondiente a la versión 1.1 del módulo de autenticación para el sistema ACAXIA.

Figura 11 Diagrama de despliegue de la solución

En el servidor Web se ejecutan diversas funcionalidades, entre las cuales se encuentra la construcción de interfaces de usuarios, el procesamiento de datos y el control de flujo. De igual modo, el servidor de base de datos se encarga de ejecutar el servidor PostgreSQL, donde se almacena la base de datos con la información referente a los usuarios del sistema ACAXIA y la base de datos de configuración de las aplicaciones suscritas al mismo. Estas bases de datos pueden estar distribuidas en otros nodos y conectadas entre ellas. Los clientes, por su parte, pueden acceder a la aplicación haciendo uso de los protocolos de comunicación TCP/IP y HTTP.

El protocolo TCP/IP se emplea para enlazar computadoras que utilizan diferentes sistemas operativos sobre redes de área local. En el caso del sistema de control de acceso ACAXIA se utiliza para interconectar los diferentes servidores, dígase servidor de base datos y servidor Web. En cambio el protocolo HTTP es usado en cada transacción de la Web, define la sintaxis y la semántica que utilizan los elementos software para comunicarse. En ACAXIA se utiliza este protocolo para el acceso de los clientes a los servicios que brinda la aplicación.

3.3 Modelo de implementación

El modelo de implementación es una representación de cómo se organizan las clases de un sistema a nivel de componentes. Simboliza, además, la dependencia que existe entre estos componentes. Todo esto permite que este modelo sea de gran utilidad para el desarrollo del sistema a través una visión más detallada de los componentes del mismo.

3.3.1 Diagrama de componentes

Los diagramas de componentes ilustran las partes del software que conforman un sistema (Larman, 1999). Un diagrama de componentes tiene un nivel más alto de abstracción que un diagrama de clase, usualmente un componente puede estar compuesto por una o más clases (Martin Díaz, et al., 2013). El diagrama de componentes muestra la vista física de la aplicación a través de componentes y sus relaciones; representa cómo un sistema de software es dividido en elementos y muestra las dependencias entre estos elementos. A continuación se muestra el diagrama de componentes perteneciente al proceso de autenticación, el cual se puede encontrar en el expediente de proyecto de la solución.

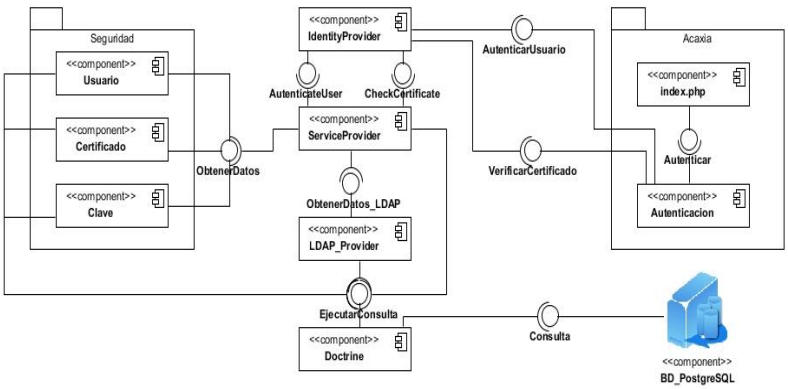


Figura 12 Diagrama de componentes del módulo de autenticación

En este diagrama de componentes se puede observar la interacción entre los principales componentes que actúan en el proceso de autenticación. El proceso comienza en la clase **index.php**, desde donde se envía la petición al componente **Autenticación** para que se realice todo el proceso. Luego se envían peticiones de autenticación de usuarios y verificación de los certificados hacia los componentes principales del subsistema **Seguridad** (**Usuario**, **Certificado**, **Clave**). Este envío se realiza a través del componente **IdentityProvider,** que a su vez utiliza el componente **ServiceProvider** como puente de enlace entre el subsistema **Seguridad** y él mismo.

Para obtener los datos, **ServiceProvider** realiza dos procedimientos. El primero se realiza enviando la petición de autenticación al subsistema de **Seguridad**, comprobando los datos de los usuarios locales. El segundo se realiza a través del protocolo de conexión LDAP, haciendo uso del componente **LDAP_Provider**, comprobando los datos del resto de los usuarios. Estos dos procedimientos obtienen la información a través de consultas realizadas mediante el componente **Doctrine**, el cual enlaza al subsistema **Seguridad** y al componente **LDAP_Provider** con el servidor de base de datos de PostgreSQL (**BD_PotgreSQL**).

3.4 Pruebas a la solución

Siempre que se desarrolla un sistema informático, es de gran importancia garantizar que el mismo posea la mayor calidad posible, por lo que se aconseja probar cada producto desarrollado. Durante la etapa de validación se le aplican pruebas al sistema, las cuales se encuentran agrupadas en dos grupos: las pruebas de caja blanca y las pruebas de caja negra. En el caso específico de la solución, se realizan pruebas de caja blanca y caja negra para garantizar la calidad del mismo.

3.4.1 Pruebas de caja blanca

"Las pruebas de caja blanca están dirigidas a las funciones internas del módulo, caso contrario a las de caja negra, que examinan los requisitos funcionales desde el exterior del módulo" (Sommerville, 2009). Las pruebas de caja blanca pueden aplicarse a los métodos de la clase, aunque generalmente son usadas en funciones complejas de programación estructurada. Teniendo en cuenta lo expuesto anteriormente, se decide aplicar este tipo de pruebas a las funciones de mayor complejidad, aprovechando así las potencialidades de las pruebas de caja blanca para detectar errores de codificación y flujo.

Dentro de las pruebas de caja blanca, se emplea la técnica del camino básico, la cual permite conocer una medida de la complejidad lógica de una función procedural y usarla como guía para definir un conjunto básico de rutas de ejecución. Mediante esta técnica se garantiza que cada instrucción se ejecute al menos una vez durante el desarrollo de la prueba.

A continuación se muestran las pruebas de caja blanca realizadas sobre una de las funciones procedurales de la solución: **getDatosUsuario()**. Se analiza la complejidad ciclomática, que es la métrica de software que proporciona una medición cuantitativa de la complejidad lógica de un programa. *"La complejidad ciclomática calcula la cantidad de caminos independientes de cada una de las funcionalidades del programa y provee el límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez"* (Pressman, 2010).

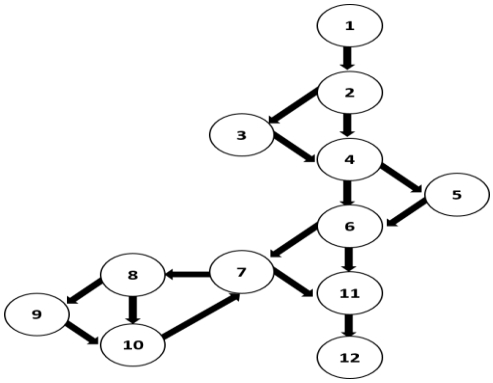


Figura 13 Grafo de flujo asociado a la funcionalidad getDatosUsuario()

Luego de elaborar el grafo de flujo asociado a la funcionalidad **getDatosUsuario()**, se calcula la complejidad ciclomática atendiendo a tres diferentes fórmulas:

1. **V(G) = R**, donde R representa la cantidad de regiones en el grafo.
   V(G) = 6
2. **V(G) = A – N + 2**, donde A es el número de aristas y N el número de nodos.
   V(G) = 16 – 12 + 2 = 6
3. **V(G) = P + 1**, donde P es el número de nodos predicado contenidos en el grafo.
   V(G) = 5 + 1 = 6

Teniendo en cuenta que el cálculo de la complejidad ciclomática arrojó el mismo resultado para las 3 variantes y que este valor representa el número mínimo de pruebas para la función, se puede afirmar

que la complejidad ciclomática de la función es 6, lo que significa que existen 6 posibles caminos, mostrados a continuación, por donde el flujo puede circular.

- Camino #1: 1-2-4-6-11-12
- Camino #2: 1-2-3-4-6-11-12
- Camino #3: 1-2-4-5-6-11-12
- Camino #4: 1-2-4-6-7-11-12
- Camino #5: 1-2-4-6-7-8-10-7-11-12
- Camino #6: 1-2-3-4-6-7-8-9-10-7-11-12

3.4.2 Pruebas de caja negra

"Las pruebas de caja negra examinan aspectos del sistema sin tener en cuenta la estructura interna del software" (Pressman, 2010). "*Se llevan a cabo sobre la interfaz gráfica de usuario, lo que permite demostrar la operatividad de las funciones internas del producto, permitiendo a su vez que se compruebe la entrada de los datos por parte del usuario, así como la salida que arroja el sistema"* (Sommerville, 2009). Todo esto permite encontrar funciones incorrectas o incompletas, errores en la interfaz gráfica de usuario, así como errores en el rendimiento, inicialización y terminación de los diferentes procesos que se manejan en el sistema.

Dentro de las pruebas de caja negra, se emplea como técnica la partición equivalente, la cual consiste en dividir el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software. Esta técnica permite descubrir de forma inmediata errores que de otro modo requerirían la ejecución de muchos casos antes de ser detectados. A continuación se muestran algunas de las pruebas realizadas al sistema, el resto de las pruebas se encuentran en el expediente de proyecto asociado a la solución.

3.4.3 Diseño de casos de prueba del requisito funcional Autenticar por código de barras

Descripción general: Este requisito se lleva a cabo con el objetivo de que los usuarios del sistema puedan autenticarse en el mismo haciendo uso de un código de barras asociado a su identificador. Se inicia una vez que el usuario haya insertado su usuario y contraseña satisfactoriamente y haya proseguido con el proceso de autenticación y termina cuando el sistema verifica que los datos entrados por el usuario son correctos, permitiéndole entrar al sistema.

Condiciones de ejecución:

1. El usuario debe estar registrado en el sistema.
2. La autenticación por usuario y contraseña debe estar activada.
3. El usuario debe haber insertado su usuario y contraseña satisfactoriamente.

Tabla 10 Escenarios del caso de prueba correspondiente al requisito Autenticar por código de barras

SC AUTENTICAR USUARIO POR CÓDIGO DE BARRAS				
Escenario	**Descripción**	**Variable 2**	**Respuesta del sistema**	**Flujo central**
EC 1.1.1 Autenticar usuario por código de barras satisfactoriamente	El usuario se autentica por código de barras de forma satisfactoria.	V	El sistema habilita las funcionalidades definidas para el usuario autenticado.	1. El usuario introduce el código de barras a través del escáner.
		1111		2. El usuario accede al sistema.
EC 1.1.2 Introducir un código de barras erróneo	El usuario inserta un código de barras que no coincide con el código que le pertenece en la base de datos.	I	El sistema muestra un error al usuario y recarga la página.	1. El usuario introduce el código de barras a través del escáner.
		2222		2. El sistema muestra el error.
EC 1.1.3 No introducir código de barras	El usuario intenta autenticarse sin insertar código de barras.	N/A	El sistema muestra un error al usuario y recarga la página.	1. El usuario no introduce el código de barras
				2. El sistema muestra el error.

Tabla 11 Descripción de las variables del caso de prueba correspondiente al requisito Autenticar por código de barras

NO	NOMBRE DE CAMPO	CLASIFICACIÓN	VALOR NULO	DESCRIPCIÓN
1	Código	Campo de texto	No	Dígitos del 0 al 9. Letras mayúsculas y minúsculas desde la a hasta la z.

3.4.4 Resultados de las pruebas

Para un mayor entendimiento de los resultados obtenidos durante el proceso de validación, se desglosaron las no conformidades (NC) detectadas en tres grupos:

- Significativas (S), las cuales incluyen los errores comunes en funciones y excepciones, así como errores de validación.
- No significativas (NS), que incluyen errores ortográficos.
- No proceden (NP), que son aquellas que están fuera del alcance del equipo de desarrollo.

A continuación se muestra una tabla donde se recogen los resultados de las pruebas, agrupando los requisitos de acuerdo a los grupos creados en la fase de captura de los requisitos.

Tabla 12 Resultados del proceso de validación de la solución

Grupos de requisitos	Iteración 1				Iteración 2			
	S	NS	NP	NC (Total)	S	NS	NP	NC (Total)
Gestionar autenticación	3	1	0	4	0	0	0	0
Gestionar usuario	2	2	0	4	0	0	0	0
Gestionar token	1	1	0	2	0	0	0	0

Se detectan un total de 10 no conformidades, todas en la primera iteración, de las cuales 4 son de funcionalidad, 2 de validación y 4 de ortografía. En la segunda iteración no se detectan no conformidades. Estos datos se reflejan en la siguiente gráfica:

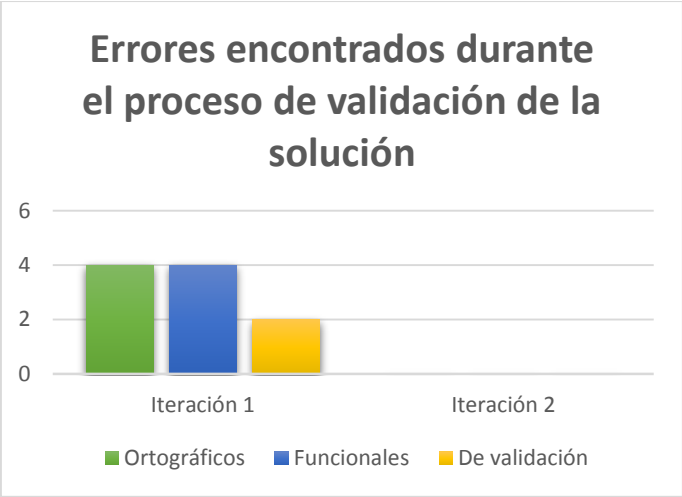


Figura 14 Errores encontrados en el proceso de validación de la solución

Luego de terminar la realización de pruebas a la solución y una vez corregidas las no conformidades detectadas, se puede afirmar que la nueva versión del módulo de autenticación para el sistema ACAXIA cumple con todas las funcionalidades definidas. El acta de liberación resultante del proceso se encuentra recogida en el Anexo 1 de la presente investigación.

3.5 Validación de la solución

Con el objetivo de validar la solución, se elabora un experimento. Durante la realización de este experimento se le aplican pruebas de penetración de carga especializada al sistema ACAXIA utilizando herramientas de hacking ético. Estas pruebas tienen como objetivo comprobar el cumplimiento de las variables: nivel de confidencialidad de la información, nivel de integridad de la información y nivel de disponibilidad de la información. Para realizar este proceso de validación se realizan un conjunto de pasos los cuales se detallan a continuación.

3.5.1 Medición del nivel de confiabilidad de la información

Para la medición del nivel de confiabilidad de la información se utiliza la herramienta **Burp Suite Intruders** con el objetivo de obtener información acerca de las vulnerabilidades a los ataques asociados

a la suplantación de identidad (**Spoofing**). Para hacer uso de esta herramienta se utiliza la interfaz gráfica que posee la misma, rellenando los capos a través de expresiones regulares previamente definidas.

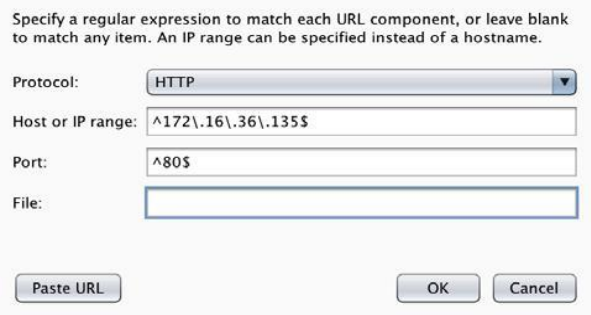


Figura 15 Interfaz gráfica de la herramienta Burp Suite Intruders para la selección del rango de IP y puertos de conexión

Luego se selecciona el tipo de ataque que se desea realizar, escogiendo el ataque de una lista predefinida en la interfaz gráfica de la herramienta. Esta lista de ataques contiene el nombre de los factores a atacar.

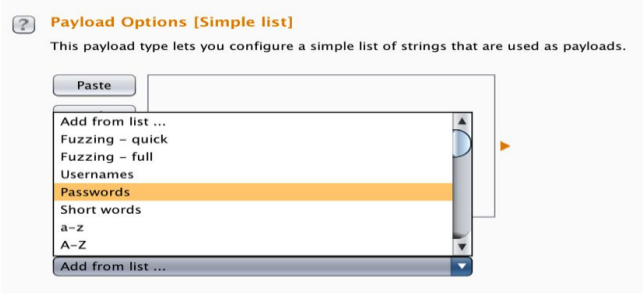
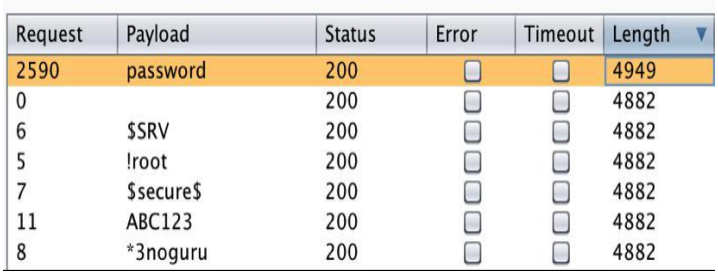


Figura 16 Interfaz gráfica de la herramienta Burp Suite Intruders para la selección del tipo de ataque que se desea realizar

Por último, la herramienta escanea el rango de IP y los puertos de conexión seleccionados y comprueba los errores de seguridad de los sistemas web que ejecutan en los mismos. Los resultados de los ataques de penetración realizados por la herramienta se muestran en otra interfaz gráfica, la cual contiene información acerca de si se pudo realizar el ataque de forma satisfactoria y el tiempo que demoró en realizarse.



Request	Payload	Status	Error	Timeout	Length
2590	password	200	☐	☐	4949
0		200	☐	☐	4882
6	$SRV	200	☐	☐	4882
5	!root	200	☐	☐	4882
7	$secure$	200	☐	☐	4882
11	ABC123	200	☐	☐	4882
8	*3noguru	200	☐	☐	4882

Figura 17 Interfaz gráfica de la herramienta Burp Suite Intruders para la obtención de los resultados

Luego de haber realizado las pruebas correspondientes a esta etapa del experimento **se obtiene como resultado** que 2 de los ataques realizados a la primera versión suceden de forma satisfactoria. Estos ataques arrojan tiempos de ejecución bajos, lo que indica que la brecha detectada es fácil de vulnerar. Todo esto conduce que dicha versión no garantice la confiabilidad de la información. Sin embargo, los ataques realizados a la nueva versión del módulo son insatisfactorios. A continuación se muestran estos resultados.

Tabla 13 Resultados de la medición del nivel de confiabilidad de la información.

RESULTADOS DE LA MEDICIÓN DEL NIVEL DE CONFIABILIDAD DE LA INFORMACIÓN			
VERSIÓN DEL MÓDULO	**CANTIDAD DE ATAQUES REALIZADOS**	**CANTIDAD DE ATAQUES SATISFACTORIOS**	**TIEMPO PROMEDIO DE EJECUCIÓN DE LOS ATAQUES SATISFACTORIOS (SEGUNDOS)**
1.0	7	2	8966
1.1	7	0	-

3.5.2 Medición del nivel de integridad de la información

Para la medición del nivel de integridad de la información se utiliza la herramienta **SQLMap** con el objetivo de obtener la información que se guarda en la base de datos. Es por eso que se le realizan ataques asociados a la modificación de la información a través de las dos versiones del módulo, intentando comprometer la integridad de la información que se maneja a través de los mismos. Para hacer uso de esta herramienta se utiliza la consola del sistema y desde esta se introduce una serie de comandos que permiten realizar el ataque. Es necesario conocer la dirección IP y el puerto por el que se ejecuta el sistema a comprobar, por lo que se utilizan los resultados obtenidos en la prueba anterior.

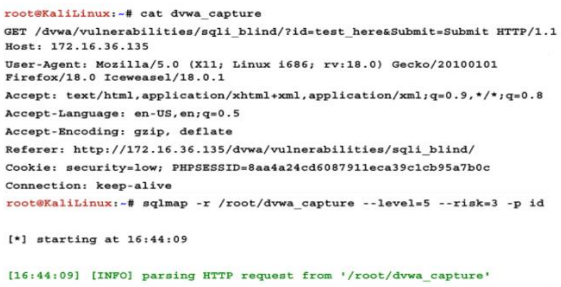


Figura 18 Consola del sistema Kali con un ejemplo de la utilización de la herramienta SQLMap

Por último, la herramienta devuelve los resultados de la prueba realizada, obteniéndose el nombre de los campos vulnerables para ser modificados. Esto permite comprobar la integridad de la información ya que esta depende de si la información puede ser modificada o no por un usuario sin acceso a ella.

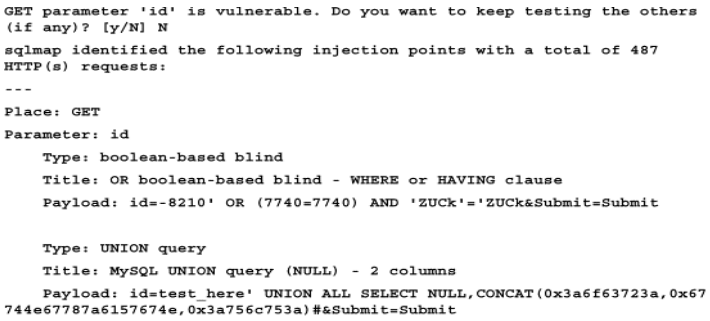


Figura 19 Resultados obtenidos con la herramienta SQLMap

Luego de haber realizado las pruebas correspondientes a esta etapa del **experimento se obtiene como resultado** que 2 de los ataques realizados a la primera versión suceden de forma satisfactoria. Esto indica que dicha versión no garantiza la integridad de la información. Sin embargo, todos los ataques realizados a la nueva versión del módulo son insatisfactorios. A continuación se muestran estos resultados.

Tabla 14 Resultados de la medición del nivel de integridad de la información

RESULTADOS DE LA MEDICIÓN DEL NIVEL DE INTEGRIDAD DE LA INFORMACIÓN		
VERSIÓN DEL MÓDULO	**CANTIDAD DE ATAQUES REALIZADOS**	**CANTIDAD DE ATAQUES SATISFACTORIOS**
1.0	7	2

1.1	7	0

3.5.3 Medición del nivel de disponibilidad de la información

Para la medición del nivel de disponibilidad de la información se utiliza la herramienta **HTTP Traffic** con el objetivo de comprometer la disponibilidad de la información en el sistema ACAXIA. Es por esto que se le realizan ataques asociados a la denegación de servicio (**DoS**) a través de las dos versiones del módulo. Para la utilización de la herramienta se hace uso de la interfaz gráfica de la misma. Es necesario tener conocimiento previo de la dirección IP y el puerto de ejecución por lo que se utilizan los resultados de la primera prueba.

Figura 20 Interfaz gráfica de la herramienta HTTP Traffic

Se necesita de un código fuente elaborado en el lenguaje Python, el cual tiene como objetivo interrumpir la ejecución del sistema parcial o totalmente. Es por esto que se utiliza el código de ejemplo que contiene la herramienta, el cual le asigna una dirección IP no válida al sistema, lo que provoca que no se pueda tener acceso al mismo.

```
#!/usr/bin/python

import socket

httprecv = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
httprecv.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
httprecv.bind(("0.0.0.0",8000))
httprecv.listen(2)

(client, ( ip,sock)) = httprecv.accept()
print "Received connection from : ", ip
data = client.recv(4096)
print str(data)

client.close()
httprecv.close()
```

Figura 21 Código fuente utilizado para la denegación de los servicios

Una vez ejecutado el ataque, se comprueba manualmente si la herramienta logra ejecutar la prueba de forma satisfactoria o si no pudo hacerlo. Para ello, se intenta acceder a la información a través de un usuario válido para el sistema ACAXIA.

Luego de haber realizado las pruebas correspondientes a esta etapa del **experimento se obtiene como resultado** que 2 de los ataques realizados a la primera versión del módulo suceden de forma satisfactoria. Esto indica que dicha versión no garantiza la disponibilidad de la información. Sin embargo, ninguno de los ataques realizados a la nueva versión del módulo es insatisfactorio. A continuación se muestran estos resultados.

RESULTADOS DE LA MEDICIÓN DEL NIVEL DE DISPONIBILIDAD DE LA INFORMACIÓN		
VERSIÓN DEL MÓDULO	**CANTIDAD DE ATAQUES REALIZADOS**	**CANTIDAD DE ATAQUES SATISFACTORIOS**
1.0	7	2
1.1	7	0

3.5.4 Supervisión del proceso de desarrollo del experimento

Se seleccionan 3 especialistas del departamento de desarrollo de componentes del centro CEIGE, los cuales permanecen durante todo el proceso con el objetivo de supervisar la correcta realización del mismo. Para la selección de estos especialistas se utilizan 3 criterios fundamentales: el nivel científico, el compromiso ético con la solución y la experiencia previa en el trabajo con el módulo de autenticación. Luego de definir estos criterios, se elabora el listado de los especialistas. Por último se le aplica una encuesta, con el objetivo de recolectar datos que se hayan quedado fuera del alcance del experimento. Esta encuesta se encuentra disponible en el expediente del proyecto, además de en el Anexo 2 de la presente investigación.

3.5.5 Resultados de la encuesta

Luego de haber realizado la encuesta a los especialistas del departamento de desarrollo de componentes del centro CEIGE, se obtiene como resultado de las preguntas directas elaboradas en el mismos que estos especialistas consideran que la versión 1.1 módulo de autenticación garantiza la confiabilidad, la integridad y la disponibilidad de la información. A su vez, están de acuerdo en que al cumplirse esto, se logra aumentar el nivel de seguridad del proceso de autenticación del sistema ACAXIA, teniendo en cuenta la comparación de los resultados del experimento y los estándares

definidos por los expertos en la materia a nivel mundial. A continuación se muestra una tabla donde se refleja la respuesta de cada uno de estos especialistas durante la encuesta.

Tabla 15 Análisis de la encuesta realizada a los especialistas

RESPUESTAS DADAS POR LOS ESPECIALISTAS A LAS PREGUNTAS DIRECTAS DURANTE LA REALIZACIÓN DE LA ENCUESTA				
ESPECIALISTA	**PREGUNTA 1**	**PREGUNTA 2**	**PREGUNTA 3**	**PREGUNTA 4**
A	9	9	8	SÍ
B	10	9	9	SÍ
C	9	10	9	SÍ
PORCIENTO DE ACEPTACIÓN	93.3 %	93.3 %	86.7 %	100 %

Para un mejor entendimiento de los resultados de la tabla, a continuación se detallan los elementos evaluados en la misma:

- La **pregunta 1** está asociada al nivel de confiabilidad de la información.
- La **pregunta 2** está asociada al nivel de integridad de la información.
- La **pregunta 3** está asociada al nivel de disponibilidad de la información.
- La **pregunta 4** está asociada al aumento del nivel de seguridad en el proceso de autenticación.

Por otra parte, también se detallan los valores de la columna especialista, con el objetivo de un mayor entendimiento:

- **Especialista A** se refiere al jefe del departamento de desarrollo de componentes del centro CEIGE.
- **Especialista B** se refiere al analista principal del departamento de desarrollo de componentes del centro CEIGE.
- **Especialista C** se refiere a uno de los desarrolladores del departamento de desarrollo de componentes del centro CEIGE, el cual presenta experiencia en el desarrollo del módulo de autenticación.

3.6 Conclusiones parciales

Al término del capítulo se concluye la etapa de construcción y validación de la presente investigación, arrojando como resultado una nueva versión del módulo de autenticación para el sistema ACAXIA. Para obtener este resultado se implementa un factor de autenticación basado en código de barras, el cual se integra con los existentes en el módulo de autenticación de ACAXIA. Además, se realizan pruebas de caja blanca y pruebas de caja negra, lo que permite afirmar que la nueva versión del módulo de autenticación para el sistema ACAXIA cumple con todas las funcionalidades definidas. Por último, se valida el cumplimiento del objetivo a través de un experimento, realizando además una encuesta a un grupo de especialistas que participa en el experimento.

CONCLUSIONES GENERALES

Al término de la investigación se concluye que:

- Hasta el momento en que se comienza la investigación, el sistema ACAXIA solo contaba con 2 niveles de autenticación, los cuales no eran suficientes para garantizar la seguridad de las aplicaciones suscritas al mismo.
- La mayoría de las herramientas y tecnologías, usadas para el desarrollo de la solución, son libres, lo que posibilitó el desarrollo de la informática en el país sin incurrir en grandes gastos de recursos.
- Se identificaron 6 requisitos funcionales, necesarios para el desarrollo de la solución, los cuales permitieron desarrollar diferentes artefactos ingenieriles correspondientes a la metodología seleccionada, que brindan una mayor comprensión y organización de los principales conceptos asociados a la solución.
- Se realizaron pruebas a la solución, lo que permitió comprobar la calidad y el cumplimiento de todos los requisitos definidos.
- Se realizó un experimento, el cual se complementó con una entrevista a un grupo de especialistas, garantizando así el cumplimiento del objetivo propuesto.
- La nueva versión del módulo de autenticación aumenta el nivel de seguridad del proceso de autenticación en el sistema ACAXIA, garantizando así la confiabilidad, la integridad y la disponibilidad de la información.

RECOMENDACIONES

Una vez concluida la investigación y teniendo en cuenta las experiencias obtenidas a lo largo del desarrollo de la misma, se recomienda:

- Incluir otros factores de autenticación al módulo de autenticación del sistema de control de acceso ACAXIA, con el fin de seguir aumentando el nivel de seguridad del proceso de autenticación en este sistema.

REFERENCIAS BIBLIOGRÁFICAS

Acosta Alvarez, Martha. 2014. *Libélula: Sistema de Información Geográfica para la Feria Internacional del Libro de La Habana.* La Habana : Universidad de las Ciencias Informáticas, 2014.

Apache Software Foundation. 2014. Apache.org. [En línea] Apache Software Foundation, 2014. [Citado el: 9 de febrero de 2015.] http://httpd.apache.org/.

Axesor. 2015. Axesor - Información de empresas. *Teknotag Identification System Sociedad Limitada.* [En línea] 2015. [Citado el: 1 de abril de 2015.] http://www.axesor.es/Informes-Empresas/5229074/TEKNOTAG_IDENTIFICATION_SYSTEMS_SOCIEDAD_LIMITADA.html.

Carro Paz, Roberto y González Gómez, Daniel. 2010. *Identificación Automática.* s.l. : Universidad Nacional de Mar del Plata, 2010.

Castelló Martínez, Vicente. 2005. *Localización y decodificación de códigos de barras en imágenes digitales.* Castellón de la Plana : Universitat Jaume I, 2005.

Fernández López, Gracia. 2007. Seguridad en Sistemas de Información. [aut. libro] Gracia Fernández Lopez. *Seguridad en Sistemas Operativos.* España : s.n., 2007.

Fumero González, Gelsys y Estopiñán Lantigua, Alejandro. 2013. *Desarrollo de una herramienta para la evaluación de la seguridad mediante métricas a los productos que utilizan Acaxia del CEIGE.* La Habana : Universidad de las Ciencias Informáticas, 2013.

Gómez Baryolo, Oiner, Rivero Pino, Noel Jesús y López Méndez, Daniel E. 2011. *Sistema de gestión integral de seguridad Acaxia.* La Habana : Serie Científica de la Universidad de las Ciencias Informáticas, 2011.

Gutiérrez, Pedro. 2014. GENBETA:Dev. *Authy: añade autenticación en dos pasos fácilmente a tus aplicaciones.* [En línea] Weblogs SL, 2 de Mayo de 2014. [Citado el: 8 de Febrero de 2015.] http://www.genbetadev.com/desarrollo-web/authy-anade-autenticacion-en-dos-pasos-facilmente-a-tus-aplicaciones.

Hernández Couce, Daryl. 2012. *Sistema de Información Geográfica para la Universidad de las Ciencias Informáticas sobre dispositivos móviles basado en los servicios Web estandarizados por la Open Geospatial Consortium.* La Habana : Universidad de las Ciencias Informáticas, 2012.

Hernández Díaz, Rita Milena y Conde Bernal, Yaciel. 2013. *Aplicación Web para la administración de tarjetas inteligentes con GlobalPlatform.* La Habana : Universidad de las Ciencias Informáticas, 2013.

Hernández Sampieri, Roberto, Fernández Collado, Carlos y Baptista Lucio, María del Pilar. 2009. *Metodología de la investigación.* México : McGrawHill Educación, 2009. 978-607-15-0291-9.

IBM. 2015. IBM. [En línea] 2015. [Citado el: 10 de Abril de 2015.] http://www-01.ibm.com.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software.* España : Pearson Educación.

Larman, Craig. 1999. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* México : PRENTICE HALL, 1999. 970-17-0261-1.

León Hernández, Rolando Alfredo y González Coello, Sayda. 2011. *El proceso de investigación científica.* La Habana : Editorial Universitaria, 2011. 978-959-16-1307-3.

Martin Díaz, Racielis y Quiroga Arencibia, José Ernesto. 2013. *Solución de autorización basada en el estándar XACML para el sistema de seguridad ACAXIA.* La Habana : Universidad de las Ciencias Informáticas, 2013.

Martínez Romero, Anisley y Companioni Sosa, Reynaldo Jesús. 2012. *Propuesta de Subsistema de control de acceso para el Sistema de Planificación y Control del Servicio de Aliment ación de la Universidad de las Ciencias Informáticas.* La Habana : Universidad de Las Ciencias Informáticas, 2012.

Microsoft Corporation. 2009. Microsoft TechNet Seguridad. [En línea] 2009. http://www.microsoft.com/latam/technet/seguridad/articulos/bpsegcorp.mspx.

Mozilla Developer Network. 2015. Mozilla Developer Network. *MDN.* [En línea] Mozilla Foundation, 2015. [Citado el: 2 de abril de 2015.] https://developer.mozilla.org/es/docs/JavaScript.

M'Raihi, D., y otros. 2011. *TOTP: Time-Based One-Time Password Algorithm.* s.l. : Internet Engineering Task Force (IETF), 2011. 2070-1721.

Oryarzabal Arocena, Marcelo. 2015. Urrats bat. *Zentroak eraberritzea.* 2015, 44.

Pelaez González, Reinaldo. 2011. *Drivers de autenticación para el sistema de seguridad ACAXIA.* La Habana : Universidad de las Ciencias Informáticas, 2011.

Pérez San-José, Pablo. 2012. Guía sobre riesgos y buenas prácticas en autenticación online. [En línea] 2012. http://www.inteco.es/guias/Guia_Autenticacion.

Pfleeger, Charles P. 2006. *Security in computing.* 2006. ISBN: 978-0-13-239077-4.

PHP. 2009. Introduction: What is PHP? . *Php.net.* [En línea] PHP, 2009. [Citado el: 9 de febrero de 2015.] http://php.net/manual/es/intro-whatis.php.

PosgreSQL. 2009. Acerca de PostgreSQL. *ProgreSQL.es.* [En línea] PostgreSQL, 2009. [Citado el: 9 de febrero de 2015.] http://www.postgresql.org.es/sobre_postgresql.

Pressman, Roger S. 2010. *Ingeniería de Software, un enfoque práctico.* Madrid : Mc Graw Hill, 2010.

Rankl, Wolfgang y Effing, Wolfgang. 2010. *Smart Card Handbook: Fourth Edition.* Hoboken, New Jersey : Wiley John + Sons Inc., 2010. 978-0470743676.

Reynoso, Carlos y Kiccillof, Nicolás. 2004. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft Versión.* Buenos Aires : Universidad de Buenos Aires, 2004.

Rodríguez Sánchez, Tamara. 2014. *Metodología de desarrollo para la Actividad productiva de la UCI.* La Habana : Universidad de las Ciencias Informáticas, 2014.

Silveiro Leyva, Osmayda y Sabuquet Hurtado, Ernesto. 2013. *Sistema de Identificación mediante Huella Dactilar.* La Habana : Universidad de las Ciencias Informáticas, 2013.

Sommerville, Ian. 2009. *Ingeniería de Software.* Madrid : Pearson Education, 2009. 84-7829-074-5.

Suhendra, V. 2011. *A Survey on Access Control Deployment.* Berlin : Springer Berlin Heidelberg 259, 2011. págs. 11-20.

TeknoTAG. 2015. TeknoTAG - Control de acceso a eventos y congresos. *Control de accesos - TeknoTAG.* [En línea] 2015. [Citado el: 1 de abril de 2015.] http://teknotag.com/index.php/control-de-accesos-eventos.

Treto Portal, Janier. 2013. *Modelo de identificación y autenticación basado en reconocimiento facial para el Sistema de Gestión Integral de Seguridad Acaxia.* La Habana : Universidad de las Ciencias Informáticas, 2013.

Villalón Huerta, Antonio. 2002. *Seguridad en UNIX y redes.* s.l. : Free Software Foundation, 2002.

BIBLIOGRAFÍA CONSULTADA

Acosta Alvarez, Martha. 2014. *Libélula: Sistema de Información Geográfica para la Feria Internacional del Libro de La Habana.* La Habana : Universidad de las Ciencias Informáticas, 2014.

Apache Software Foundation. 2014. Apache.org. [En línea] Apache Software Foundation, 2014. [Citado el: 9 de febrero de 2015.] http://httpd.apache.org/.

Axesor. 2015. Axesor - Información de empresas. *Teknotag Identification System Sociedad Limitada.* [En línea] 2015. [Citado el: 1 de abril de 2015.] http://www.axesor.es/Informes-Empresas/5229074/TEKNOTAG_IDENTIFICATION_SYSTEMS_SOCIEDAD_LIMITADA.html.

Babylon. 2015. Definición de Java. *Babylon 10.* [En línea] Babylon, 2015. [Citado el: 9 de febrero de 2014.] http://diccionario.babylon.com/java?&tl=/.

Carpenter, A. E. 2011. www.ehowenespanol.com. [En línea] 2011. http://www.ehowenespanol.com/son-sistemas-biometricos-huellas-dactilares-info_224048/.

Carro Paz, Roberto y González Gómez, Daniel. 2010. *Identificación Automática.* s.l. : Universidad Nacional de Mar del Plata, 2010.

Castelló Martínez, Vicente. 2005. *Localización y decodificación de códigos de barras en imágenes digitales.* Castellón de la Plana : Universitat Jaume I, 2005.

Exponda, S. 2007. *Sistema de autorización para Web Services basadois en XACML.* Buenos Aires : Universidad de Belgrano, 2007.

Farlex. 2015. The Free Dictionary. *The Free Dictionary.com.* [En línea] Farlex Inc., 2015. [Citado el: 9 de febrero de 2015.] http://es.thefreedictionary.com/Java.

Febles Parker, Michel Evaristo. 2012. *Desarrollo de algoritmo de comparación para biblioteca.* La Habana : Universidad de las Ciencias Informáticas, 2012.

Fernández López, Gracia. 2007. Seguridad en Sistemas de Información. [aut. libro] Gracia Fernández Lopez. *Seguridad en Sistemas Operativos.* España : s.n., 2007.

Fumero González, Gelsys y Estopiñán Lantigua, Alejandro. 2013. *Desarrollo de una herramienta para la evaluación de la seguridad mediante métricas a los productos que utilizan Acaxia del CEIGE.* La Habana : Universidad de las Ciencias Informáticas, 2013.

Furfaro, Alejandro. 2010. *Manejo de Bibliotecas OpenCV.* 2010.

Gómez Baryolo, Oiner, Rivero Pino, Noel Jesús y López Méndez, Daniel E. 2011. *Sistema de gestión integral de seguridad Acaxia.* La Habana : Serie Científica de la Universidad de las Ciencias Informáticas, 2011.

Gracia, Luis Miguel. 2011. JJIL: Librería procesamiento imágenes. *Un poco de Java.* [En línea] 2011. [Citado el: 9 de febrero de 2015.] https://unpocodejava.wordpress.com/2011/01/12/jjil-libreria-procesamiento-imagenes/.

Gutiérrez, Pedro. 2014. GENBETA:Dev. *Authy: añade autenticación en dos pasos fácilmente a tus aplicaciones.* [En línea] Weblogs SL, 2 de Mayo de 2014. [Citado el: 8 de Febrero de 2015.] http://www.genbetadev.com/desarrollo-web/authy-anade-autenticacion-en-dos-pasos-facilmente-a-tus-aplicaciones.

Hernández Couce, Daryl. 2012. *Sistema de Información Geográfica para la Universidad de las Ciencias Informáticas sobre dispositivos móviles basado en los servicios Web estandarizados por la Open Geospatial Consortium.* La Habana : Universidad de las Ciencias Informáticas, 2012.

Hernández Díaz, Rita Milena y Conde Bernal, Yaciel. 2013. *Aplicación Web para la administración de tarjetas inteligentes con GlobalPlatform.* La Habana : Universidad de las Ciencias Informáticas, 2013.

Hernández Sampieri, Roberto, Fernández Collado, Carlos y Baptista Lucio, María del Pilar. 2009. *Metodología de la investigación.* México : McGrawHill Educación, 2009. 978-607-15-0291-9.

IBM. 2015. IBM. [En línea] 2015. [Citado el: 10 de Abril de 2015.] http://www-01.ibm.com.

ImageJ. 2014. ImajeJ.net. [En línea] ImajeJ, 2014. [Citado el: 9 de febrero de 2015.] http://www.imagej.net/.

Intel. 2013. OpenCV. *OpenCV.org.* [En línea] Intel, 2013. [Citado el: 9 de febrero de 2015.] http://www.opencv.org/.

International Biometric Group. 2006. Comparative Biometric Testing Round 6 Public Report. [En línea] 2006. http://www.nws-sa.com/biometrics/CBT6_public_report.pdf.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software.* España : Pearson Educación.

Larman, Craig. 1999. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* Mexico : Prentice Hall, 1999. 970-17-0261-1.

León Hernández, Rolando Alfredo y González Coello, Sayda. 2011. *El proceso de investigación científica.* La Habana : Editorial Universitaria, 2011. 978-959-16-1307-3.

Lora Pomar, Claudia y Delgado Mesa, Yisel. 2010. *Algoritmo de detección facial para sistemas de autenticación biométrica.* La Habana : Universidad de las Ciencias Informáticas, 2010.

Martin Díaz, Racielis y Quiroga Arencibia, José Ernesto. 2013. *Solución de autorización basada en el estándar XACML para el sistema de seguridad ACAXIA.* La Habana : Universidad de las Ciencias Informáticas, 2013.

Martínez Romero, Anisley y Companioni Sosa, Reynaldo Jesús. 2012. *Propuesta de Subsistema de control de acceso para el Sistema de Planificación y Control del Servicio de Aliment ación de la Universidad de las Ciencias Informáticas.* La Habana : Universidad de Las Ciencias Informáticas, 2012.

Microsoft Corporation. 2009. Microsoft TechNet Seguridad. [En línea] 2009. http://www.microsoft.com/latam/technet/seguridad/articulos/bpsegcorp.mspx.

Mozilla Developer Network. 2015. Mozilla Developer Network. *MDN.* [En línea] Mozilla Foundation, 2015. [Citado el: 2 de abril de 2015.] https://developer.mozilla.org/es/docs/JavaScript.

M'Raihi, D., y otros. 2011. *TOTP: Time-Based One-Time Password Algorithm.* s.l. : Internet Engineering Task Force (IETF), 2011. 2070-1721.

OMG. 2013. UML. [En línea] OMG, 2013. [Citado el: 9 de febrero de 2015.] http://www.uml.org.

Oryarzabal Arocena, Marcelo. 2015. Urrats bat. *Zentroak eraberritzea.* 2015, 44.

Pelaez González, Reinaldo. 2011. *Drivers de autenticación para el sistema de seguridad ACAXIA.* La Habana : Universidad de las Ciencias Informáticas, 2011.

Pérez San-José, Pablo. 2012. Guía sobre riesgos y buenas prácticas en autenticación online. [En línea] 2012. http://www.inteco.es/guias/Guia_Autenticacion.

Pfleeger, Charles P. 2006. *Security in computing.* 2006. ISBN: 978-0-13-239077-4.

PHP. 2009. Introduction: What is PHP? . *Php.net.* [En línea] PHP, 2009. [Citado el: 9 de febrero de 2015.] http://php.net/manual/es/intro-whatis.php.

PosgreSQL. 2009. Acerca de PostgreSQL. *ProgreSQL.es.* [En línea] PostgreSQL, 2009. [Citado el: 9 de febrero de 2015.] http://www.postgresql.org.es/sobre_postgresql.

Pressman, Roger S. 2010. *Ingeniería de Software, un enfoque práctico.* Madrid : Mc Graw Hill, 2010.

Rankl, Wolfgang y Effing, Wolfgang. 2010. *Smart Card Handbook: Fourth Edition.* Hoboken, New Jersey : Wiley John + Sons Inc., 2010. 978-0470743676.

Reyes Bermúdez, Enrique. 2009. *Sistema de autenticación para el módulo Seguridad del proyecto Guardián del ALBA.* La Habana : Universidad de las Ciencias Informáticas, 2009.

Reynoso, Carlos y Kiccillof, Nicolás. 2004. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft Versión.* Buenos Aires : Universidad de Buenos Aires, 2004.

Rodríguez Sánchez, Tamara. 2014. *Metodología de desarrollo para la Actividad productiva de la UCI.* La Habana : Universidad de las Ciencias Informáticas, 2014.

Silveiro Leyva, Osmayda y Sabuquet Hurtado, Ernesto. 2013. *Sistema de Identificación mediante Huella Dactilar.* La Habana : Universidad de las Ciencias Informáticas, 2013.

Sommerville, Ian. 2009. *Ingeniería de Software.* Madrid : Pearson Education, 2009. 84-7829-074-5.

Suhendra, V. 2011. *A Survey on Access Control Deployment.* Berlin : Springer Berlin Heidelberg 259, 2011. págs. 11-20.

Tapiador, Marino y Sigüenza, Juan Alberto. 2005. *Tecnologías biométricas aplicadas a la seguridad.* s.l. : RAMA, 2005. ISBN 13 / Cód Barra: 9788478976362.

TeknoTAG. 2015. TeknoTAG - Control de acceso a eventos y congresos. *Control de accesos - TeknoTAG.* [En línea] 2015. [Citado el: 1 de abril de 2015.] http://teknotag.com/index.php/control-de-accesos-eventos.

Torres Marquez, Joaquin. 2006. *Nuevo Marco de Autenticación para Tarjetas Inteligentes en Red. Aplicación al Pago Electrónico en entornos inalámbricos.* Madrid : Universidad Carlos III, 2006.

Treto Portal, Janier. 2013. *Modelo de identificación y autenticación basado en reconocimiento facial para el Sistema de Gestión Integral de Seguridad Acaxia.* La Habana : Universidad de las Ciencias Informáticas, 2013.

Villalón Huerta, Antonio. 2002. *Seguridad en UNIX y redes.* s.l. : Free Software Foundation, 2002.

Visual Guard. 2011. Visual Guard. *Visual Guard Enterprise Edition – Modo de autenticación mixta.* [En línea] Visual Guard, 2011. [Citado el: 8 de febrero de 2015.] http://www.visual-guard.com/SP/net-powerbuilder-aplicacion-seguridad-autenticacion-permiso-acceso-control-rbac/visual-guard-modo-mixto-autenticacion.html.

Wordpress. 2015. Definición de Java. *Definición.de.* [En línea] Wordpress, 2015. [Citado el: 9 de febrero de 2014.] http://definicion.de/java/.

GLOSARIO DE TÉRMINOS

Autenticación: Proceso mediante el cual una identidad se identifica en el sistema obteniendo unas credenciales (usuario, grupo), las cuales determinan los permisos de acceso a los recursos.

Código de barras: Arreglo que contiene información codificada en las barras y espacios del símbolo. Esta información puede ser leída por medio de dispositivos ópticos.

Código QR: Módulo para almacenar información en una matriz de puntos o en un código de barras bidimensional.

Factor de autenticación: Requisito fundamental para los sistemas informáticos, según el nivel de seguridad con que se desea proteger la información.

RFID: Sistema de identificación por radiofrecuencia usado por algunos dispositivos de hardware.

SAML: Estándar abierto que define un esquema XML para el intercambio de datos de autenticación y autorización.

Seguridad informática: Conjunto de métodos y herramientas destinados a proteger los bienes o activos informáticos de una institución.

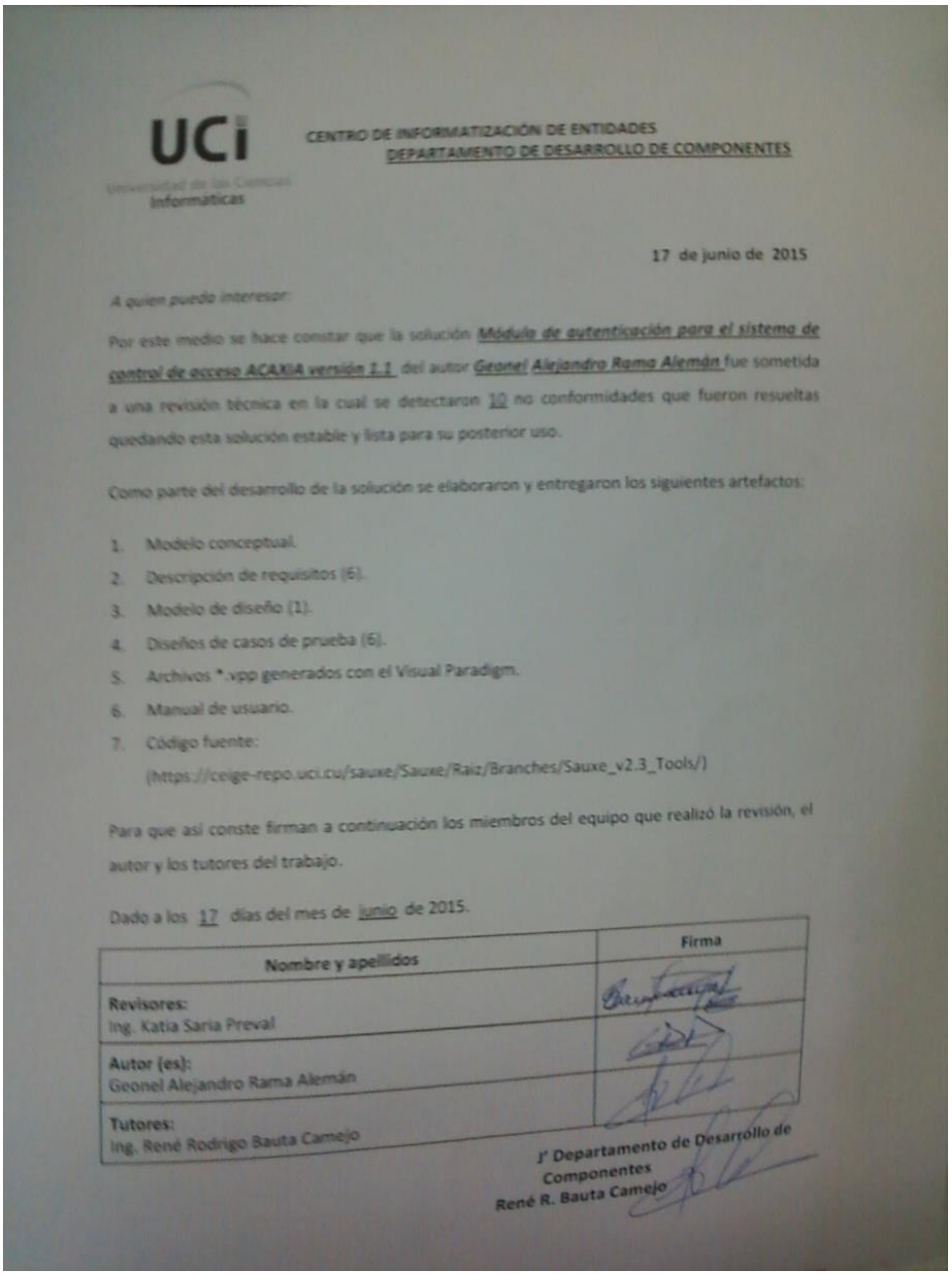
Single Sig-On: Arquitectura de sistemas que le permite al usuario acceder a diferentes aplicaciones con una sola validación de acceso.

Sistema de control de acceso: Aplicación destinada a implementar factores de autenticación, que definen responsabilidades y reglas a seguir, con el fin de minimizar los efectos que puedan traer consigo las amenazas e intentar prevenir posibles ataques.

Token: Dispositivo asignado a un usuario, con el fin de posibilitar su autenticación en un sistema determinado. Dispositivo de hardware que los usuarios cargan consigo para autorizar el acceso a un servicio de red.

ANEXOS

Anexo 1: Acta de liberación de calidad



UCI
Universidad de las Ciencias Informáticas

CENTRO DE INFORMATIZACIÓN DE ENTIDADES
DEPARTAMENTO DE DESARROLLO DE COMPONENTES

17 de junio de 2015

A quien pueda interesar:

Por este medio se hace constar que la solución ***Módulo de autenticación para el sistema de control de acceso ACAXIA versión 1.1*** del autor ***Geonel Alejandro Rama Alemán*** fue sometida a una revisión técnica en la cual se detectaron 10 no conformidades que fueron resueltas quedando esta solución estable y lista para su posterior uso.

Como parte del desarrollo de la solución se elaboraron y entregaron los siguientes artefactos:

1. Modelo conceptual.
2. Descripción de requisitos (6).
3. Modelo de diseño (1).
4. Diseños de casos de prueba (6).
5. Archivos *.vpp generados con el Visual Paradigm.
6. Manual de usuario.
7. Código fuente:
   (https://ceige-repo.uci.cu/sauxe/Sauxe/Raiz/Branches/Sauxe_v2.3_Tools/)

Para que así conste firman a continuación los miembros del equipo que realizó la revisión, el autor y los tutores del trabajo.

Dado a los 17 días del mes de junio de 2015.

Nombre y apellidos	Firma
Revisores: Ing. Katia Saria Preval	
Autor (es): Geonel Alejandro Rama Alemán	
Tutores: Ing. René Rodrigo Bauta Camejo	

J' Departamento de Desarrollo de Componentes
René R. Bauta Camejo

Figura 22 Anexo 1: Acta de liberación de calidad

Anexo 2: Encuesta realizada a los especialistas

ENCUESTA DE APOYO AL EXPERIMENTO DE VALIDACIÓN DEL OBJETIVO DE LA INVESTIGACIÓN

Con el objetivo de validar la idea a defender asociada a la versión 1.1 del módulo de autenticación para el sistema de control de acceso ACAXIA, se elabora un experimento, donde se le aplican pruebas de penetración de carga especializada al sistema ACAXIA utilizando las siguientes herramientas: Burp Suite Intruders para la medición del nivel de confiabilidad, SQLMap para la medición del nivel de integridad y HTTP Traffic para la medición del nivel de disponibilidad. Con el objetivo de apoyar la obtención de los datos durante el desarrollo del experimento se seleccionan 3 especialistas del departamento de desarrollo de componentes del centro CEIGE, los cuales permanecen durante todo el proceso con el objetivo de supervisar la correcta realización del mismo. Para la selección de estos especialistas se utilizan 3 criterios fundamentales: nivel científico, compromiso ético con la solución y experiencia previa en el trabajo con el módulo de autenticación. Luego de definir estos criterios, se elabora el listado de los especialistas:

- Ing. René Rodrigo Bauta Camejo, jefe del departamento de desarrollo de componentes del centro CEIGE.
- Ing. Katia Saria Preval, analista principal del departamento de desarrollo de componentes del centro CEIGE.
- Ing. Claudia Bravo Batista, desarrolladora del departamento de desarrollo de componentes del centro CEIGE.

Se decide aplicarle una encuesta a estos especialistas, con el objetivo de recolectar datos que hayan quedado fuera del alcance del experimento realizado.

PREGUNTAS

Pregunta 1. ¿Qué nivel de mejora en cuanto a garantizar la **confiabilidad** cree usted que representa el desarrollo de la versión 1.1 del módulo de autenticación con respecto a la versión anterior? _____

Pregunta 2. ¿Qué nivel de mejora en cuanto a garantizar la **integridad** cree usted que representa el desarrollo de la versión 1.1 del módulo de autenticación con respecto a la versión anterior? _____

Pregunta 3. ¿Qué nivel de mejora en cuanto a garantizar la **disponibilidad** cree usted que representa el desarrollo de la versión 1.1 del módulo de autenticación con respecto a la versión anterior? _____

Pregunta 4. ¿Cree usted que con la nueva versión del módulo de autenticación se aumenta el nivel de **seguridad** con respecto al anterior? Marque la respuesta que considere correcta: ___ Sí ___No

Nota: los valores cualitativos de las respuestas se asocian a niveles cuantitativos: INEFICIENTE (0-2), BAJO (3-5), MEDIO (6-8), ALTO (9-10).

_____ _____

ESPECIALISTA **FIRMA**

Figura 23 Anexo 2: Encuesta realizada a los especialistas