Universidad de las Ciencias Informáticas
FACULTAD 3

Título: Drivers de autenticación para el sistema de seguridad ACAXIA.

Trabajo de Diploma para optar por el título de

Ingeniero Informático

Autor(es): Reinaldo Pelaez González

Tutor(es):Msc. Oiner Gómez Baryolo

Ing. Daniel E. López Méndez

Ciudad de La Habana, Junio 2011

"Año 53 de la Revolución"

DECLARACIÓN DE AUTORÍA

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

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

Reinaldo Pelaez González **Oiner Gómez Baryolo**

_____ _____

Firma del Autor Firma del Tutor

AGRADECIMIENTOS

Desde lo más profundo de mi corazón: a mi mamá Yanelys González porque ha estado conmigo siempre en las buenas y en las malas, porque desde hace 23 años ha dado lo mejor de sí, porque es gracias a su fe, apoyo y sinceridad que he podido llegar hasta acá. Gracias por todo mamá.

A mi papá Reinaldo Peláez que me ha querido mucho y lo ha hecho a su manera, gracias por aconsejarme y darme el aliento de aprender más y más.

A mi tía querida Milagros González, mi otra madre, la que siempre ha estado allí en los buenos momentos pero también en los sacrificios que exige la vida.

Le agradezco a mi primo - hermano Adalbertico, a mis tíos Juan Carlos, Alfredo y a Israel, a Migue. A mi abuela-mamá Rafaela, mi tía Cusa, a mi tío Osvaldo y a mi abuelo-papá Rafael que no están físicamente pero siempre los llevaré en la memoria. A mi abuela-mamá María, a mi tío Omar, a Yudelmis, Osvaldito y a Joaquina. Le agradezco a mi familia que es mi refugio y mi hogar.

Le agradezco a la UCI, mis amigos y hermanos en especial a Grabiel, Oscar y el viejo Driggs, que espero volverlos a ver. Les agradezco a mis tutores Oiner y el chino por guiarme en este trabajo.

Gracias a todos los que de una manera u otra me apoyaron.

RESUMEN

Actualmente la seguridad informática es un aspecto fundamental en la vida de cualquier institución. La información representa el todo para las empresas y la protección de la misma es una tarea de vital importancia porque que la pérdida, mala manipulación o desvío de los datos ocasionan grandes daños económicos y sociales irreparables. Es por ello que se deben tomar medidas para proteger estos recursos. Entre las medidas más esenciales se encuentra el control de acceso lógico a los recursos.

En la actualidad, la tendencia que existe a nivel mundial para el control de acceso a la información es el uso de sistemas de seguridad que implementen varios niveles o factores de autenticación. En Cuba y específicamente en la Universidad de las Ciencias Informáticas se han aplicado estas soluciones pero cada aplicación gestiona su propia seguridad, lo cual ocasiona pérdida de tiempo y gastos innecesarios de cuantiosos recursos humanos y materiales. Además no se realiza una administración centralizada de los sistemas de seguridad, lo que trae consigo que sea muy difícil controlar que no ocurran violaciones en los sistemas y en caso de que ocurra poder detectarlas. Es por ello que en el centro se lleva a cabo el Sistema de Gestión Integral de la Seguridad Acaxia, el cual implementa un componente de autenticación regido por estándares internacionales como es el uso del protocolo SAML, la inclusión de una firma y un certificado digital y la implementación de una arquitectura de Single Sign-On. Esta versión tenía los inconvenientes que dichos procesos solo utilizaban un factor de autenticación y no lograba integrarse a otros sistemas que utilizaban técnicas generalizadas como el protocolo LDAP. El siguiente trabajo se centra en el proceso de autenticación, el mismo tiene como valor agregado la gestión centralizada de la autenticación en entornos multi-entidad y multi-sistema incorporando drivers de autenticación con código de barra y con LDAP al sistema Acaxia garantizando el control de acceso a la información. Para el desarrollo de la solución se realizó un estudio de los estándares internacionales más utilizados, estos aumentan la seguridad de los sistemas, la confianza de los usuarios, los niveles de integración con otras soluciones y el nivel de generalización. Con la utilización de los drivers se garantiza la seguridad del componente de autenticación, disminuyendo los costos, el tiempo de respuesta al usuario y el esfuerzo y aumentando la seguridad de las aplicaciones suscritas a Acaxia.

PALABRAS CLAVES:

Autenticación, LDAP, código de barra.

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

En la actualidad la seguridad informática juega un papel clave en el combate del delito cibernético; la tecnología, el software y con ello el malware[1]revolucionan constantemente, esto ocasiona que las grandes entidades a nivel mundial destinen cuantiosos recursos para mejorar sus sistemas de seguridad. La fortaleza de estos sistemas depende en gran medida de los componentes de autenticación que implementan, puesto que en estos se verifica si el usuario que trata de identificarse es válido, previniendo así que las áreas restringidas sean visitadas por intrusos.

La autenticación, aunque no debe ser el único mecanismo de seguridad implementado en un sistema, es uno de los más importantes, puesto que no hay duda que la mejor prevención ante un ataque es la de denegar el acceso desde un principio al atacante. Este proceso se ha perfeccionado en el transcurso de los años incorporando nuevas técnicas y distintos niveles de seguridad (autenticación multifactor) que autentican al usuario con una probabilidad muy baja de errores y hacen cada vez más difícil de burlar la seguridad de los sistemas.

La Universidad de Ciencias Informáticas creada por el pensamiento visionario del compañero Fidel es un centro de enseñanza superior de nuevo tipo, que ha tenido como uno de sus objetivos principales el impulso del software cubano. Para cumplir esta meta se ha trabajado constantemente para elevar el nivel de preparación en los temas de seguridad informática y aún con la escasa disponibilidad de recursos, se aplican iniciativas para elevar la seguridad del software desplegado y se diseñan componentes de autenticación combinando estrategias donde se utilizan las nuevas tecnologías.

Sin embargo un estudio realizado en la tesis “Propuesta de procesos de autenticación, autorización y auditoría (AAA) para aplicaciones basadas en servicios web XML” del Msc Karel Gómez determinó que las aplicaciones implantadas en la universidad en los proyectos donde existe manejo de la información implementan su propia seguridad, lo cual ocasiona pérdida de tiempo y gastos innecesarios de

[1] Programa maligno. Son todos aquellos programas diseñados para causar daños al hardware, software, redes,… como los virus, troyanos, gusanos… Es un término común que se utiliza al referirse a cualquier programa malicioso.

cuantiosos recursos humanos y materiales. Además no se realiza una administración centralizada de los sistemas de seguridad, lo que trae consigo que sea muy difícil controlar que no ocurran violaciones en los sistemas y en caso de que ocurra poder detectarlas. (1) Es por ello que se decide el desarrollo del Sistema de Gestión Integral de Seguridad (SIGIS) Acaxia, el cual incorpora las mejores prácticas de los sistemas de seguridad más utilizados en Cuba y el mundo, además está integrado a un sistema de estructura y composición que permite la gestión de estructuras dinámicamente, e incluye soluciones a escenarios no resueltos hasta el momento en sistemas de este tipo.

El SIGIS Acaxia implementa un componente de autenticación que está regido por el estándar internacional SAML, brindando la posibilidad de utilizar firma, certificados digitales y la implementación de una arquitectura de Single Sign-On. Pero actualmente la solución se ve limitada al no poder gestionar de manera centralizada los distintos tipos de autenticación de las aplicaciones suscritas al sistema. Estas aplicaciones no pueden definir sus propias fuentes de autenticación y la técnica utilizada para autenticar es mediante usuario y contraseña la cual tiene la desventaja que la clave puede ser revelada o forzada. En Acaxia están registradas aplicaciones que manejan información sensible y requieren mayor seguridad.

Por la problemática existente surge el siguiente **problema a resolver**: ¿Cómo aumentar los niveles de seguridad de las aplicaciones suscritas al SIGIS Acaxia?

Objeto de estudio: El proceso de autenticación en aplicaciones web de gestión.

Campo de acción: El componente de autenticación del sistema de seguridad Acaxia.

Hipótesis:

Si se implementan los drivers de autenticación mediante código de barra y LDAP se aumentarán los niveles de seguridad de las aplicaciones suscritas al SIGIS Acaxia.

Para precisar esta idea se tiene como **objetivo general** desarrollar drivers de autenticación que permitan gestionar los niveles de seguridad de las aplicaciones suscritas al SIGIS Acaxia.

Objetivos específicos:

1. Construir el marco teórico sobre los niveles de autenticación en las aplicaciones.
2. Realizar el diseño y la implementación de la solución.
3. Validar la solución propuesta.

El documento cuenta con la siguiente estructura:

Capítulo 1: Fundamentación teórica. Se realiza un estudio del estado del arte del tema de la autenticación, así como de las tecnologías usadas en este tema nacional e internacionalmente.

Capítulo 2: Este capítulo contiene la propuesta de solución, en su contenido abarca los procesos de negocios, los requisitos a cumplir por el componente además de los artefactos generados durante el diseño.

Capítulo 3: Se reflejan las pruebas realizadas y analizan los resultados obtenidos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

La autenticación en un sistema de gestión de seguridad centralizada tiene gran importancia. Actualmente en la universidad y en Cuba este proceso no se implementa de forma adecuada: cada sistema gestiona su seguridad de forma aislada, esto ocasión que aunque se han implementado técnicas de autenticación avanzadas, existan grandes amenazas de seguridad. Por lo que este trabajo está enmarcado en el desarrollo de drivers de autenticación para el SIGIS Acaxia. En este capítulo se describe el marco conceptual que existe alrededor de la autenticación en sistemas de gestión, las tecnologías y herramientas propuestas, así como las técnicas y los estándares existentes para llevar a cabo la solución.

1.1 Conceptos fundamentales

1.1.1 Drivers

Programa que determina cómo una computadora se comunica con un dispositivo periférico. (2)

Un Driver, o controlador, es un programa que controla un dispositivo. Cada dispositivo, ya sea una impresora, un teclado, entre otros, debe tener un programa controlador.

El autor considera que un driver es el programa destinado para integrar sistemas de diferentes tecnologías, logrando la comunicación entre las partes.

1.1.2 Control de acceso

Mecanismo que en función a la identificación ya autenticada permite acceder a datos o recursos. (3)

El control de acceso es el proceso de conceder permisos a usuarios o grupos de acceder a los recursos de un sistema. El control de acceso está basado en tres conceptos fundamentales: identificación, autenticación y autorización. Los controles de accesos son necesarios para proteger la confidencialidad, integridad y disponibilidad de los objetos, y por extensión de la información que contienen, pues permiten que los usuarios autorizados accedan solo a los recursos que ellos requieren para realizar sus tareas.

1.1.3 Identificación

La identificación no es más que la acción por parte de un usuario de presentar su identidad a un sistema, usualmente se usa un identificador de usuario. Establece además que el usuario es responsable de las acciones que lleve a cabo en el sistema, lo que se relaciona con los registros de auditorías que permiten guardar las acciones realizadas dentro del sistema y rastrearlas hasta el usuario autenticado. (4)

Acción de reconocer o probar que una persona o cosa es la misma que se busca o se supone. (5)

Demostración de que dos cosas son idénticas o equiparables. (5)

1.1.4 Autenticación

La autenticación es la verificación de que el usuario que trata de identificarse es válido, usualmente se implementa con una contraseña en el momento de iniciar una sesión. Existen cuatro tipos de técnicas que permiten realizar la autenticación de la identidad del usuario, las cuales pueden ser utilizadas individualmente o combinadas (autenticación de varios factores). (4)

La autenticación es el proceso de detectar y comprobar la identidad de una entidad de seguridad mediante el examen de las credenciales del usuario y la validación de las mismas consultando a una autoridad determinada. (6)

1.1.5 Autorización

La autorización es el procedimiento para determinar si el usuario o proceso previamente identificado y autenticado tiene permitido el acceso a los recursos. (4)

Autorización se refiere a conceder servicios específicos (entre los que se incluye la “negación de servicio”) a un determinado usuario, basándose para ellos en su propia autenticación, los servicios que está solicitando, y el estado actual del sistema. Es posible configurar restricciones a la autorización de determinados servicios en función de aspectos como, por ejemplo, la hora del día, la localización del usuario, o incluso la posibilidad o imposibilidad de realizar múltiples “logins” de un mismo usuario. (7)

1.2 Estándares de autenticación

1.2.1 Radius

RADIUS (acrónimo en inglés de Remote Authentication Dial-In User Server). Es un protocolo de autenticación y autorización para aplicaciones de acceso a la red o movilidad IP. Utiliza el puerto 1813 UDP para establecer sus conexiones. (8)

Cuando se realiza la conexión mediante módem[2], DSL[3], cable módem, Ethernet o Wi-Fi[4], se envía una información que generalmente es un nombre de usuario y una contraseña. Esta información se transfiere a un dispositivo Servidor de Acceso a la Red, en inglés Network Access Server (NAS), quien redirige la petición a un servidor RADIUS sobre el protocolo RADIUS. El servidor RADIUS comprueba que la información es correcta. Si es aceptado, el servidor autorizará el acceso al sistema y le asigna los recursos de red como una dirección IP, etc. (8)

Una de las características más importantes del protocolo RADIUS es su capacidad de manejar sesiones, notificando cuando comienza y termina una conexión, así que al usuario se le podrá determinar su consumo y facturar en consecuencia; los datos se pueden utilizar con propósitos estadísticos. (8)

El uso de este protocolo no es factible para la solución ya que su implementación es desarrollada para la comunicación en redes inalámbricas.

1.2.2 Kerberos

Kerberos funciona mediante la encriptación de datos con una clave simétrica. Una clave simétrica es un tipo de autenticación en el cliente y el servidor de acuerdo en utilizar una codificación concreta y clave de descifrado para enviar o recibir datos. Cuando se trabaja con la clave de cifrado, los detalles

[2]Un **módem** es un dispositivo que sirve para enviar una señal llamada portadora mediante otra señal de entrada llamada moduladora.

[3]**DSL** (Digital Subscriber Line) es un protocolo de banda ancha de internet.

[4]**Wi-Fi** es un sistema de envío de datos sobre redes computacionales que utiliza ondas de radio en lugar de cables, además es una marca de la *Wi-Fi Alliance* (anteriormente la *WECA: Wireless Ethernet Compatibility Alliance*), la organización comercial que adopta, prueba y certifica que los equipos cumplen los estándares 802.11.

son enviados a un centro de distribución de claves, o KDC, en lugar de enviar los datos directamente entre cada equipo. (9)

Desventajas

En primer lugar está la necesidad de que el servidor de Kerberos. Este servidor se encargará de todas las funciones necesarias para la autenticación. Si este servidor no se encuentra brindando sus servicios, nadie puede autenticarse, y por lo tanto, la red está abajo. Esto se puede prevenir mediante el uso de más de un servidor Kerberos, pero es más costoso de lo que algunas personas les gustarían pensar.

Se tiene la cuestión de la sincronización del reloj. Puesto que Kerberos utiliza marcas de tiempo para manejar todas las actividades, los relojes en todas las máquinas de acogida deben ser inferiores a 10 minutos de reloj del servidor de Kerberos. Dado que no todos los relojes son perfectos, el reloj de acogida y el reloj del servidor con el tiempo se desalinean lo suficiente para causar una falla. Normalmente, esto puede ser remediado por mantener los relojes hasta la fecha, o utilizar un Network Time Protocol, o NTP. (9)

1.2.3 SAML

SAML, desarrollado por el Servicio de Seguridad del Comité Técnico de OASIS, es un marco basado en XML para la comunicación de la autenticación de usuarios, el derecho, y la información de atributos. Como su nombre lo indica, SAML permite a las entidades empresariales hacer afirmaciones sobre la identidad, atributos y derechos de un sujeto (una entidad que es a menudo un usuario humano) a otras entidades, tales como una compañía asociada u otra aplicación de empresa. (10)

SAML permite crear declaraciones o afirmaciones sobre que cierto proceso de autenticación se ha llevado a cabo, pero no incluye como requisito ni especifica los procesos de autenticación en sí, tan sólo que se llevaron a cabo. Los consumidores de las aserciones de autenticación deberían ser cautos y no confiar ciegamente en estas afirmaciones a menos que conozcan los fundamentos sobre los que fueron hechas. La confianza en las aserciones nunca debe ser mayor que la confianza que llevó a crearla por parte de la entidad que la realizó.

Por qué usar SAML

- ✓ Independencia de Plataforma. SAML abstrae el framework de seguridad de arquitecturas de plataformas e implementaciones de soluciones SSO particulares. Haciendo que la seguridad sea más independiente de la lógica de las aplicaciones, esto es un principio importante de las Arquitecturas Orientadas a Servicios.

- ✓ Desacoplamiento de directorios. SAML no necesita información del usuario para mantenerse y sincronizarse entre directorios.

- ✓ Mejoramiento la experiencia en línea para los usuarios finales. SAML habilita SSO permitiéndole a los usuarios autenticarse ante el proveedor de identidades y posteriormente acceder a proveedores de servicios sin autenticación adicional. Adicionalmente, la federación de identidades (vinculación entre múltiples identidades) con SAML permite mejorar la experiencia al usuario de una manera personalizada en cada servicio promoviendo la privacidad del mismo.

- ✓ Reducción de costos administrativos para los proveedores de servicios. Usando SAML para reutilizar un único proceso de autenticación (como es el proceso de logueo con nombre de usuario y contraseña) varias veces entre varios servicios puede reducir el costo de mantener información de las cuentas. Esta carga es transferida al proveedor de identidades.

- ✓ Transferencia de riesgos. SAML puede actuar para transferir la responsabilidad de la administración adecuada de identidades al proveedor de identidades, que con más frecuencia es compatible con su modelo de negocio que en los proveedores de servicios. (11)

1.3 Técnicas de identificación y autenticación

Técnicas para realizar autenticación de identidad de usuarios:

1. Algo que solamente el individuo conoce: por ejemplo una contraseña.

2. Algo que la persona posee: por ejemplo una tarjeta magnética.

3. Algo que el individuo es y que lo identifica unívocamente: por ejemplo las huellas digitales.

4. Algo que solo el individuo es capaz de hacer: por ejemplo los patrones de escritura. (4)

Estas técnicas pueden ser utilizadas individualmente o combinadas que es la autenticación con varios factores.

La fortaleza de la autenticación es mayor mientras más factores se adicionen, generalmente solo se utilizan hasta 3 factores:

- 1 factor = contraseña
- 2 factores = contraseña + token[5]
- 3 factores = contraseña + token + biometría
- 4 factores = contraseña + token + biometría + localización geográfica (GPS)
- 5 factores = contraseña + token + biometría + localización geográfica + perfil de usuario. (4)

1.3.1 Sistemas basados en algo conocido: Contraseñas

El modelo de autenticación más básico consiste en decidir si un usuario es quien dice ser simplemente basándonos en una prueba de conocimiento que a priori sólo usuario puede superar; esa prueba de conocimiento no es más que una contraseña que en principio es secreta. Evidentemente, esta aproximación es la más vulnerable a todo tipo de ataques, pero también la más barata, por lo que se convierte en la técnica más utilizada en entornos que no precisan de una alta seguridad. (3)

La seguridad de las contraseñas se ve afectada por los siguientes factores:

- Fortaleza de la contraseña: deben ser largas, normalmente más de 7 caracteres, y se deben usar combinaciones de letras mayúsculas y minúsculas, números y símbolos. Ejemplos de contraseñas fuertes serían las siguientes: tastY=wheeT34, pArtei@34! y #23kLLflux.
- Formas de almacenar las contraseñas: se debe usar un algoritmo criptográfico irreversible (o función resumen), los más comunes son MD5 y SHA1.
- Método de retransmisión de la contraseña: deben ser transmitidas mediante algún método criptográfico, en el caso de las redes locales se usa con mucha frecuencia Kerberos.

5 Un token de seguridad es un pequeño dispositivo de hardware que los usuarios cargan consigo para autorizar el acceso a un servicio de red. El dispositivo puede ser en forma de una tarjeta inteligente o puede estar incorporado en un objeto utilizado comúnmente, como un llavero.

- Longevidad de la contraseña: deben ser cambiadas con cierta periodicidad. (4)

1.3.2 Sistemas de autenticación biométricos

La biometría aplicada al campo de la seguridad consiste en la utilización de tecnologías que hacen uso de métodos automáticos para el reconocimiento de seres humanos basados en uno o más rasgos físicos intrínsecos. El término se deriva de las palabras griegas bios, vida y metron, medida. El propósito de las tecnologías biométricas es fundamentalmente la identificación y la autentificación en control de accesos. (12)

Los sistemas biométricos, podemos clasificarlos en:

- ✓ Fisiológicos: huella dactilar, iris, retina, cara, geometría de la mano, huella palmar, estructura de las venas, estructura de la oreja, termografía facial.
- ✓ Conductuales: voz, escritura, firma manuscrita, modo de teclear, modo de andar.

Figura 1 Autenticación por biometría. De izquierda a derecha ejemplo distintos tipos de reconocimientos: huella dactilar, facial, forma de la mano, iris, retina, firma.

Características a tener en cuenta en la elección de las técnicas de autenticación biométrica en general:

- ✓ Necesidad de un dispositivo de adquisición específico (lector de huella dactilar, micrófono, cámara, etc.) allí donde esté el usuario.
- ✓ Posible variabilidad con el tiempo del patrón a identificar (afonías o catarros en voz, uso de gafas/bigote/barba/etc. en rostro, etc.).

- ✓ Aceptación por parte del usuario de cada una de las técnicas, en función de si son o no técnicas intrusivas, cómodas, que mantengan (o al menos lo parezca) la privacidad, sencillas de usar, etc.

Las principales razones por la que estos sistemas no se han impuesto de forma generalizada es su elevado precio, fuera del alcance de muchas organizaciones, y su dificultad de mantenimiento.

1.3.3 Sistemas basados en algo que la persona posee

A diferencia de los sistemas basados en algo conocido, este sistema requiere de un objeto físico, o sea, un pequeño dispositivo de hardware que los usuarios cargan consigo para autorizar el acceso a un servicio de red. El dispositivo o token puede ser en forma de tarjeta o puede estar incorporado en un objeto utilizado comúnmente, como un llavero. Los tokens proveen un nivel de seguridad adicional utilizando el método conocido como autenticación de dos factores: el usuario tiene un número de identificación personal (PIN), que le autoriza como el propietario del dispositivo; luego el dispositivo despliega un número que identifica en forma única al usuario ante el servicio, permitiéndole ingresar.

Las tarjetas pueden ser de banda magnética, tarjetas de código de barras, tarjetas inteligentes entre otras.

Tarjetas de banda magnética

La seguridad de las bandas magnéticas es prácticamente nula ya que no garantiza la confidencialidad ni la integridad de los datos que contiene. De hecho los datos pueden ser leídos con cualquier lector que los escribe en el Bloc de notas de nuestro ordenador, y pueden ser modificados con cualquier grabador magnético siguiendo unas reglas marcadas por varias ISO (ISO/IEC 7810, ISO/IEC 7811, ISO/IEC 7812, ISO/IEC 7813, ISO 8583, and ISO/IEC 4909). (13)

Estos problemas de seguridad han provocado que las entidades bancarias cada vez más dejen de utilizar la banda magnética como medio de identificación para los pagos. (13)

Por lo general la durabilidad de las bandas magnéticas ronda los 2 años. Pero por la propia tecnología en la que se basa su funcionamiento hay que tener especial cuidado para que no se deterioren tanto a nivel físico como a nivel interno de los datos que guardan. Cualquier campo electromagnético lo suficientemente potente podría alterar los datos que se almacenan en las bandas magnéticas. (13)

Tarjetas inteligentes

Desde un punto de vista formal (14), una tarjeta inteligente (o smartcard) es un dispositivo de seguridad del tamaño de una tarjeta de crédito, resistente a la adulteración, que ofrece funciones para un almacenamiento seguro de información y también para el procesamiento de la misma en base a tecnología VLSI. En la práctica, las tarjetas inteligentes poseen un chip empotrado en la propia tarjeta que puede implementar un sistema de ficheros cifrado y funciones criptográficas, y además puede detectar activamente intentos no válidos de acceso a la información almacenada; este chip inteligente es el que las diferencia de las simples tarjetas de crédito, que solamente incorporan una banda magnética donde va almacenada cierta información del propietario de la tarjeta. (15)

Ventajas y desventajas

Las ventajas de utilizar tarjetas inteligentes como medio para autenticar usuarios son muchas frente a las desventajas; se trata de un modelo ampliamente aceptado entre los usuarios, rápido, y que incorpora hardware de alta seguridad tanto para almacenar datos como para realizar funciones de cifrado. Además, su uso es factible tanto para controles de acceso físico como para controles de acceso lógico a los hosts, y se integra fácilmente con otros mecanismos de autenticación como las contraseñas; y en caso de desear bloquear el acceso de un usuario, no tenemos más que retener su tarjeta cuando la introduzca en el lector o marcarla como inválida en una base de datos (por ejemplo, si se equivoca varias veces al teclear su PIN, igual que sucede con una tarjeta de crédito normal). Como principal inconveniente de las smartcards podemos citar el coste adicional que supone para una organización el comprar y configurar la infraestructura de dispositivos lectores y las propias tarjetas; aparte, que un usuario pierda su tarjeta es bastante fácil, y durante el tiempo que no disponga de ella o no puede acceder al sistema, o hemos de establecer reglas especiales que pueden comprometer nuestra seguridad (y por supuesto se ha de marcar como tarjeta inválida en una base de datos central, para que un potencial atacante no pueda utilizarla). También la distancia lógica entre la smartcard y su poseedor - simplemente nos podemos fijar en que la tarjeta no tiene un interfaz para el usuario - puede ser fuente de varios problemas de seguridad (16) (17).

Tarjetas con código de barra

La primera patente de código de barras fue registrada en octubre de 1952 por los inventores Joseph Woodland, Jordin Johanson y Bernard Silver en Estados Unidos, estudiantes de la Facultad de Tecnología de Drexel (Filadelfia), a petición del dueño de una tienda de comestibles que quería

gestionar su almacén de forma automática. La implementación fue posible gracias al trabajo de los ingenieros Raymond Alexander y Frank Stietz. El resultado de su trabajo fue un método para identificar los vagones del ferrocarril utilizando un sistema automático. Sin embargo, no fue hasta 1966 que el código de barras comenzó a utilizarse comercialmente y no fue un éxito comercial hasta 1980. (18)

Figura 2 Estructura de una tarjeta con código de barra

1: Zona tranquila.

2: Carácter inicio (derecha), Carácter terminación (izquierda).

3: Carácter de datos.

4: Suma de comprobación.

El código de barra es una técnica de entrada de datos, con imágenes formadas por combinaciones de barras y espacios paralelos de anchos variables. Representan números que pueden ser leídos y descifrados por lectores ópticos y scanners. Sirven para identificar los recursos de forma única. Pues cuentan con información específica, del tipo de objeto que guardan a través de una asociación con una base de datos determinada.

Funcionamiento

La información contenida en la tarjeta puede ser leída por dispositivos ópticos, los cuales envían la información leída hacia una computadora como si se hubiera tecleado.

La información, se procesa y almacena con base en un sistema digital binario donde todo se resume a sucesiones de unos y ceros. (19)

Ventajas del código de barras:

Poseen 2 ventajas básicas sobre la entrada manual de datos: velocidad y precisión.

Por cada 12 caracteres de datos, la entrada por teclado tarda aproximadamente 6 segundos mientras que el escaneo de un código de barras de 12 caracteres demora aproximadamente 3 segundos.

1.4 Servicios de directorios

Un directorio es una base de datos optimizada para lectura, navegación y búsqueda. Los directorios tienden a contener información descriptiva basada en atributos y tienen capacidades de filtrado muy sofisticada. Los directorios generalmente no soportan transacciones complicadas ni esquemas de vuelta atrás como los que se encuentran en los sistemas de bases de datos diseñados para manejar grandes y complejos volúmenes de actualizaciones. Las actualizaciones de los directorios son normalmente cambios simples, o todo o nada, siempre y cuando estén permitidos. Los directorios están afinados para dar una rápida respuesta a grandes volúmenes de búsquedas. Estos tienen la capacidad de replicar la información para incrementar la disponibilidad y la fiabilidad, al tiempo que reducen los tiempos de respuesta. Cuando la información de un directorio se replica, se pueden producir inconsistencias temporales entre las réplicas mientras esta se está sincronizando. (20)

1.4.1 LDAP

LDAP ("Lightweight Directory Acces Protocol", en español Protocolo Ligero de Acceso a Directorios) es un protocolo de tipo cliente-servidor para acceder a un servicio de directorio.

Se usó inicialmente como un Front-end o interfaz final para x.5006, pero también puede usarse con servidores de directorio únicos y con otros tipos de servidores de directorio. (21)

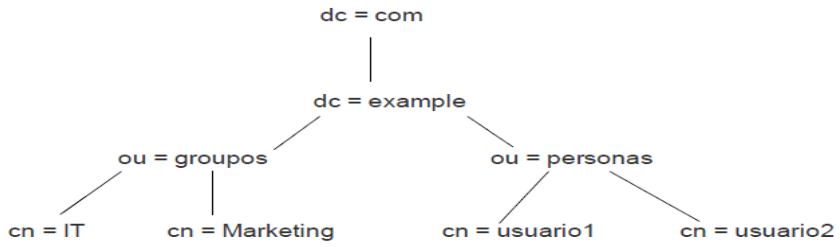
LDAP aparece con el estándar de los directorios de servicios. La versión original fue desarrollada por la Universidad de Michigan. La primera versión no se usó y fue en 1995 cuando se publicaron los RFC (Request For Comments) de la versión LDAPv2. Los RFC para la versión LDAPv3 fueron publicados en 1997. La versión 3 incluía características como las listas de acceso (Control Access Lists) y replicación de directorios.

Estructura

[6]El servicio de directorio X.500 es un servicio de directorio global. Sus componentes cooperan para gestionar la información sobre los objetos como los países, organizaciones, personas, máquinas, y así sucesivamente en un ámbito de aplicación en todo el mundo. Proporciona la capacidad de buscar información por nombre (un servicio de páginas blancas) y para navegar y buscar información (un servicio de páginas amarillas).

La estructura de un directorio de LDAP, como puede verse en la figura, se presenta como un árbol y utiliza los niveles siguientes:

- ✓ dc: domaincomponent , componente de dominio, no es único
- ✓ ou: organizationalunit , unidad de la organización
- ✓ cn: commonname, nombre común (por ejemplo nombre de usuario)



Aquí, en este árbol, se han creado dos usuarios (usuario1, usuario2) y dos grupos (IT, Marketing) dentro del dominio example.com

Cada entidad puede definirse con varios parámetros. El atributo member permite definir cuáles son los usuarios que pertenecen a él. Por ejemplo se puede definir el grupo IT como en el listado siguiente, en este caso, solo el usuario2 (línea 4) es miembro de este grupo.

1. dn: cn=IT,ou=grupos,dc=example,dc=com
2. objectclass: accessGroup
3. cn: IT
4. member: cn=usuario2, ou=personas,dc=example,dc=com

Funcionamiento

LDAP define operaciones para interrogar y actualizar el directorio. Provee operaciones para añadir y borrar entradas del directorio, modificar una entrada existente y cambiar el nombre de una entrada. La mayor parte del tiempo, sin embargo, LDAP se utiliza para buscar información almacenada en el directorio. Las operaciones de búsqueda de LDAP permiten buscar entradas que concuerdan con algún criterio especificado por un filtro de búsqueda. La información puede ser solicitada desde cada entrada que concuerda con dicho criterio. (20)

El servicio de directorio de LDAP está basado en el modelo cliente/servidor. Uno o más servidores LDAP contienen los datos que conforman la información del árbol del directorio[7] (DIT). El cliente se conecta a los servidores y les formula preguntas. Los servidores responden con una respuesta o con un puntero donde el cliente puede obtener información adicional (normalmente otro servidor LDAP). No importa a que servidor LDAP se conecte un cliente, éste siempre obtendrá la misma visión del directorio; un nombre presentado por un servidor LDAP referencia la misma entrada que cualquier otro servidor LDAP. Esta es una característica muy importante del servicio global de directorio, como LDAP. (20)

Uso de Distintos Niveles de Verificación

Dentro del uso del LDAP, se pueden estructurar varios niveles de jerarquización para la autorización y autenticación para la gestión de datos o el acceso a recursos. Dentro de la definición de los niveles, se puede definir los siguientes:

a) Simple Autenticación Usuario /Contraseña

b) Usando certificación electrónica.

c) Usando filtrado de Seriales sobre certificados.

Ventajas de LDAP

La mayor ventaja de LDAP es que se puede consolidar información para toda una organización dentro de un repositorio central. Por ejemplo, en vez de administrar listas de usuarios para cada grupo dentro de una organización, puede usar LDAP como directorio central, accesible desde cualquier parte de la red. Puesto que LDAP soporta la Capa de conexión segura (SSL) y la Seguridad de la capa de transporte (TLS), los datos confidenciales se pueden proteger de los curiosos. (22)

LDAP también soporta un número de bases de datos back-end en las que se guardan directorios. Esto permite que los administradores tengan la flexibilidad para desplegar la base de datos más indicada para el tipo de información que el servidor tiene que diseminar. También, ya que LDAP tiene una interfaz de programación de aplicaciones (API) bien definida, el número de aplicaciones acreditadas para LDAP son numerosas y están aumentando en cantidad y calidad. (22)

[7]En inglés: directory information tree

Principales implementaciones de LDAP

- **OpenLDAP**

El proyecto OpenLDAP nació como la continuación de la versión 3.3 del servidor LDAP de la Universidad de Michigan cuando dejaron de desarrollarlo.
OpenLDAP es un servidor LDAP que se distribuye bajo licencia GNU (OpenSource), que permite que el software se pueda usar de forma gratuita tanto de forma educativa como profesional. Además se dispone del código fuente para poder realizar modificaciones.

- **Active Directory**

Active Directory es una implementación propietaria (creada por Microsoft) de los Servicios de Directorio, y proporciona una manera de compartir información entre recursos y usuarios de la red. Además de proporcionar una fuente centralizada para esa información, Active Directory también funciona como autoridad de seguridad centralizada de autenticación para la red.
Active Directory combina capacidades que tradicionalmente se hallaban en sistemas separados y especializados de directorio, como integración simplificada, gestión y seguridad de los recursos de la red. El paquete SAMBA[8] puede configurarse para usar los servicios de Active Directory desde un controlador de dominio de Windows.

Diferencias entre OpenLDAP y Active Directory

Existen algunos campos que presentan diferencias en los esquemas de estas dos implementaciones de LDAP:

Tabla 1 Diferencias entre OpenLDAP y Active Directory

OpenLDAP	Active Directory
dn	dn
uid	samaccountname
displayname	displayname
givenname	givennanme

[8]Samba es una implementación libre del protocolo de archivos compartidos de Microsoft Windows (antiguamente llamado SMB, renombrado recientemente a CIFS) para sistemas de tipo UNIX

1.5 Soluciones para la autenticación a nivel mundial

Una de las soluciones de autenticación que destaca a nivel mundial es SecureAuth. Esta es una aplicación de la Plataforma de Identidad (IEP) que asegura y simplifica el acceso a cada VPN[9] y aplicaciones web con autenticación integrada, SSO, acceso y servicios de gestión de usuarios. SecureAuth ofrece configurar fácilmente las opciones de SAML que eliminan el costo y la experiencia necesarios para integrar una solución SAML en su directorio.

Entre sus características principales se encuentran las siguientes:

- ✓ Ofrece un método de autenticación algorítmicamente demostrado que impide los ataques de phishing "man-in-the-middle.
- ✓ Autenticación de 2 factores (certificados, SMS, telefonía, Nombre de usuario / contraseña) es configurable para el nivel adecuado de autenticación basado en grupo de usuarios del mapa, las aplicaciones de acceso, y las políticas de seguridad.
- ✓ Auditoría de Primera Instancia de todos los inicios de sesión del usuario, independientemente de si la solicitud se encuentra alojado reduce auditoría fracaso.
- ✓ Se integra con Microsoft Active Directory para la identidad utilizada para la autenticación es el mismo que se utiliza en la solicitud de procesamiento, los permisos y la administración de funciones.

Otra solución importante actualmente es Google Apps, la cual aumenta significativamente la seguridad de la nube con la verificación de dos pasos.

La verificación de dos pasos es fácil de configurar, administrar y utilizar. Cuando está habilitado por un administrador, se requiere de dos medios de identificación para acceder a una cuenta de Google Apps, algo que usted sabe: una contraseña, y algo que usted tiene: un teléfono móvil.

El usuario después de introducir su contraseña, se le envía un código de verificación al teléfono móvil a través de SMS, llamadas de voz, o se generan en una aplicación que se puede instalar en su Android, BlackBerry o iPhone. Esto hace que sea mucho más probable que el usuario sea el único que tenga acceso a sus datos: incluso si alguien ha robado la contraseña, necesitará más que eso para acceder a su cuenta. También puede indicar cuando se está usando un equipo de confianza y no desea que se le dé un código de verificación de esa máquina en el futuro.

[9] Una red privada virtual (VPN) es una forma segura de conectarse a una red de área local privada en una ubicación remota, a través de Internet o cualquier otra red pública no segura para el transporte de los paquetes de datos de la red privada, mediante el cifrado.

Además, Google Apps ofrece un servicio de inicio de sesión único basado en SAML que ofrece a las organizaciones un control total de la autorización y de la autenticación de cuentas de usuario alojadas que pueden acceder a aplicaciones basadas en web como Gmail o Google Calendar. Con el modelo SAML, Google actúa como proveedor y ofrece servicios como Gmail y páginas de inicio. Los partners de Google actúan como proveedores de identidad y administran los nombres de usuario, las contraseñas y otra información para identificar, autenticar y autorizar a los usuarios en aplicaciones web alojadas por Google. (23)

1.6 Soluciones para la autenticación en la universidad

En la facultad 10 se llevó a cabo el desarrollo de un software de autenticación y control centralizado para la corporación de PDVSA capaz de mapear la credenciales de los usuarios hacia todas las aplicaciones de dicho sistema y otros sistemas heterogéneos, proporcionando una infraestructura para simular un login único para el usuario corporativo frente a una plataforma tecnológica heterogénea dentro del ambiente de aplicativos de integración, en que los usuarios puedan acceder a diferentes aplicaciones con solo un conjunto de credenciales. Está aplicación trajo como resultado principal la importancia de mapeos entre diferentes entornos que manejan diferentes mecanismos de seguridad. (24)

En la universidad, prácticamente desde sus inicios, se han implantado los sistemas de autenticación mediante código de barra: cada estudiante y trabajador de la UCI cuenta con un solapín (tarjeta con foto del individuo y código de barra), que lo identifica, los comedores por ejemplo utilizan este sistema y han logrado con ello el control del acceso de manera eficaz y la agilización del paso de los usuarios hacia el área controlada. Otros lugares donde se han aplicado esta tecnología es en el control de acceso a los laboratorios de producción, ya que la información que en ellos se almacena y los medios que están presentes necesitan una mayor protección.

La universidad cuenta además con un servicio de directorio LDAP para su intranet, el servidor con el dominio uci.cu registra todas las personas vinculadas al centro (trabajadores y estudiantes). Es ampliamente utilizado y a él se incorporan todos los sistemas que requieran de este tipo de autenticación. Aunque están implementadas estas técnicas de autenticación las aplicaciones no están integradas a un sistema de seguridad que gestione los distintos niveles de autenticación.

1.6.1 Solución de ACAXIA

El componente de autenticación de ACAXIA utiliza el estándar SAML y brinda importantes funcionalidades, como son la inclusión de la firma y el certificado digital así como la implementación de una arquitectura de Single Sign-On. La solución está realizada sobre Zend Framework.

En su nivel más simple, Zend Framework es una librería de componentes escritos en PHP5, para facilitar el desarrollo de sitios web. Basada en PHP5 (5.1.4 es la versión mínima necesaria), es completamente Orientada a Objetos. (24)

Zend_Auth_Adapter_Ldap admite la autenticación de aplicaciones web con los servicios de LDAP. Sus características incluyen nombre de usuario y canonización de nombre de dominio, la autenticación de varios dominios, y las capacidades de conmutación por error. No solo ha sido probado para trabajar con Microsoft Active Directory y OpenLDAP, sino que también debe trabajar con otros proveedores de servicio LDAP. (25)

Aunque Zend Framework presenta esta característica, Acaxia no está integrada a la tecnología de LDAP, la autenticación se realiza utilizando una base datos.

Existen sistemas y aplicaciones que manejan información sensible por lo que requieren mayor seguridad, estos sistemas pueden tener uno o varios tipos de autenticación y su propio mecanismo de autorización o sea que no requieren estar registrados en Acaxia, solo que se les brinde el servicio de autenticación utilizando la fuente previamente establecida. Pero actualmente Acaxia no puede gestionar los distintos tipos de autenticación de las aplicaciones a las que brinda su seguridad y estas aplicaciones no pueden gestionar sus propias fuentes de autenticación.

1.7 Valoración de las técnicas y soluciones estudiadas

El estudio refleja que para aumentar los niveles de seguridad de las aplicaciones se debe utilizar autenticación con varios factores: entre más factores sean incorporados al proceso de autenticación de un sistema más elevado será su nivel de seguridad.

Entre las técnicas o factores estudiados, el sistema basado en contraseñas es el más vulnerable, por lo que para fortalecer la seguridad de una organización que implemente este sistema se hace necesaria la integración de otras técnicas de autenticación.

Por otra parte la autenticación utilizando técnicas biométricas aumenta la seguridad de un sistema pero el costo de instalación y mantenimiento de esta tecnología la hace inalcanzable para las organizaciones en países subdesarrollados como Cuba.

Se llega a la conclusión además, que no es factible utilizar tarjetas magnéticas por la poca seguridad que brindan; las tarjetas inteligentes por su parte aumentan considerablemente la seguridad sin embargo el propio chip que incorporan las hace vulnerables en cuanto a la durabilidad y por tanto aumenta el costo general al reponerle la tarjeta al usuario.

La tarjeta con código de barra es la tecnología más acorde a las circunstancias actuales. La universidad tiene instalado este sistema y el hecho de que cada usuario (trabajador o estudiante) cuente con un solapín es un punto importante para la elección de un sistema basado en código de barra.

La autenticación utilizando LDAP arroja grandes ventajas ya que es una técnica barata y fácil de incorporar a un sistema de gestión. Su estudio refleja que con la aplicación de esta técnica se logra ordenar, manejar y centralizar la información disminuyendo redundancia y facilitando los trabajos de administración. Por otra parte el estudio de las soluciones de autenticación existentes a nivel mundial muestra la tendencia actual de aumentar los niveles de seguridad de los sistemas de autenticación. Estas soluciones son privadas y para el país constituyen una tecnología imposible de incorporar por su elevado costo.

Todo esto demuestra que se hace necesario el desarrollo de drivers de autenticación para el sistema de Seguridad Acaxia, de manera que al incorporar las nuevas técnicas de autenticación aumenten con ello los niveles de seguridad de las aplicaciones suscritas al SIGIS Acaxia. El desarrollo de esta solución se ejecutará sobre el marco de trabajo Sauxe guiado por el modelo de desarrollo propuesto para componentes de este tipo.

1.8 Metodología de desarrollo

La selección de una metodología para el desarrollo de un determinado sistema está condicionada por varios factores que definen la más apropiada, entre ellos se encuentran, el tipo de sistema a desarrollar, el personal participante, los recursos disponibles y las necesidades del cliente, por lo que no existe una metodología estándar a seguir, más bien se necesita que este proceso sea adaptable y configurable de acuerdo al proyecto. Existen dos tipos fundamentales de metodologías para tratar este

tema, las que siguen los métodos tradicionales, que son generalmente para software de gran envergadura, y las que proponen procesos ágiles para el desarrollo, más usadas en softwares pequeños y con marcadas diferencias entre sí**.** Para este caso se centra el estudio de las metodologías ágiles dado el tipo de sistema en cuestión, y poniendo en práctica el modelo de desarrollo utilizado en el proyecto, descrito en el trabajo de diploma de los Ingenieros en Ciencias Informáticas Sergio Hernández Cisneros y Mileidy Magalys Sarduy Pérez. (26)

1.9 Tecnologías y Herramientas para el desarrollo

El desarrollo de este sistema estará guiado y dirigido principalmente por el modelo de desarrollo descrito en el trabajo de diploma de los Ingenieros en Ciencias Informáticas Sergio Hernández Cisneros y Mileidy Magalys Sarduy Pérez, basado en el estudio de metodologías ágiles y donde se recogen todos los artefactos a generar que necesita el proyecto para el desarrollo de un software.

Las herramientas y tecnologías a usar para la construcción del software se les hacen mención a continuación.

1.9.1 Apache

El servidor HTTP Apache es un servidor web de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA[10] Http 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Brian Behelendorf: quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de Estados Unidos y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de Internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA .Era, en inglés, a patchy server (un servidor "parcheado"). El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. (27)

1.9.2 PostgreSQL 8.3

[10]El NCSA HTTPd era un Servidor WEB desarrollado originalmente en el National Center forSupercomputingApplications por Robert McCool y una lista de colaboradores

PostgreSQL se ha beneficiado también de su larga historia. Hoy en día, PostgreSQL es uno de los servidores de base de datos más avanzados disponibles. Aquí están algunas de las características que se encuentran en una distribución estándar de PostgreSQL:

- Objeto-relación: En PostgreSQL, cada tabla define una clase. PostgreSQL implementa la herencia entre tablas(o, si se quiere, entre las clases). Funciones y operadores son polimórficos.

- Código abierto: Un equipo internacional de desarrolladores de PostgreSQL realiza el mantenimiento. Una de las ventajas de la naturaleza de PostgreSQL de código abierto es que el talento y el conocimiento pueden ser contratados, según sea necesario. El hecho de que este equipo es internacional asegura que PostgreSQL es un producto que puede ser utilizado de manera productiva en cualquiera de las lenguas naturales, no sólo inglés.

- PostgreSQL soporta el desarrollo de aplicaciones cliente en varios idiomas. Actualmente PostgreSQL se puede conectar a C, C ++, ODBC, Perl, PHP, Tcl/Tky Python. (28)

1.9.3 Herramientas CASE

CASE corresponde a las iniciales de: Computer Aided Software Engineering; y en su traducción al Español significa Ingeniería de Software Asistida por Computación.

Las Herramientas CASE son el conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante la investigación preliminar, análisis, diseño, implementación e instalación de un Software.

1.9.4 Visual Paradigm

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (29)

1.10 Herramientas para el desarrollo colaborativo

Este tipo de herramientas permiten el desarrollo en conjunto, logrando la comunicación, integración del equipo y el control de versiones durante el desarrollo del software.

1.10.1Subversion

Subversion es un software gratis y de código abierto de sistema de control de versiones, en inglés version control system (VCS). Subversion maneja ficheros y directorios, y los cambios introducidos en ellos, con el tiempo. Esto le permite recuperar versiones antiguas de sus datos o examinar la historia de cómo cambiaron sus datos. Subversion puede operar a través de redes, que permite que sea utilizado por personas en diferentes equipos. En algún nivel, la capacidad para que varias personas puedan modificar y administrar el mismo conjunto de datos de sus respectivas ubicaciones fomenta la colaboración. (30)

1.11 FrameWork de desarrollo

FrameWork es un concepto sumamente genérico, se refiere a “ambiente de trabajo, y ejecución”, por ejemplo “.Net” es considerado un “framework” para desarrollar aplicaciones (Aplicaciones sobre Windows). En general los framework son soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución). (31)

1.11.1 Zend Framework

Zend está formado por muchos componentes, estos están divididos en grupos uno de estos grupos se encarga de gestionar la seguridad, los componentes de este grupo son: Zend_Auth permite chequear y guardar credenciales de usuario de distintas maneras: utilizando la Base de Datos, el método Diguest de Apache, o autenticación http simple. Provee un Adapter de Interfaz para personalizar los mecanismos de autenticación, además almacenamiento automático de identidad para una fácil personalización. Es simple y extensible. A su vez Zend_Session trabaja como un administrador de datos de sesión, al igual que en PHP, solo que ofrece algo de valor agregado. (24)

1.11.2 Marco de trabajo Sauxe

Sauxe es un Marco de Trabajo que contiene un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo. (32)

1.11.3 Doctrine

Doctrine es un potente y completo sistema ORM (object relational mapper) para PHP 5.2+ con un DBAL (database abstraction layer) incorporado. Se está empezando a ver su potencial, pero de la documentación se puede decir que tiene todas las características necesarias para ser funcional en casi cualquier proyecto. Entre muchas otras funcionalidades tiene la posibilidad de exportar una base de datos existente a sus clases correspondientes y también inversa, es decir convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos. Por otro lado, como la librería es bastante grande, ésta tiene un método para ser "compilada" al pasar a producción. (33)

1.11.4 ExtJS

ExtJS es una librería Javascript que permite construir aplicaciones complejas en Internet. Esta librería incluye:

- Componentes UI del alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API fácil de usar.
- Licencias Open source y comerciales. (34)

1.12 Lenguajes de modelado y desarrollo

El lenguaje de modelado, muy utilizado desde hace varias décadas, es el lenguaje que se comunica gráficamente para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.

1.12.1 UML

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores. (35)

1.12.2 PHP

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser incrustado en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP. (36)

1.12.3 Javascript

El lenguaje JavaScript, trabajando en conjunto con las características relacionadas con el navegador, es una tecnología de mejora de la Web. Cuando se emplea en el equipo cliente, el lenguaje puede ayudar a convertir una página estática de contenido en una experiencia atractiva, interactiva e inteligente. Las solicitudes pueden ser tan sutil como la bienvenida de visitantes a un sitio con el saludo" ¡Buenos días!" cuando es por la mañana en la zona horaria del equipo cliente, incluso a pesar de que es hora de la cena, donde se encuentra el servidor. (37)

1.13 Conclusiones parciales

Luego del estudio bibliográfico realizado para el desarrollo de este capítulo, se concluye que las distintas técnicas de identificación y autenticación son esenciales para proteger la confidencialidad, disponibilidad e integridad de la información.

En el estudio de las soluciones a nivel mundial se refleja la tendencia actual de aumentar los niveles de seguridad en las aplicaciones pero con el uso de una herramienta de este tipo, se estaría incorporando un riesgo de seguridad inminente ya que no se tendría el dominio del funcionamiento interno del

sistema y en vez de disminuir las brechas de seguridad se estaría aumentando el nivel de incertidumbre y vulnerabilidad de nuestras aplicaciones. Por otra parte en las soluciones desarrolladas en la universidad se encuentran implementadas técnicas avanzadas de autenticación, pero no están integradas a un sistema de seguridad que gestione los distintos niveles de autenticación de las aplicaciones. Esto último constituye básicamente el objetivo principal de Acaxia.

La no disponibilidad de sistemas que brinden soluciones para aumentar los niveles de seguridad que se ajusten a las circunstancias del país hace necesario la implementación de una solución que se integre al SIGIS Acaxia logrando gestionar la autenticación con las distintas técnicas y aumentar los niveles de seguridad de las aplicaciones suscritas a este sistema.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

El presente capítulo está realizado con los resultados obtenidos durante el proceso de desarrollo de la solución, así como con algunos de los artefactos generados por el mismo. Se describen los procesos de negocio relativos al campo de acción y los requisitos funcionales de la solución para lograr un entendimiento de lo que se quiere lograr, se desarrolla el modelo de análisis y diseño, con los artefactos correspondientes a cada uno y se describen los artefactos de implementación de la solución en cuestión.

2.1 Descripción de la solución

La solución de los drivers debe permitir la autenticación de los usuarios en un entorno multi-entidad y multi-sistema, para ello se debe gestionar la autenticación de los sistemas y los usuarios. Estos últimos pueden estar registrados en Acaxia y autenticarse a su vez con servidores LDAP o requerir autenticación por código de barra. Mientras que los sistemas pueden autenticarse con servidores LDAP, por lo que la solución debe permitir además, que los usuarios no registrados en Acaxia que solicitan un sistema con autenticación LDAP, se autentiquen usando este tipo de tecnología.

A continuación la Figura 5 muestra un mapa conceptual que relaciona los principales conceptos identificados en la investigación, además se muestran los diagramas de procesos fundamentales con una breve descripción para que el equipo de desarrollo comprenda el negocio que debe cubrir la aplicación. Es necesario que se traten puntos clave del desarrollo como la arquitectura de la aplicación, patrones de diseño utilizados, algunos artefactos que propone la metodología seleccionada y el diseño de clases del sistema.

Para lograr la calidad del proceso de desarrollo será guiado por los estándares y normativas dictadas por el departamento de tecnología del CEIGE. Estas normativas regulan el diseño de interfaces de usuarios, la nomenclatura de las clases, componentes y objetos a nivel de datos, estilos de codificación, validaciones a todos los niveles y la forma en la que se deben describir cada uno de los componentes desarrollados para facilitar el mantenimiento y la reutilización.

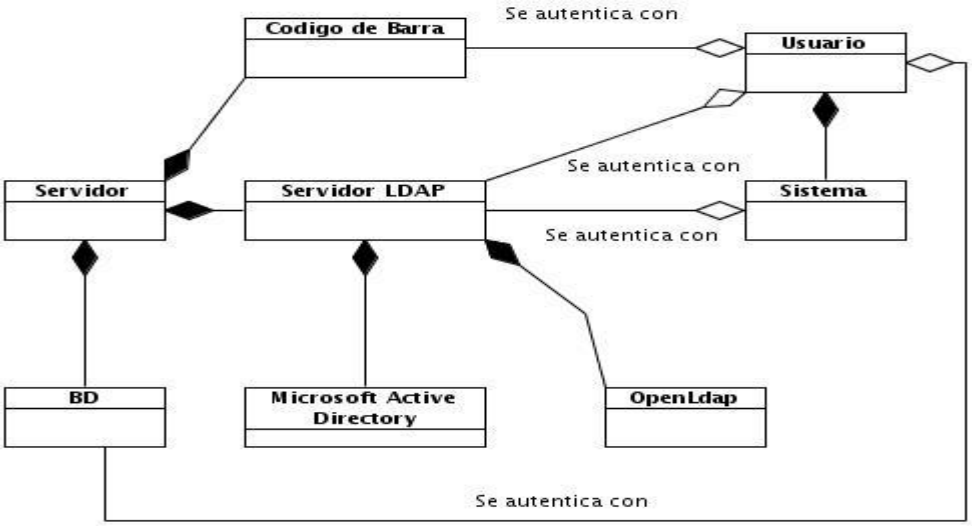


Figura 3 Modelo conceptual

Usuario: Cualquier persona que interactúa con el sistema Acaxia y solicita acceder a un sistema.
Sistema: Recurso o aplicación a la que Acaxia le brinda el servicio de seguridad.
Código de Barra: Concepto que permite la autenticación con código de barra.
Servidor: Es el concepto que agrupa la configuración de los distintos tipos de servidores.
Servidor LDAP: Es el concepto que representa la autenticación con servidores LDAP.
Microsoft Active Directory: Es el concepto que permite la autenticación con servidores Active Directory.
OpenLDAP: Es el concepto que permite la autenticación con servidores OpenLDAP.
BD: Es el concepto que permite la autenticación con servidores de bases de datos.

2.2 Descripción de los procesos de negocio

Un proceso de negocio es un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido. La descripción de los procesos de negocio hace más viable el paso a las actividades del análisis ya que posibilita una compresión más clara de los procesos en cuestión y contribuye a que los requisitos que se definan satisfagan las necesidades del usuario. (38)

El autor, teniendo en cuenta la opinión de especialistas y de los usuarios finales, ha identificado tres procesos de negocio que deben ser asistidos por la solución. Estos procesos son descritos a continuación.

Tabla 2 Descripción del proceso: Autenticar mediante código de barra

Objetivo	Lograr la autenticación de los usuarios a través de tarjetas de código de barra.
Evento(s) que lo genera(n)	Necesidad de acceder al sistema utilizando la identificación por código de barra.
Pre condiciones	Existencia de usuario(s) con este tipo de autenticación en Acaxia. El usuario solicita acceder a un recurso que gestiona Acaxia.
Marco legal	
Clientes internos	Usuarios de Cedrux.
Clientes externos	Entidades cubanas.
Entradas	Solicitar introducir usuario y contraseña: El sistema solicita al usuario que introduzca los datos de autenticación.
Flujo de eventos	
Flujo básico Exportar comprobantes de operaciones	
1. Autenticar por usuario y contraseña: El usuario introduce sus datos envía una petición de autenticación a Acaxia.	
2. Autenticación correcta? El sistema comprueba los datos del usuario y la autenticación tiene éxito.	
3. Solicitar introducir código de barra: El sistema solicita al usuario introducir el código de barra que lo identifica.	
4. Autenticar por código de barra: El usuario introduce su código y envía una petición de autenticación a Acaxia.	
5. Autenticación correcta: El sistema comprueba los datos del usuario y la autenticación tiene éxito.	
Pos-condiciones	
1. Se ha autenticado el usuario.	
Salidas	
1. Autorización: Se da inicio al proceso de autorización.	
Flujos paralelos	
1.	
2.	
Pos-condiciones	
1.	
Salidas	

1.
Flujos alternos
2. a Autenticación correcta.
2.1 Autenticación correcta: El sistema comprueba los datos del usuario y la autenticación no tiene éxito. Se regresa a la tarea: Solicitar introducir usuario y contraseña
Pos-condiciones
1. Se ha denegado el acceso al usuario.
Salidas
1. Solicitar introducir usuario y contraseña: El sistema muestra un mensaje de Acceso denegado y solicita introducir usuario y contraseña.
5. a Autenticación correcta: El sistema comprueba los datos del usuario y la autenticación no tiene éxito.
Pos-condiciones
2. Se ha denegado el acceso al usuario.
Salidas
2. Solicitar introducir código de barra: El sistema muestra un mensaje de Acceso denegado y regresa a la tarea: Solicitar introducir código de barra.

Asuntos pendientes
Posibles mejoras al proceso.

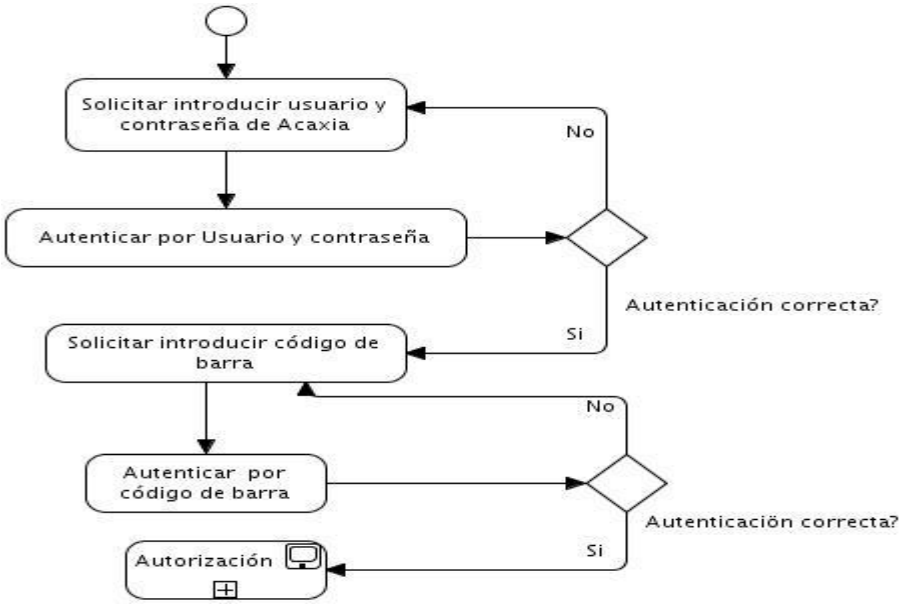


Figura 4 Proceso: Autenticar mediante código de barra

Tabla 3 Descripción del proceso: Autenticar usuario no registrado en Acaxia con LDAP

Objetivo	Lograr la autenticación de los usuarios no registrados en Acaxia con un servidor LDAP.
Evento(s) que lo genera(n)	Necesidad autenticar con servidores LDAP los usuarios que no están registrados en Acaxia y desean acceder a recursos.
Pre condiciones	Existencia de sistemas en Acaxia con este tipo de autenticación. Existencia de servidores LDAP registrados en Acaxia. El usuario desea acceder a un recurso gestionado por Acaxia.
Marco legal	
Clientes internos	Usuarios de Cedrux.
Clientes externos	
Entradas	Solicita usuario contraseña: El sistema solicita al usuario introducir los datos requeridos para la autenticación.
Flujo de eventos	
Flujo básico Gestionar servidores	
6. Autenticar usuario: El usuario introduce los datos y solicita autenticarse.	
7. Buscar servidor LDAP del sistema al que pertenece el usuario: El sistema Acaxia busca el servidor LDAP del sistema al que el usuario desea acceder.	
8. Autenticar usuario con LDAP: el sistema autentica el usuario con el servidor LDAP que éste tiene configurado en Acaxia.	
9. Autenticación correcta? La autenticación con servidor LDAP es realizada con éxito.	
10. Proceso de autorización: La autenticación finaliza con éxito y comienza el proceso de autorización.	
Pos-condiciones	
2. Se ha autenticado el usuario con el servidor LDAP.	
Salidas	
2. Proceso de autorización	
Flujos paralelos	
3.	
4.	
Pos-condiciones	
2.	
Salidas	
2.	
Flujos alternos	
3.a Autenticación correcta? La autenticación con el servidor LDAP no es realizada con éxito..	

3.1	Solicita usuario contraseña: El sistema muestra un mensaje de Acceso denegado y re regresa a esta tarea.
Pos-condiciones	
3.	
Salidas	
3. Solicita usuario contraseña: El sistema muestra un mensaje de Acceso denegado y re regresa a esta tarea.	
Asuntos pendientes	
Posibles mejoras al proceso.	

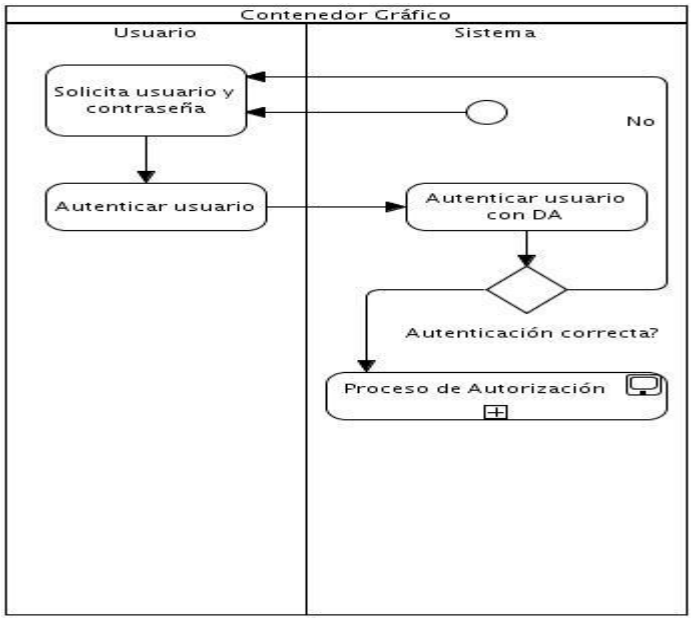


Figura 5 Proceso: Autenticar usuario no registrado en Acaxia con LDAP

Tabla 4 Descripción del proceso: Autenticar usuario registrado en Acaxia con LDAP

Objetivo	Lograr la autenticación de los usuarios registrados en Acaxia con un servidor LDAP.
Evento(s) que lo genera(n)	Necesidad autenticar con servidores LDAP.
Pre condiciones	Existencia de usuarios en Acaxia con este tipo de autenticación. Existencia de servidores LDAP registrados en Acaxia. El usuario desea acceder a un recurso gestionado por Acaxia.
Marco legal	

Clientes internos	Usuarios de Cedrux.
Clientes externos	
Entradas	Solicita usuario contraseña: El sistema solicita al usuario introducir los datos requeridos para la autenticación.
Flujo de eventos	
Flujo básico Gestionar servidores	
11. Autenticar usuario: El usuario introduce los datos y solicita autenticarse.	
12. Autenticar usuario con LDAP: el sistema autentica el usuario con el servidor LDAP que éste tiene configurado en Acaxia.	
13. Autenticación correcta? La autenticación con servidor LDAP es realizada con éxito.	
14. Proceso de autorización: La autenticación finaliza con éxito y comienza el proceso de autorización.	
Pos-condiciones	
3. Se ha autenticado el usuario con el servidor LDAP.	
Salidas	
3. Proceso de autorización	
Flujos paralelos	
5.	
6.	
Pos-condiciones	
3.	
Salidas	
3.	
Flujos alternos	
3.a Autenticación correcta? La autenticación con el servidor LDAP no es realizada con éxito..	
3.1 Solicita usuario contraseña: El sistema muestra un mensaje de Acceso denegado y re regresa a esta tarea.	
Pos-condiciones	
4.	
Salidas	
Solicita usuario contraseña: El sistema muestra un mensaje de Acceso denegado y re regresa a esta tarea.	
Asuntos pendientes	
Posibles mejoras al proceso.	

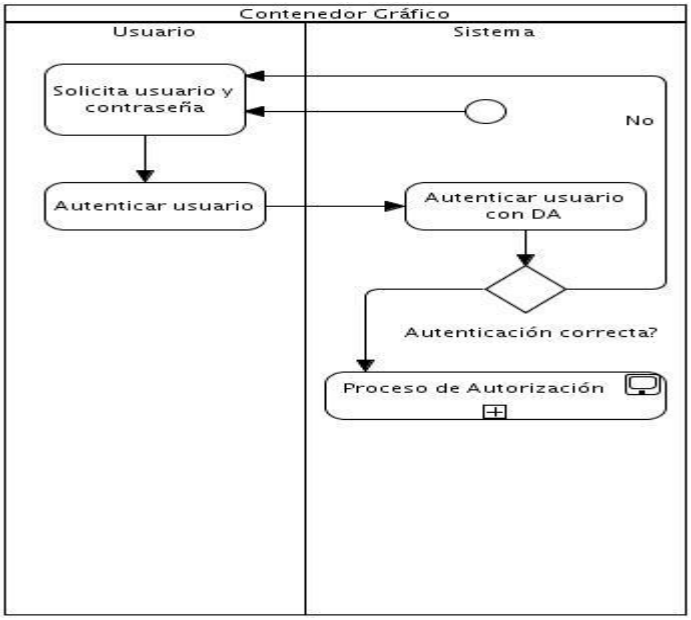


Figura 6 Proceso: Autenticar usuario registrado en Acaxia con LDAP

2.3 Requisitos de software

Los requisitos de software se clasifican en funcionales y no funcionales. Los requisitos funcionales describen qué es lo que el sistema debe hacer para dar soporte a las funciones y objetivos del usuario. Los requisitos no funcionales imponen restricciones de cómo los requisitos funcionales deben ser implementados. (39)

2.3.1 Requisitos funcionales

Los requisitos funcionales son los que definen las funciones que el sistema será capaz de realizar. Los mismos se centran en qué y no cómo se deben hacer esas funciones. Una vez descritos los procesos de negocios, sé identificaron los requisitos a cumplir por el sistema, los cuales son listados a continuación:

R 1. Gestionar autenticación.

R 1.1. Autenticar con LDAP.

R 1.2. Autenticar con código barra.

R 2. Requisito funcional Gestionar servidores.

R 1.1. Listar servidores.

R 1.2. Adicionar servidor.

R 1.3. Modificar servidor.

R 1.4. Eliminar servidor.

R 1.5. Probar conexión.

R 3. Requisito funcional Gestionar usuarios.

R 1.1. Listar usuario.

R 1.2. Adicionar usuario.

R 1.3. Modificar usuario.

R 1.4. Eliminar usuario.

R 1.5. Asignar rol.

R 1.6. Cambiar contraseña.

R 1.7. Buscar usuario.

R 4. Requisito funcional Gestionar sistema.

R 1.1. Listar sistema.

R 1.2. Adicionar sistema.

R 1.3. Modificar sistema.

R 1.4. Eliminar sistema.

R 1.5. Importar sistema.

R 1.6. Exportar sistema.

2.3.2 Especificación de requisitos

Tabla 5 Especificación del requisito funcional: Autenticar usuario por contraseña

Precondiciones	
Flujo de eventos	
Flujo básico	
1	Se muestra la interfaz de autenticación de Acaxia.
2	Se introduce el usuario y la contraseña.
3	El sistema valida (ver validación 1) los datos introducidos
4	El sistema busca el usuario en la base de datos de Acaxia.
5	Si el usuario existe busca el tipo de autenticación que tiene.
6	Si el tipo de autenticación es con un servidor LDAP se autentica el usuario con el servidor LDAP.
7	Si la autenticación es correcta se verifica que la contraseña en la base de datos en Acaxia y la contraseña introducida por el usuario son iguales.
8	El usuario se autentica satisfactoriamente, se crea el certificado y se re

	direcciona al recurso solicitado.
9	Concluye el requisito
Flujos alternativos	
	Flujo alternativo 2.a Información errónea.
1	El sistema informa los datos erróneos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 4 del flujo básico.
	Pos-condiciones
	N/A
	Flujo alternativo 2.bInformación incompleta.
1	El sistema informa los datos vacíos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 4 del flujo básico.
	Pos-condiciones
	N/A
	Flujo alternativo 4.aEl usuario no existe en Acaxia.
1	El sistema busca el tipo de autenticación que tiene el sistema al que el usuario intenta acceder.
2	Si el tipo de autenticación es con un servidor LDAP se autentica el usuario con el servidor LDAP.
3	Si la autenticación es correcta seguir al paso 8 del flujo básico
	Pos-condiciones
1	Se autentica el usuario.
	Flujo alternativo 4.a.2Si el tipo de autenticación no es con un servidor LDAP.
1	Se deniega el acceso del usuario y se retorna al paso 1 del flujo básico
	Pos-condiciones
1	Se deniega el acceso.
	Flujo alternativo 4.a.3Si la autenticación no es correcta
1	Se deniega el acceso del usuario y se retorna al paso 1 del flujo básico
	Pos-condiciones
1	Se deniega el acceso.
	Flujo alternativo 6.aSi la autenticación es con el servidor de Acaxia
1	Si el usuario requiere autenticación con código de barra se lanza el requisito Autenticar con código de barra
2	Concluye el requisito
	Pos-condiciones
1	Comienza el requisito Autenticar usuario por código de barra.
	Flujo alternativo 6.a.1Si el usuario no requiere autenticación con código de barra
1	Se autentica el usuario con el servidor de base datos de Acaxia.

	Si la autenticación es correcta seguir al paso 8 del flujo básico.
	Pos-condiciones
	N/A
	Flujo alternativo 6.a.1.2Si la autenticación no es correcta
1	Se deniega el acceso del usuario y se retorna al paso 1 del flujo básico
	Pos-condiciones
	Se deniega el acceso.
Validaciones	
1	Se validan los datos según lo establecido en el Modelo conceptual CSG-ERP-N-SEG-i2201.
Conceptos	Visible en la interfaz: Usuario, Contraseña
Requisitos especiales	N/A
Asuntos pendientes	N/A

Figura 7 Prototipo de interfaz de usuario del requisito funcional Autenticar usuario por contraseña

Tabla 6 Autenticar usuario por código de barra

Precondiciones	El proceso Gestionar autenticación ha concluido con éxito.
Flujo de eventos	
Flujo básico	
10	Se muestra la interfaz Autenticar con código de barra.

11	Se introduce el código de barra.
12	El sistema valida (ver validación 1) el dato introducido
13	Si los datos son correctos el sistema autentica al usuario con el código de barra.
14	Si el usuario se autentica satisfactoriamente se crea el certificado y se re direcciona al recurso solicitado.
15	Concluye el requisito
Flujos alternativos	
	Flujo alternativo 4.a Información errónea.
1	El sistema señala los datos erróneos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 4 del flujo básico.
	Pos-condiciones
	N/A
	Flujo alternativo 4.bInformación incompleta.
1	El sistema señala los datos vacíos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 4 del flujo básico.
	Pos-condiciones
	N/A
	Flujo alternativo 5.aLa autenticación no tiene éxito.
1	Se deniega el acceso y se retorna al paso 1 del flujo básico.
	Pos-condiciones
	Se deniega el acceso.
Validaciones	
2	Se validan los datos según lo establecido en el Modelo conceptual CSG-ERP-N-SEG-i2201.
Conceptos	**Código de Barra** Visible en la interfaz: Código de barra
Requisitos especiales	N/A
Asuntos pendientes	N/A

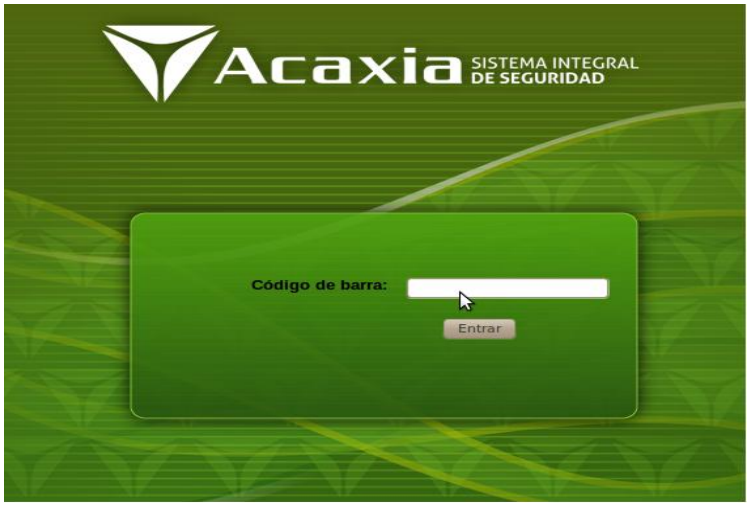


Figura 8 Prototipo de interfaz de usuario del requisito funcional Autenticar usuario por código de barra

La especificación de los demás requisitos se puede encontrar en la memoria general del proyecto Acaxia.

2.3.3 Requisitos no funcionales

Como se ha mencionado con anterioridad, el presente trabajo forma parte de un proceso productivo iniciado por el CEIGE y los resultados que se obtengan formarán parte del marco de trabajo desarrollado en el mismo. De esta manera los requisitos no funcionales a los que se debe acoger la aplicación a desarrollar son los que fueron establecidos por el centro al inicio del proceso de desarrollo, a continuación son descritos algunos de los más importantes.

Seguridad (SEG)

- Autenticación (Contraseña de acceso). Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.

Tabla 7 Requisitos de software y hardware

Requerimientos	Cliente	Servidor 1	Servidor 2

Software	• Navegador Mozilla Firefox V2.2 en adelante	• Sistema operativo Windows Advancer Server (2000 o superior) • Un servidor Apache 2.0 o superior con módulo PHP 5.0 disponible	• Linux distribución de Ubuntu v 8.8 en adelante • PostgreSQL 8.3
Hardware	• Procesador: 1.40 GHZ. • RAM: 256 MB (recomendado 512 Mb). • Tarjeta de Red: 1	• Procesador: 3.00 GHZ. • RAM: 1GB. • Disco duro: 160 GB. • UPS: 1. • Lector de CD: 1. • Tarjeta de Red: 1.	• Procesador: 3.00 GHZ. • RAM: 1GB. • Disco duro: 160 GB. • UPS: 1. • Lector de CD: 1. • Tarjeta de Red: 1.

2.4 Modelo de diseño

En el diseño se muestran las vistas más significativas para la arquitectura, compuestas por subsistemas de diseño encargados de definir y brindar las interfaces. En el modelo de diseño se muestran los diferentes componentes y las relaciones entre ellos. Se especifican las interfaces, clases y sus relaciones agrupadas por subsistemas y paquetes de diseño, así como un resumen de las responsabilidades de cada una de ellas.

Dentro de este epígrafe se tratan los elementos tenidos en cuenta durante el diseño de la solución, dando paso así a la exposición de los resultados de este flujo de trabajo, que refleja cómo será implementado el sistema en términos de clases del diseño.

2.4.1 Patrones Utilizados

Patrón arquitectónico Modelo-Vista-Controlador (MVC)

MVC es un patrón de arquitectura utilizado en sistemas Web para separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos, permitiendo flexibilidad y facilidad a la hora de hacer futuros cambios. (40)

La Vista es la información presentada al usuario. Una vista puede ser una página Web o una parte de una página. (40)

El Controlador actúa como intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para generar una página, es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. (40)

El Modelo representa las estructuras de datos. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos. (40)

Este patrón es empleado en el marco de trabajo Sauxe y determina la estructura de los paquetes internos de los componentes a desarrollar.

Patrones de diseño

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores y debe ser aplicable a diferentes problemas de diseño en distintas circunstancias. (41)

Para hacer un diseño eficiente se tomaron en cuenta un conjunto de patrones, que al ser experiencias de diseñadores expertos en orientación a objetos permiten dar soluciones eficientes a problemas existentes, facilitando notablemente el trabajo posterior. Algunos de estos patrones son:

- Fachada

Es un patrón estructural y consiste en crear una única clase de manejo más fácil, que permita acceder a un conjunto numeroso y complicado de clases. La fachada satisface a la mayoría de los clientes, sin ocultar las funciones de menor nivel a aquellos que necesiten acceder a ellas. (41)

En la solución se pone de manifiesto este patrón en las clases pertenecientes al paquete “services” del componente las cuales sirven de fachada a las funcionalidades brindadas por el mismo al resto de los componentes.

- Solitario

Es un patrón creacional que tiene como propósito garantizar una única instancia de una clase, proporcionando un punto de acceso global a la misma. (41)

En la solución se hacen varios usos de este patrón, un ejemplo de estos lo constituye el IoC (patrón inversión de control) que es utilizado por los componentes para integrarse entre sí.

2.4.2 Diagrama de Clases

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación, contiene información como clases, asociaciones, atributos, métodos y dependencias. A continuación se muestra el diagrama de clases del diseño basado en estereotipos web que se realizó durante el proceso de desarrollo.

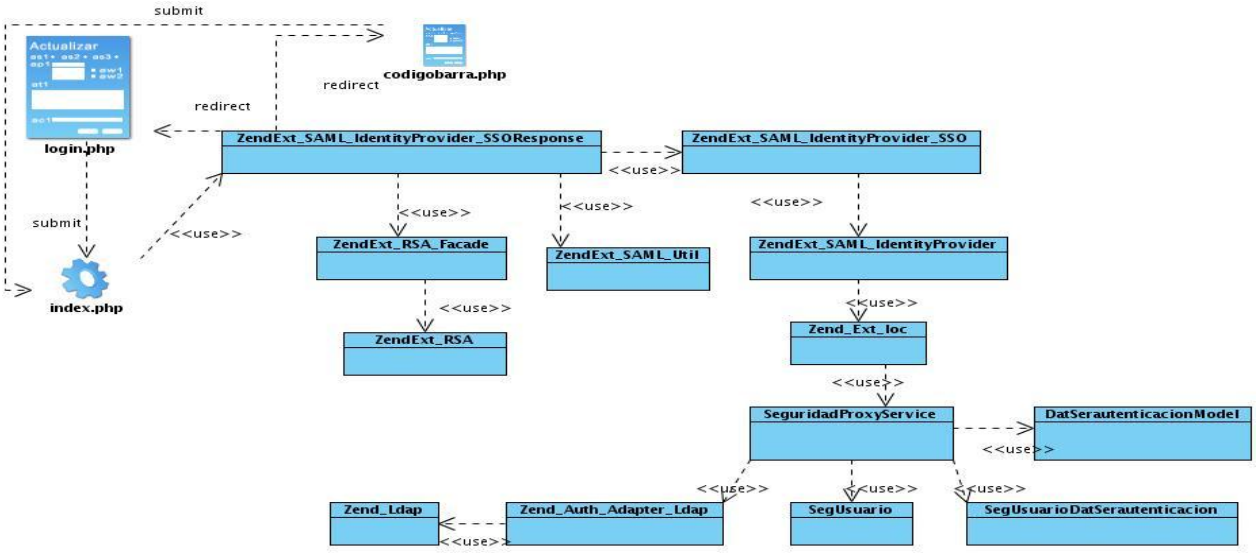


Figura 9 Diagrama de clases del diseño: Gestionar autenticación

2.4.3 Descripción de las clases del diseño asociadas al proceso: Gestionar autenticación

Tabla 8 Descripción de las clases del diseño asociadas al proceso: Gestionar autenticación

Clases	Descripción

ZendExt_SAML_IdentityProvider_SSOResponse	Clase interfaz del proveedor de identidad, en ella se inicializa la sesión, se crea un objeto de ZendExt_SAML_IdentityProvider_Server y es donde se gestiona la autenticación del usuario. En caso de que no exista el certificado re direcciona para login.php o código de barra respectivamente. Tiene como principales objetivos gestionar la autenticación, crear el certificado y redireccionar al usuario hacia el sistema solicitado.
login.php	Es el formulario de autenticación de nivel 1. (Usuario y contraseña).
codigobarra.php	Es el formulario de autenticación de nivel 2. (Código de barra).
index.php	Recibe lo enviado por login.php y codigobarra.php respectivamente y crea un objeto ZendExt_SAML_IdentityProvider_SSOResponse para que este se encargue de procesar los datos.
ZendExt_RSA_Facade	Clase utilizada para encriptar mediante el algoritmo RSA
ZendExt_Ioc	Integrador de servicios de negocio entre módulos
SeuridadProxieService	En general esta clase es una Interfaz o Proxy utilizado por el SGIS para brindar los servicios de autenticación, autorización, auditoria y administración de perfiles. En la solución es utilizada para autenticar el usuario con la base de datos de Acaxia o con servidores LDAP según corresponda. En esta clase se crea el

	certificado.
SegUsuario	Clase entidad donde se ejecutan las consultas de doctrine para obtener datos referentes a los usuarios.
DatSerautenticacion	Clase entidad donde se ejecutan las consultas de doctrine para obtener datos referentes a los servidores de autenticación LDAP.
SegUsuarioDatSerautenticacion	Clase entidad donde se ejecutan las consultas de doctrine para obtener datos referentes a los usuarios que utilizan servidores de autenticación LDAP.
Zend_Auth_Adapter_Ldap	Clase interfaz de Zend_Ldap
Zend_Ldap	Permite la autenticación de usuarios con cualquier servidor LDAP.

2.4.4 Modelo de datos

Un modelo de datos es una colección de conceptos bien definidos matemáticamente que ayudan a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo. (42)

El modelo de datos de la solución cuenta con un total de 6 tablas principales. Para su confección se tuvieron en cuenta los procesos de normalización y la utilización de nomencladores para facilitar su diseño. A continuación se muestra el diagrama entidad-relación que lo describe.

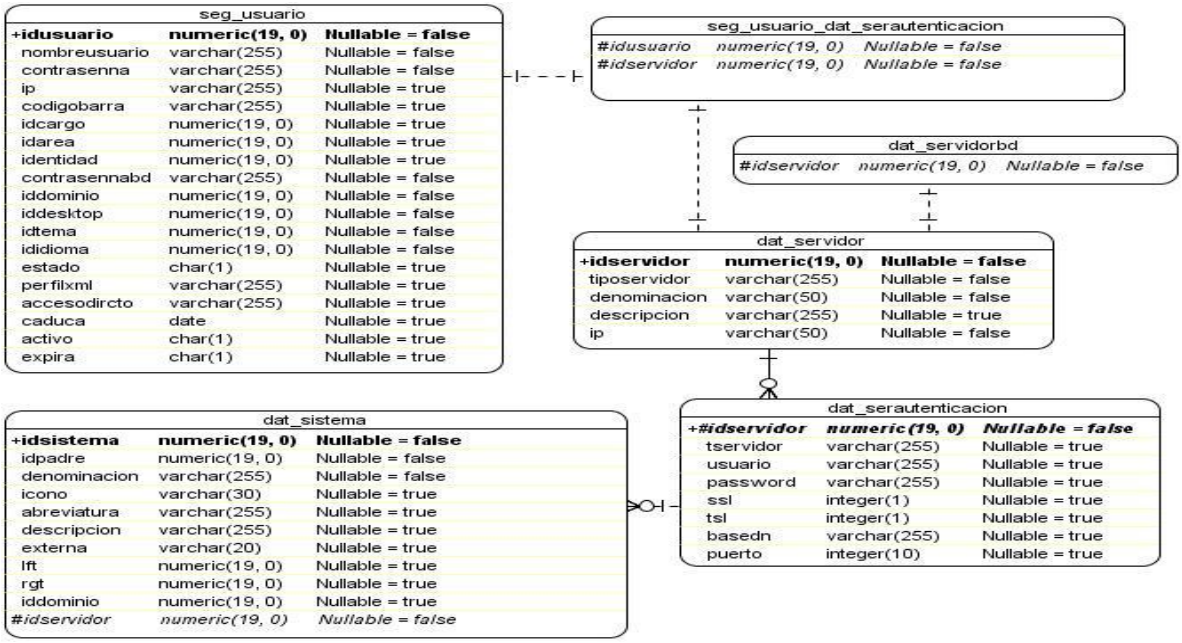


Figura 10 Modelo de datos

2.5 Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño, se implementan en términos de componentes. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros, además de los recursos necesarios para poder ejecutar el sistema desarrollado.

2.5.1 Diagrama de componentes

Los diagramas de componentes se utilizan para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

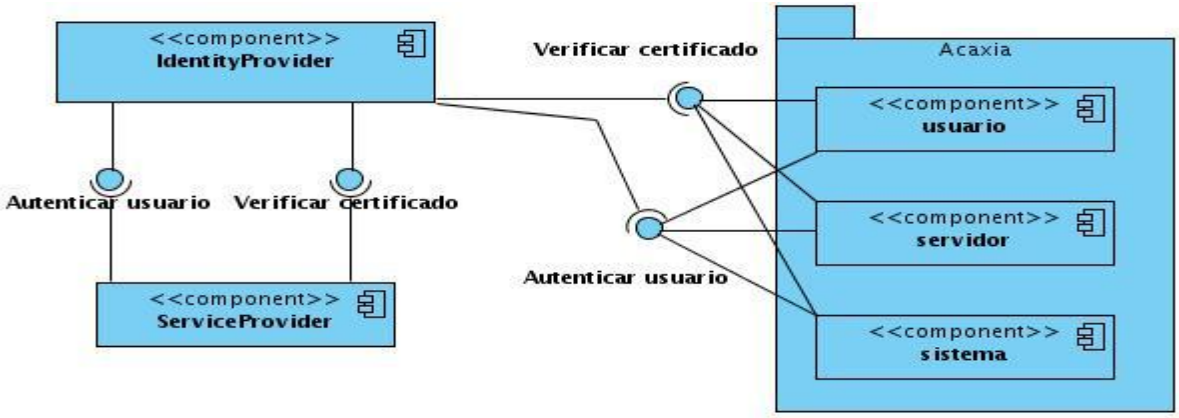


Figura 11 Diagrama de componentes

2.6 Conclusiones parciales

En el capítulo fueron mostrados los distintos artefactos generados durante el desarrollo de la solución. El equipo de desarrollo se comportó como usuario y desarrollador, los requisitos mostrados fueron identificándose en el transcurso de la implementación y puesta en práctica de las distintas ideas planteadas. El diagrama de las clases del diseño representa la integración de la solución con el estándar SAML; el diagrama de clases en la capa de datos representa las distintas clases y atributos necesarios. Se representa además el diagrama de componentes donde se muestra la organización y dependencias de los componentes involucrados en la solución. Con el desarrollo de la solución se adquirieron conocimientos sobre la seguridad informática y el desarrollo de aplicaciones web de gestión. Una vez concluido el diseño e implementación de la solución puede darse paso a la validación de la solución.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN

En el presente capítulo se muestran los casos de prueba y se analizan las métricas que se aplican en la actualidad y cuáles de ellas se aplicaron al diseño de los componentes que integran la solución, asegurando así la calidad del producto.

3.1 Métricas del software

El proceso del software y las métricas del producto son una medida cuantitativa que permite a la gente del software tener una visión profunda de la eficacia del proceso del software y de los proyectos que dirigen utilizando el proceso como un marco de trabajo. Se reúnen los datos básicos de calidad y productividad. Estos datos son entonces analizados, comparados con promedios anteriores, y evaluados para determinar las mejoras en la calidad y productividad. Las métricas son también utilizadas para señalar áreas con problemas de manera que se puedan desarrollar los remedios y mejorar el proceso del software. (38)

Dentro de los principales atributos de calidad que se incluyen en la aplicación de las métricas, se encuentran:

- **Responsabilidad**: se le asigna a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
- **Complejidad de implementación**: grado de dificultad en la implementación de un diseño de clases determinado.
- **Reutilización**: nivel de reutilización que tiene una clase o estructura de clase, dentro de un diseño de software determinado.
- **Acoplamiento**: valor de dependencia de una clase o estructura de clase con otras. Este atributo está muy ligado al de Reutilización.
- **Complejidad del mantenimiento**: categoría de esfuerzo para realizar un arreglo, mejora o rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costes y la planificación del proyecto.

- **Cantidad de pruebas**: número de esfuerzos para realizar las pruebas de calidad (Unidad) del producto (componente, clase, conjunto de clases, etc.) diseñado.

En un lenguaje orientado a objetos como es el PHP, las clases constituyen la unidad básica y fundamental. Por tanto debido a que la solución está realizada sobre este tipo de programación, su validación está centrada en la aplicación de métricas dirigidas a sus clases de forma individual, sus jerarquías y colaboraciones, haciendo uso de los atributos de calidad descritos anteriormente. Dichas métricas se encuentran desarrolladas a continuación:

Tamaño operacional de clase (TOC): Está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad:

Tabla 9 Atributos de calidad evaluados por la métrica TOC.

Atributo de calidad	Modo en que lo afecta
Responsabilidad	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento del TOC implica una disminución del grado de reutilización de la clase.

Para los cuales están definidos los siguientes criterios y categorías de evaluación:

Tabla 10 Criterios de evaluación para la métrica TOC.

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

Relaciones entre clases (RC): Está dado por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad:

Tabla 11 Atributos de calidad evaluados por la métrica RC.

Atributo de calidad	Modo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del Acoplamiento de la clase.
Complejidad de mantenimiento	Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
Reutilización	Un aumento del RC implica una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

Para los cuales están definidos los siguientes criterios y categorías de evaluación:

Tabla 12 Criterios de evaluación para la métrica RC.

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
Complejidad de mantenimiento	Baja	<=Promedio
	Media	Entre Promedio y 2*Promedio
	Alta	>2*Promedio
Reutilización	Baja	>2*Promedio
	Media	Entre Promedio y 2*Promedio
	Alta	<=Promedio
Cantidad de pruebas	Baja	<=Promedio
	Media	Entre Promedio y 2*Promedio
	Alta	>2*Promedio

3.1.1 Resultados obtenidos en la aplicación de la métrica TOC

Los resultados obtenidos luego de aplicar las métricas TOC arrojan que el diseño propuesto tiene una calidad aceptable, teniendo en cuenta que el 67 % de las clases poseen una cantidad de procedimientos menor o igual al promedio general de 11,3, esto conlleva a que las evaluaciones sean positivas en los atributos de calidad involucrados (responsabilidad, complejidad de implementación y reutilización). A continuación se muestran los resultados obtenidos.

Tabla 13 Instrumento de evaluación de la métrica TOC.

Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
SeguridadProxyService	17	Media	Media	Media
Zend_Auth_Adapter_Ldap	8	Baja	Baja	Alta
Zend_Ldap	24	Alta	Alta	Baja
ZendExt_SAML_IdentityProvider_SSOResponse	9	Baja	Baja	Alta
login.php	3	Baja	Baja	Alta
codigobarra.php	3	Baja	Baja	Alta
ZendExt_SAML_IdentityProvider_Sso	8	Baja	Baja	Alta
ZendExt_RSA	10	Baja	Baja	Alta
ZendExt_SAML_Util	17	Media	Media	Media
ZendExt_SAML_IdentityProvider_Server	8	Baja	Baja	Alta
SegUsuario	37	Alta	Alta	Baja
SegUsuarioDatSerautenticacion	3	Baja	Baja	Alta
DatSerautenticacion	4	Baja	Baja	Alta
ZendExt_Ioc	11	Media	Media	Media
ZendExt_RSA_Facade	8	Baja	Baja	Alta
	11,33333333			

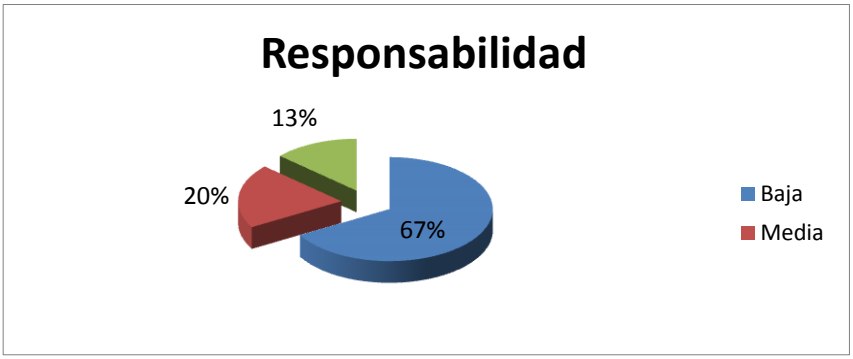


Figura 12 Resultados de la evaluación de la métrica TOC para el atributo Responsabilidad.

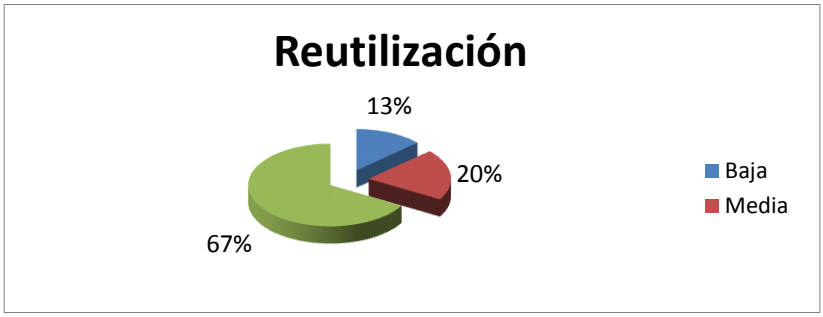


Figura 13 Resultados de la evaluación de la métrica TOC para el atributo Reutilización.

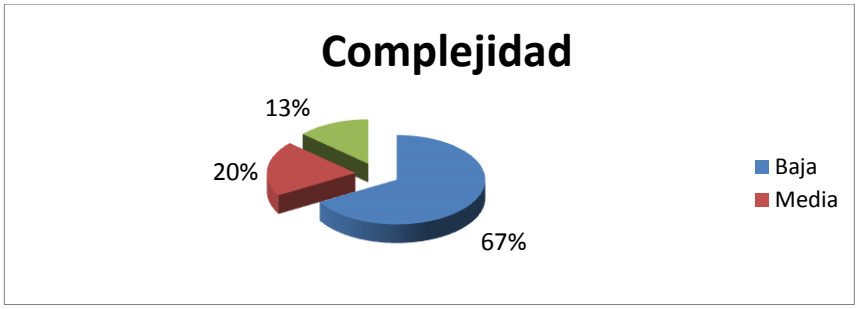


Figura 14 Resultados de la evaluación de la métrica TOC para el atributo Complejidad.

3.1.2 Resultados obtenidos de la aplicación de la métrica RC

Los resultados obtenidos luego de aplicar las métricas RC arrojan que el diseño propuesto tiene una calidad aceptable, teniendo en cuenta que el 67 % de las clases poseen una cantidad de relaciones de uso menor o igual al promedio general de 1,4, esto conlleva a que las evaluaciones sean positivas en los atributos de calidad involucrados (acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización). A continuación se muestran los resultados obtenidos.

Tabla 14 Instrumento de evaluación de la métrica RC.

Clase	Cantidad de Relaciones de Uso	Acoplamiento	Complejidad Mant.	Reutilización	Cantidad de Pruebas
SeguridadProxyService	3	Alto	Alto	Bajo	Alto
Zend_Auth_Adapter_Ldap	1	Bajo	Bajo	Alto	Bajo
Zend_Ldap	1	Bajo	Bajo	Alto	Bajo
ZendExt_SAML_IdentityProvider_SSOResponse	1	Bajo	Bajo	Alto	Bajo

login.php	1	Bajo	Bajo	Alto	Bajo
codigobarra.php	1	Bajo	Bajo	Alto	Bajo
ZendExt_SAML_IdentityProvider_Sso	1	Bajo	Bajo	Alto	Bajo
ZendExt_RSA	2	Medio	Medio	Medio	Medio
ZendExt_SAML_Util	1	Bajo	Bajo	Alta	Bajo
ZendExt_SAML_IdentityProvider_Server	1	Bajo	Bajo	Alta	Bajo
SegUsuario	2	Medio	Medio	Medio	Medio
SegUsuarioDatSerautenticacion	1	Bajo	Bajo	Alta	Bajo
DatSerautenticacion	2	Medio	Medio	Medio	Medio
ZendExt_Ioc	1	Bajo	Bajo	Alta	Bajo
ZendExt_RSA_Facade	2	Medio	Medio	Medio	Medio

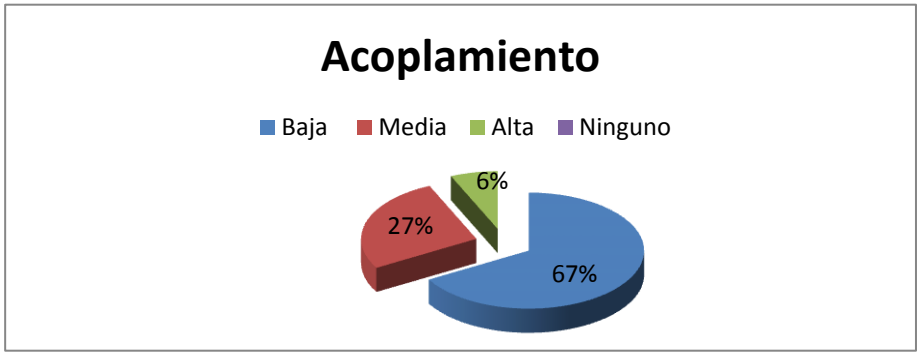


Figura 15 Resultados de la evaluación de la métrica RC para el atributo Acoplamiento.

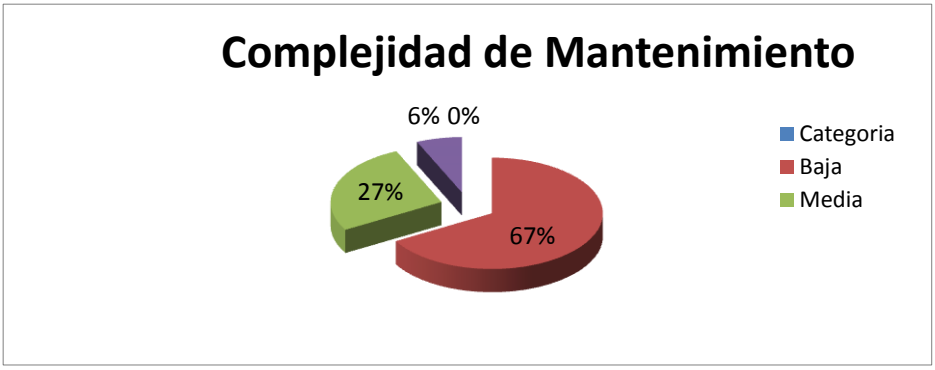


Figura 16 Resultados de la evaluación de la métrica RC para el atributo Complejidad de mantenimiento.

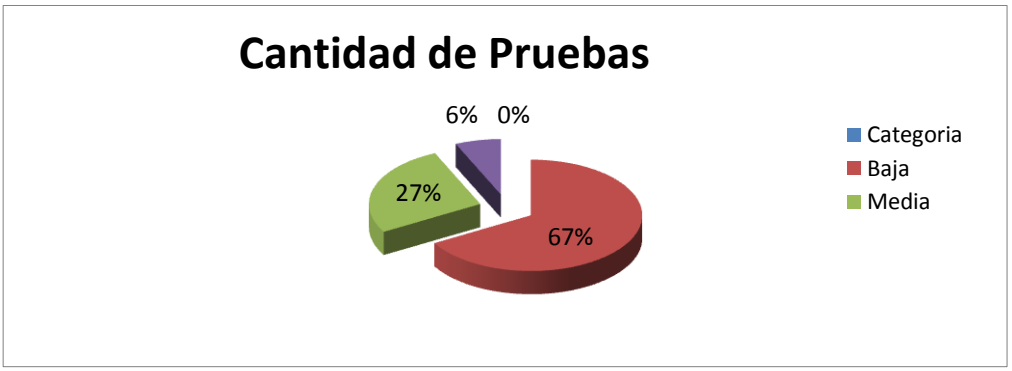


Figura 17 Resultados de la evaluación de la métrica RC para el atributo Cantidad de pruebas.

Figura 18 Resultados de la evaluación de la métrica RC para el atributo Reutilización.

3.1.3 Matriz de inferencia de indicadores de calidad

La matriz inferencia de indicadores de calidad, también llamada matriz de cubrimiento, es una representación estructurada de los atributos de calidad y métricas utilizadas para evaluar la calidad del diseño propuesto. Dicha matriz permite conocer si los resultados obtenidos de las relaciones atributo/métrica es positivo o no, llevando estos resultados a una escalabilidad numérica donde, si los resultados son positivos se le asigna el valor de 1, si son negativos toma valor 0 y si no existe relación es considerada como nula y es representada con un guión simple (-). Luego se puede obtener un resultado general para cada atributo promediando todas sus relaciones no nulas.

A continuación se muestran los resultados obtenidos.

Tabla 15 Resultados de la evaluación de la relación atributo/métrica.

Atributos\Métricas	TOC	RC	Promedio
Responsabilidad	1	(-)	1
Complejidad de Implementación	1	(-)	1
Reutilización	1	1	1
Acoplamiento	(-)	1	1
Complejidad de Mantenimiento	(-)	1	1
Cantidad de pruebas	(-)	1	1

Tabla 16 Rango de valores para la evaluación de la relación atributo/métrica

Categoría	Rango de valores
Malo	<=0.4
Regular	>0.4 y <0.7
Bueno	>=0.7

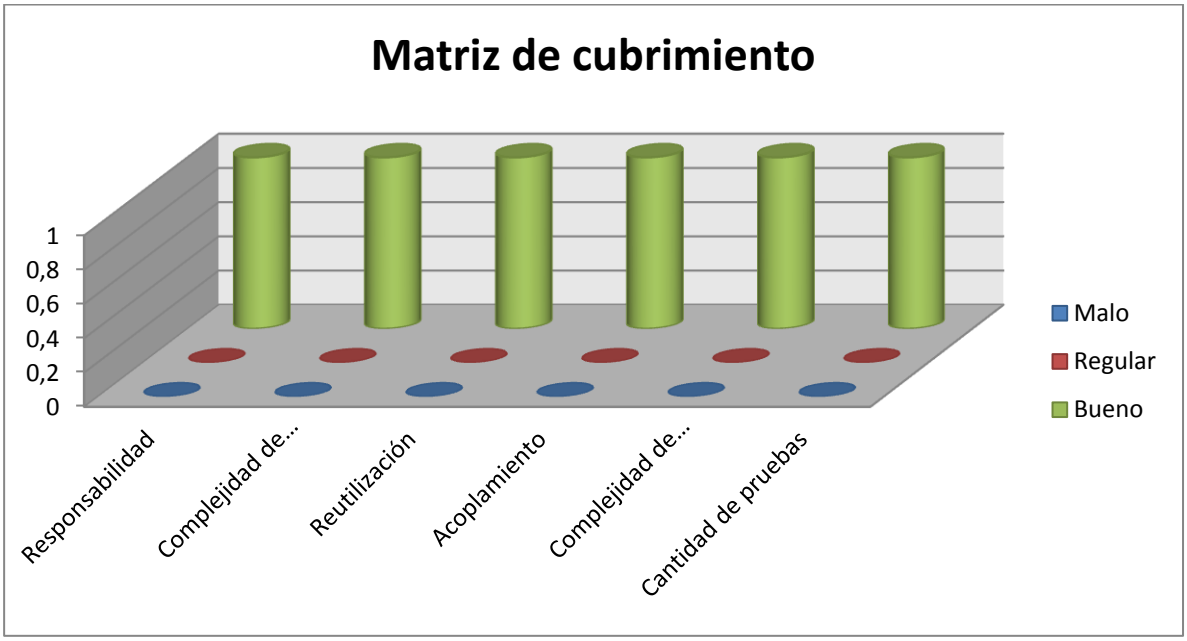


Figura 19 Resultados obtenidos de la evaluación de los atributos de calidad.

3.2 Pruebas de software

El desarrollo de sistemas de software implica una serie de actividades de producción en las que las posibilidades de que aparezca el fallo humano son enormes. Los errores pueden empezar a darse desde el primer momento del proceso, en el que los objetivos pueden estar especificados de forma errónea o imperfecta, así como en posteriores pasos de diseño y desarrollo. Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo de software ha de ir acompañado de una actividad que garantice la calidad. (38)

Las pruebas son las actividades que garantizan la calidad y el correcto funcionamiento de un software. A continuación se reflejan algunas características que se asumen para las pruebas a realizar:

- El objetivo principal de las pruebas será validar la corrección de cualquier artefacto que se vaya a probar. Las pruebas realmente exitosas son aquellas que encuentran defectos.
- Es posible probar todos los artefactos, no solamente el código fuente. Como mínimo se deben revisar los modelos y documentos para de esta forma encontrar y corregir los defectos antes que lleguen al código.

- En la medida en que pueda ser más riesgoso algún artefacto o sistema en general, será más necesario que este sea revisado y probado.

- Es posible creer que la aplicación funciona correctamente, pero solo cuando se muestren los resultados de las pruebas se podrá estar seguro de la veracidad de este hecho.

3.2.1 Pruebas de cajas blancas

La prueba de caja blanca, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que:

1- Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo.

2- Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.

1- Ejecuten todos los bucles en sus límites y con sus límites operacionales;

2- Ejerciten las estructuras internas de datos para asegurar su validez. (38)

Pruebas de camino básico

La prueba del camino básico es una técnica de prueba de caja blanca propuesta inicialmente por Tom McCabe. El método del camino básico permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. (38)

El empleo de este tipo de pruebas permitirá diseñar casos de prueba que comprueben que todas las sentencias del sistema se ejecuten al menos una vez, además de todas las condiciones, tanto verdaderas como falsas. Para esto primero se procede a enumerar las sentencias del código y a partir del mismo se construye el grafo de flujo asociado.

```
private function AutenticarusuarioacaxiaconLDAP($user,$password,$usuario_acaxia,$mac,$idservidor){
    $SA = DatServidor::usuarioservidoraut($idservidor); (1)
    $RSA = new ZendExt_RSA_Facade();      (1)
      if($this->AD_Autentication($SA, $user, $password)){    (2)
    if ($RSA->decrypt($usuario_acaxia[0]->contrasenna) != $password) {  (3)
                        $objusurio = Doctrine::getTable('SegUsuario')->find($usuario_acaxia[0]->idusuario); (4)
                        $objusurio->contrasenna = $RSA->encrypt($password);  (4)
                        $objusurio->save();     (4)
                        $result = $this->returnCertificate($mac, $usuario_acaxia[0]->idusuario); (4)
                        return $result;   (4)
                    } else {                (5)
                        $result = $this->returnCertificate($mac, $usuario_acaxia[0]->idusuario); (6)
                        return $result;  (6)
                    }
      }
      return 0; (7)
}
```

Figura 20 Código fuente de la funcionalidad Autenticar usuario de Acaxia con LDAP.

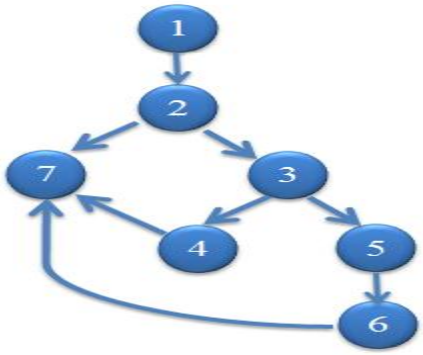


Figura 21 Grafo de flujo asociado a la funcionalidad: Autenticar usuario de Acaxia con LDAP.

Luego de haber construido el grafo se realiza el cálculo de la complejidad ciclomática mediante las tres fórmulas descritas a continuación, las cuales deben arrojar el mismo resultado para asegurar que el cálculo de la complejidad es correcto.

1. **V (G) = (A – N) + 2**

Siendo “A” la cantidad total de aristas y “N” la cantidad total de nodos.

V (G) = (8 - 7) + 2

V (G) = 3

2. **V (G) = P + 1**

Siendo “P” la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).

V (G) = 2 + 1

V (G) = 3

3. V (G) = R

Siendo “R” la cantidad total de regiones, para cada fórmula “V (G)” representa el valor del cálculo.
V (G) = 3.

Los cálculos efectuados mediante las tres fórmulas anteriores dan el mismo resultado V(G) = 3, lo que revela que existen tres posibles caminos por donde el flujo puede circular y determina el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez. A continuación se representan los caminos básicos por los que puede recorrer el flujo:

- Camino básico #1:1-2-3-4
- Camino básico #2: 1-2-3-5-6
- Camino básico #3:1-2-7

Para realizar los casos de prueba de cada camino básico es preciso cumplir con las siguientes exigencias:

- Descripción: se hace la entrada de datos necesaria, validando que ningún parámetro obligatorio pase nulo al procedimiento o no se entre algún dato erróneo.
- Condición de ejecución: se especifica cada parámetro para que cumpla una condición deseada para ver el funcionamiento del procedimiento.
- Resultados Esperados: se expone resultado que se espera que devuelva el procedimiento.
- Evaluación de los resultados: se exhibe la evaluación que dio el resultado final del procedimiento.

Para cada camino se realiza un caso de prueba:

- **Caso de prueba para el Camino básico #1:**

- **Descripción**: Se quiere determinar si el método devuelve el certificado de autenticación cuando todos los parámetros son correctos. Los datos de entrada cumplirán con los siguientes

requisitos: el atributo user y usuario_acaxia deben ser cadenas que comienzan con letras y estar registrados en la base de datos de Acaxia, los atributos mac e idservidor son numéricos y están igualmente registrados en Acaxia.

- **Condición de ejecución**. . El atributo password debe ser igual a la contraseña del usuario en el servidor LDAP y distinto en la base de datos de Acaxia.

- **Resultados esperados**: al ejecutar el procedimiento se actualizará el password en la base de datos de Acaxia y se retornará el certificado del usuario.

- **Evaluación de los resultados obtenidos**: los resultados de la realización del caso de prueba de caja blanca para el caso de autenticar un usuario de Acaxia con un servidor LDAP, fueron satisfactorios al lograr que se actualizara la contraseña del usuario en la base de datos de Acaxia y se retornara el certificado del usuario.

➢ **Caso de prueba para el Camino básico #2:**

- **Descripción:** Se quiere determinar si el método devuelve el certificado de autenticación cuando todos los parámetros son correctos. Los datos de entrada cumplirán con los siguientes requisitos: el atributo user y usuario_acaxia deben ser cadenas que comienzan con letras y estar registrados en la base de datos de Acaxia, los atributos mac e idservidor son numéricos y están igualmente registrados en Acaxia.

- **Condición de ejecución**. . El atributo password debe ser igual a la contraseña del usuario en el servidor LDAP y en la base de datos de Acaxia.

- **Resultados esperados**: al ejecutar el procedimiento se retornará el certificado del usuario.

- **Evaluación de los resultados obtenidos**: los resultados de la realización del caso de prueba de caja blanca para el caso de autenticar un usuario de Acaxia con un servidor LDAP, fueron satisfactorios al lograr que se retornara el certificado del usuario.

➢ **Caso de prueba para el Camino básico #3:**

- **Descripción:** Se quiere determinar si el método devuelve el certificado de autenticación cuando algunos de sus parámetros son incorrectos. Los datos de entrada no cumplirán con los siguientes requisitos: el atributo user y usuario_acaxia deben ser cadenas que comienzan con

letras y estar registrados en la base de datos de Acaxia, los atributos mac e idservidor son numéricos y este último está registrado en Acaxia.

- **Condición de ejecución**. El atributo password debe ser distinto a la contraseña del usuario en el servidor LDAP o el atributo idservidor no se encuentra registrado en Acaxia, o el atributo user no está se encuentra en el servidor LDAP.

- **Resultados esperados**: al ejecutar el procedimiento se retornará 0 (el certificado = 0 representa que éste no ha sido creado).

- **Evaluación de los resultados obtenidos**: los resultados de la realización del caso de prueba de caja blanca para el caso de autenticar un usuario de Acaxia con un servidor LDAP, fueron satisfactorios al lograr que se retornara 0cuando al menos uno de los atributos recibidos por parámetros es incorrecto.

3.3 Pruebas de caja negra

La validación del software se consigue mediante una serie de pruebas de caja negra que demuestran la conformidad con los requisitos. Un plan de prueba traza la clase de pruebas que se han de llevar a cabo, y un procedimiento de prueba define los casos de prueba específicos en un intento por descubrir errores de acuerdo con los requisitos. Tanto el plan como el procedimiento estarán diseñados para asegurar que se satisfacen todos los requisitos funcionales, que se alcanzan todos los requisitos de rendimiento, que la documentación es correcta e inteligible y que se alcanzan otros requisitos (por ejemplo, portabilidad, compatibilidad, recuperación de errores, facilidad de mantenimiento) (38)

Las pruebas realizadas por el usuario constituyen pruebas de caja negra que se centran en el cumplimiento de los requisitos definidos para el sistema una vez concluido. Por lo tanto, estas pruebas constituyen la validación de la aplicación por parte del usuario final.

Algunas técnicas utilizadas en la prueba de caja negra son:

- **Partición de equivalencia:** Técnica que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a una definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

- **Análisis de valores límite:** La experiencia muestra que los casos de prueba que exploran las condiciones límite producen mejor resultado que aquellos que no lo hacen. Las condiciones límite son aquellas que se hallan en los márgenes de la clase de equivalencia, tanto de entrada como de salida. Por ello, se ha desarrollado el análisis de valores límite como técnica de prueba. Esta técnica lleva a elegir los casos de prueba que ejerciten los valores límite.

- **Grafos de causa-efecto:** Es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Para la aplicación de estas técnicas se diseñaron un conjunto de escenarios de pruebas con el objetivo de evaluar el cumplimiento de los requisitos del sistema. A continuación se muestran los escenarios empleados.

Tabla 17 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
Gestionar autenticación	Se realizará la autenticación de un usuario en el sistema Acaxia	EP 1.1 Petición de autenticación de un usuario que está en el servidor Acaxia y utiliza autenticación LDAP.	– El cliente hace una petición del recurso. – Se muestra la ventana de autenticación. – El usuario introduce sus credenciales de autenticación (usuario y contraseña). – El sistema valida los datos introducidos – El sistema autentica el usuario con el servidor LDAP. – El usuario se autentica satisfactoriamente, se crea el certificado y se re direcciona al recurso solicitado.
		EP 1.2 Petición de autenticación de un usuario que no está	– El cliente hace una petición del recurso. – Se muestra la ventana de

		en el servidor Acaxia y utiliza autenticación LDAP.	autenticación. – El usuario introduce sus credenciales de autenticación (usuario y contraseña). – El sistema valida los datos introducidos – El sistema autentica el usuario con el servidor LDAP. – El usuario se autentica satisfactoriamente, se crea el certificado y se re direcciona al sistema solicitado.
		EP 1.3 Petición de autenticación de un usuario que está en el servidor Acaxia y utiliza autenticación con código de barra.	– El cliente hace una petición del recurso. – Se muestra la ventana de autenticación. – El usuario introduce sus credenciales de autenticación (usuario y contraseña). – El sistema valida los datos introducidos. – El sistema autentica el usuario con el servidor de Acaxia. – El usuario se autentica satisfactoriamente, comienza el EP 1.4
		EP 1.4 Petición de autenticación con código de barra.	– Se muestra la ventana de autenticación por código de barra. – El usuario introduce su código de barra. – El sistema valida los datos introducidos – El sistema autentica el

			usuario en la base de datos de Acaxia. – Se crea el certificado y se re direcciona al recurso solicitado.
		EP 1.5 Petición de autenticación introduciendo datos incorrectos en la ventana del login por usuario y contraseña.	– El usuario introduce los datos – El sistema muestra un mensaje informando los datos erróneos
		EP 1.6 Petición de autenticación introduciendo datos inválidos en la ventana del login por código de barra	– El usuario introduce el código de barra incorrecto – El sistema muestra un mensaje informando el dato erróneo

Tabla 18 Juego de datos a probar

No	Nombre de campo	Tipo	Válido	Inválido	Descripción
1	*usuario*	text	Letras y números (sólo después de las letras)	-	Nombre de usuario
2	*contraseña*	text	Letras, números y caracteres especiales.	-	Contraseña de usuario
3	Código de barra	text	Cadena alfanumérica		Código de barra de usuario

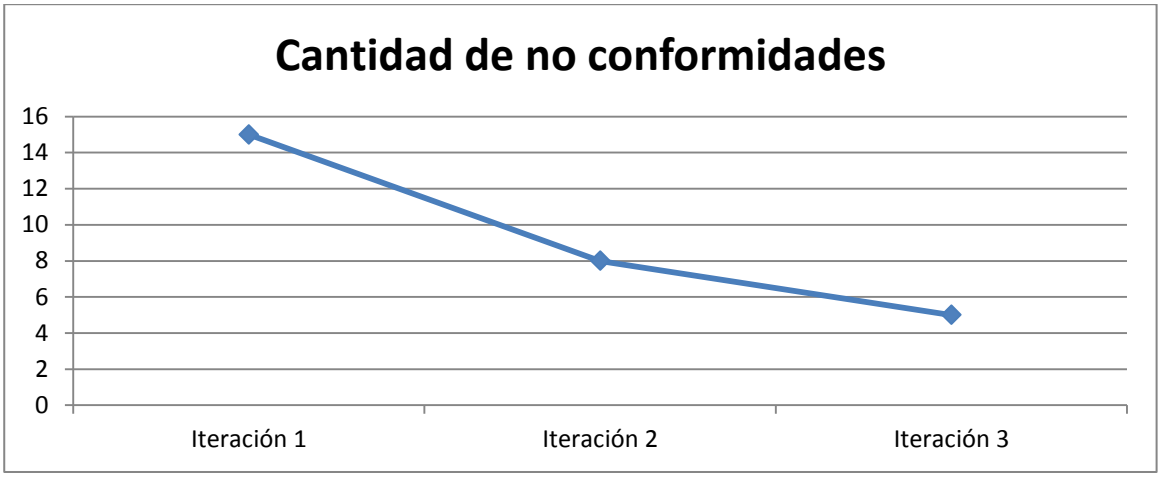
Tabla 19 Descripción de variables

Id del Escenario	Escenario	Variable1 (usuario)	Variable2 (contraseña)	Variable3 (código de barra)	Respuesta del Sistema	Resultado de la prueba
EP 1.1	Petición de autenticación de un usuario que está en el servidor Acaxia y utiliza autenticación LDAP.	V(rpelaez)	V(Contraseña UCI)	-	El usuario se autentica satisfactoriamente, se crea el certificado y se re direcciona al sistema solicitado.(Portal de Cedrux)	–
EP 1.2	Petición de autenticación de un usuario que no está en el servidor Acaxia y utiliza autenticación LDAP.	V(usuario uci)	V(Contrseña UCI)		El usuario se autentica satisfactoriamente, se crea el certificado y se re direcciona al sistema solicitado.	–
EP 1.3	Petición de autenticación de un usuario que está en el servidor Acaxia y utiliza autenticación con código de barra.	V(codigobarra)	V(codigobarra1”)	-	El usuario se autentica satisfactoriamente, comienza el EP 1.4	–
EP 1.4	Petición de autenticación con código de barra.	-	-	V(123)	Se crea el certificado y se re direcciona al recurso solicitado.	–

EP 1.5	Petición de autenticación introduciendo datos incorrectos en la ventana del login por usuario y contraseña.	I(campo vacio)	I(campo vacio)		El sistema muestra el mensaje: Usuario y contraseña incorrectos	–
		V(rpelaez)	I(contraseña1")		El sistema muestra el mensaje: Usuario y contraseña incorrectos.	–
EP 1.6	Petición de autenticación introduciendo datos inválidos en la ventana del login por código de barra			I(null)	El sistema muestra el mensaje: Código de barra incorrecto.	–
				I(asdhu12)	El sistema muestra el mensaje: Código de barra incorrecto.	–

En las pruebas realizadas al sistema y al manual se detectaron un total de 28 no conformidades. La Figura 31 muestra una gráfica donde se refleja la cantidad de no conformidades detectadas por iteración. El sistema fue liberado en la tercera iteración y el manual en la quinta iteración. Finalmente CALISOFT emitió su certificado de liberación que se puede encontrar en la memoria colectiva del proyecto Acaxia (43).

Figura 22 Cantidad de no conformidades



3.4 Conclusiones parciales

En este capítulo fueron aplicadas las métricas para valorar la calidad de la solución. Los resultados de aplicar la métrica Tamaño Operacional de Clase arrojaron que la mayoría de las clases tienen la complejidad de implementación y una responsabilidad, predominantemente baja y que a su vez tienen una alta reutilización. Por su parte los resultados de la métrica de Relaciones entre Clases demuestran que la implementación del sistema tiene una calidad aceptable, con niveles bajos de acoplamiento, de cantidad de pruebas de unidad y de la complejidad para el mantenimiento, por lo que se demuestra el uso de un buen diseño de software. La matriz de cubrimiento o matriz de inferencia de los atributos de calidad permitió valorar el impacto que tuvieron los atributos que se midieron en las métricas aplicadas en el diseño de la solución propuesta. Por lo que se puede concluir que el diseño propuesto no presenta complejidad, es fácil de implementar y dar mantenimiento, tiene una alta cohesión y es fácil de probar.

Para verificar el buen funcionamiento y eficiencia del código, se realizaron las pruebas del camino básico como parte de las pruebas de caja blanca, y para demostrar el cumplimiento de los requisitos definidos, se aplicaron pruebas de caja negra que fueron realizadas por el usuario.

Todo esto permitió verificar que las funcionalidades implementadas responden a las necesidades y propósitos de los clientes. La validación principal de un sistema es la aceptación de los usuarios, los avales que se exponen en los anexos de la memoria colectiva dan fe de ello. Se concluye por tanto que el sistema cumple con los requisitos definidos.

CONCLUSIONES

Al realizar un estudio del estado del arte de los niveles de autenticación de las aplicaciones implementadas tanto dentro como fuera del país, se obtuvo como resultado que la mayoría presentan soluciones particulares para escenarios muy específicos. Los sistemas extranjeros demuestran la tendencia actual de aumentar los niveles de seguridad de los sistemas de autenticación. Estas soluciones son privadas y para el país constituye una tecnología imposible de incorporar por su elevado costo.

Luego del análisis de las soluciones existentes se procedió al desarrollo de una solución para integrarla al SIGIS Acaxia y así gestionar la autenticación con las distintas técnicas aumentando los niveles de seguridad de las aplicaciones suscritas a este sistema.

El desarrollo del mismo se ejecutó sobre las herramientas y tecnologías libres, guiado por los modelos y estándares más utilizados para componentes de este tipo.

La solución fue sometida a un proceso de pruebas de funcionamiento guiado por casos de pruebas. Los resultados de aplicar la métrica Tamaño Operacional de Clase arrojaron que la mayoría de las clases tienen la complejidad de implementación y una responsabilidad, predominantemente baja y que a su vez tienen una alta reutilización. Por su parte los resultados de la métrica de Relaciones entre Clases demuestran que la implementación del sistema tiene una calidad aceptable, con niveles bajos de acoplamiento, de cantidad de pruebas de unidad y de la complejidad para el mantenimiento, por lo que se demuestra el uso de un buen diseño de software.

RECOMENDACIONES

El autor que realiza el presente trabajo considera importante distinguir las siguientes recomendaciones:

- Que los sistemas web de gestión desarrollados en la UCI y en general en Cuba se suscriban a la nueva versión de ACAXIA, ya que esta brindará mayor seguridad y nuevas funcionalidades.

- Trabajar en nuevas versiones del sistema para fortalecer la gestión de la seguridad brindada por este.

- Que la nueva versión sea avalada por las instituciones pertinentes como la ONRI y por instituciones que evalúen la calidad en cuanto a seguridad respecta como Segurmática.

Bibliografía

1. **Gómez Velázquez, Karel.** *PROPUESTA DE PROCESOS DE AUTENTICACIÓN, AUTORIZACIÓN Y AUDITORÍA (AAA) PARA APLICACIONES BASADAS EN SERVICIOS WEB XML.* Ciudad de la Habana : s.n., 2010.
2. WordNet Search -3.0. [En línea] [Citado el: 07 de 03 de 2011.] http://wordnetweb.princeton.edu/perl/webwn?s=driver.
3. **Villalon Huerta, Antonio.** RedIRIS. *RedIRIS.* [En línea] 15 de 07 de 2002. [Citado el: 18 de 02 de 2011.] http://www.rediris.es/cert/doc/unixsec/node14.html.
4. EcuRed. [En línea] 2002. [Citado el: 01 de 03 de 2011.] http://www.ecured.cu/index.php/Control_de_acceso.
5. The Free Dictionary. *The Free Dictionary.* [En línea] [Citado el: 3 de 4 de 2011.] http://es.thefreedictionary.com/identificaci%C3%B3n. 1.
6. msdn. *msdn.* [En línea] [Citado el: 4 de 3 de 2011.] http://msdn.microsoft.com/es-es/library/syf5yeat.aspx.
7. Instalación y configuración de un servidor RADIUS. *Instalación y configuración de un servidor RADIUS.* [En línea] [Citado el: 20 de 05 de 2011.] http://www.grc.upv.es/docencia/tra/PDF/Radius.pdf.
8. Cayu- Wiki de Sergio Cayuqueo. *Cayu- Wiki de Sergio Cayuqueo.* [En línea] [Citado el: 20 de 04 de 2011.] http://wiki.cayu.com.ar/doku.php?id=manuales:servidor_freeradius.
9. Learn networking. [En línea] [Citado el: 08 de 03 de 2011.] ttp://learn-networking.com/network-security/how-kerberos-authentication-works.
10. Cover Pages. *Cover Pages.* [En línea] 23 de 02 de 2010. [Citado el: 10 de 03 de 2011.] http://xml.coverpages.org/saml.html.
11. **Méndez, Daniel Enrique López.** *Estandarización del componente autenticación del Sistema de Gestión Integral de Seguridad (ACAXIA).* 2010.
12. **Rodríguez, Joaquín González.** Criptonomicón. *Identificación biométrica.* [En línea] [Citado el: 01 de 03 de 2011.] http://www.iec.csic.es/criptonomicon/articulos/expertos73.html.
13. **Rojas, David.** Kyuden Akomachi. *Kyuden Akomachi.* [En línea] 17 de 11 de 2010. [Citado el: 22 de 2 de 2011.] http://www.akomachi.com/bandas-magneticas/.
14. **Guillou, Louis Claude, Ugon, Michel y Quisquater, Jean Jacques.** The smart card a standardized security device dedicated to public cryptology. In Contemporary Cryptology. *The Science of Information Integrity.* s.l. : IEEE Press, 1992.
15. **Torres, Márques, Joaquin.** *Nuevo Marco de Autenticación para Tarjetas Inteligentes en Red. Aplicación al Pago Electrónico en entornos inalámbricos.* [En línea] 10 de 2006. [Citado el: 01 de 03 de 2011.] http://www.google.com/url?sa=t&source=web&cd=1&ved=0CBUQFjAA&url=http%3A%2F%2Fe-archivo.uc3m.es%2Fbitstream%2F10016%2F781%2F1%2FTesis_Doctoral-Joaquin_Torres_Marquez.pdf&rct=j&q=Desde%20el%20comienzo%20de%20su%20desarrollo%2C%20las%20tarjetas%20inteligent.
16. *Hand-held computers can be better smart cards,In Proceedings of the 8th USENIX Security Symposium. The USENIX Association.* **Felten, Balfanz, Dirk y W., Eduard.** 1999.
17. *Smart cards in hostile environments. In Proceedings of the 2nd USENIX Workshop on Electronic Commerce. The USENIX Association.* **Gobioff, H, y otros, y otros.** 1996.

18. La nueva.com. [En línea] 23 de 01 de 2011. [Citado el: 01 de 03 de 2011.] http://www.lanueva.com/edicion_impresa/nota/23/01/2011/b1n112/nota_papel.pdf.
19. duradisc. [En línea] [Citado el: 01 de 03 de 2011.] http://www.duradisc.com/es/ayuda-codigo-barras.php.
20. **Gonzalez, Gonzalez, Sergio.** Integración de redes con OpenLDAP, Samba, CUPS y PyKota. [En línea] 2004. [Citado el: 02 de 03 de 2011.] http://www.ibiblio.org/pub/linux/docs/LuCaS/Tutoriales/doc-openldap-samba-cups-python/html/ldap+samba+cups+pykota.html#openldap-que-es.
21. LDAP. [En línea] 30 de 11 de 2004. [Citado el: 03 de 03 de 2011.] http://www.ldap-es.org/contenido/04/11/1.1.-descripci%C3%B3n-de-ldap.
22. Red Hat Enterprise Linux 4: Manual de referencia. [En línea] 2005. [Citado el: 07 de 03 de 2011.] http://www.plonechile.cl/documentacion/rhel-rg-es.pdf.
23. Google Code. [En línea] 2010. [Citado el: 11 de 03 de 2011.] http://code.google.com/intl/es-ES/googleapps/domain/sso/saml_reference_implementation.html.
24. **Rigazzi, Pablo.** Zend Framework. [En línea] 9 de 2008. [Citado el: 08 de 03 de 2011.] http://spanish.zendfw.com/.
25. Zend Framework. *Zend Framework.* [En línea] [Citado el: 04 de 03 de 2011.] http://framework.zend.com/manual/en/zend.auth.adapter.ldap.html.
26. **Hernández Cisneros, Sergio y Sarduy, Mileidy Magalys.** *Propuesta de modelo de desarrollo de software tecnológico del Centro de Soluciones de Gestión.* 2009.
27. The Apache Software Foundation. *The Apache Software Foundation.* [En línea] [Citado el: 01 de 03 de 2011.] http://www.apache.org/.
28. **Douglas, Korry y Douglas, Susan.** *PostgreSql.* 2003.
29. Free Donwload Manager. *Free Donwload Manager.* [En línea] 07 de 03 de 2007. [Citado el: 27 de 02 de 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.
30. **Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato.** Version Control with Subversion. *Version Control with Subversion.* [En línea] 2010. [Citado el: 07 de 03 de 2011.] http://svnbook.red-bean.com/nightly/en/svn-book.html.
31. SOA Agenda. *SOA Agenda.* [En línea] 25 de 10 de 2009. [Citado el: 02 de 03 de 2011.] http://soaagenda.com/journal/articulos/que-son-los-frameworks/.
32. **Gómez Baryolo, Oiner y Silega Martínez, Nemuris.** *Plantilla Registro de la Propiedad Intelectual (Sauxe).* Ciudad de la Habana : s.n., 2008.
33. **Perez, Mata, Manuel.** TecnoRetales. *TecnoRetales.* [En línea] 03 de 07 de 2009. [Citado el: 04 de 03 de 2011.] http://www.tecnoretales.com/tag/doctrine/.
34. Sencha. *Sencha.* [En línea] [Citado el: 2 de 03 de 2011.] http://www.sencha.com/.
35. Clikear.com. *Clikear.com.* [En línea] [Citado el: 05 de 03 de 2011.] http://www.clikear.com/manuales/uml/introduccion.aspx.
36. Manual de PHP. [En línea] 2001. [Citado el: 02 de 03 de 2011.] http://www.php.net/manual/es.
37. **Goodman, Danny y Morrison, Michael .** *JavaScript Bible 5th Edition.* s.l. : Wiley Publishing, 2004.
38. **Pressman, Roger S.** *Ingenieria del Software - Un Enfoque Practico.* s.l. : McGraw-Hill Companies, 2002. 8448132149.
39. **Sommerville, Ian y Sawyer, Pete.** *Requirements Engineering: A good practice guide.* Lancaster University. 1997.

40. *Documento línea base del proyecto.*
41. **Gamma, Erich, y otros, y otros.** *Design Patterns: Elements of Reusable Object-Oriented Software.* s.l. : Addison-Wesley, 1994. 0-201-63361-2.
42. *Un método para el diseño de la base de datos a partir del modelo orientado a objetos.* **Hernández González, Dra. Anaisa.** 4, México DF : s.n., 2004, Computación y Sistemas, Vol. 7. 1405-5546.
43. **Gómez Baryolo, Oiner, I:D.G.T y Rivero Pino, Noel Jesus.** *Memoria colectiva del proyecto Acaxia.* 2011.
44. NetBeans. [En línea] [Citado el: 05 de 03 de 2011.] http://netbeans.org/index_es.html.
45. **Corzo, Giancarlos.** Desarrollo en la Web. *Desarrollo en la Web.* [En línea] 22 de 10 de 2008. [Citado el: 06 de 03 de 2011.] http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/.
46. **Resig, John.** *Pro JavaScript Techniques.* 2006.
47. [En línea] [Citado el: 07 de 03 de 2011.] http://www.idautomatica.com/informacion-tecnica/codigo-de-barras.php.
48. **Sæther,Bakken, Stig; Aulbach, Alexander; Schmid, Egon; Winstead, Jim; Torben, Wilson, Lars; Lerdorf, Rasmus; Zmievski, Andrei; Ahto, Jouni;.** *Manula de php.* 2002.
49. **Valle, Jose.** Web Estilo. *Web Estilo.* [En línea] [Citado el: 04 de 03 de 2011.] http://www.webestilo.com/php/php12a.phtml.
50. **Alvarez, Marañon, Gonzalo.** Que son las cookies. [En línea] 1999. [Citado el: 07 de 03 de 2011.] http://www.iec.csic.es/criptonomicon/cookies/reales.html.
51. **Nuseibeh, Bashar y Easterbrook, Steve.** Requirements engineering: a roadmap. *Proceedings of the Conference on The Future of Software Engineering.* Limerick : ACM, 2000.
52. **IEEE.** *Compilation of IEEE Standard Computer Glossaries, 610-1990.* 1991. 1559370793 .
53. **Kotonya, Gerald y Sommerville, Ian.** *Requirements Engineering: Processes and Techniques.* s.l. : John Wiley & Sons, 1998. 978-0-471-97208-2.
54. **Gómez Baryolo, Oiner.** *Solución informática de autorización en entornos multientidad y multisistemas.* Ciudad de la Habana : s.n., 2010.