ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"IMPLEMENTACIÓN DE MÓDULO DE AUTENTICACIÓN PARA APLICACIONES MÓVILES CORPORATIVAS QUE UTILICEN UN SISTEMA DE SERVICIO DE DIRECTORIO"

TESIS DE GRADO

Previo a la obtención del grado de:

MAGISTER EN SEGURIDAD INFORMÁTICA APLICADA

Lilian Teresa Benavides Ostaiza

Guayaquil – Ecuador

Año 2015

AGRADECIMIENTO

Mi agradecimiento va en primer lugar para Dios, porque ha estado presente con sus bendiciones en cada paso de mi vida profesional y personal. Hago también un agradecimiento especial a mi esposo, Christian Merchán, a quien amo y admiro, por ser ejemplo constante de superación y quien me ha alentado a culminar este trabajo de tesis y alcanzar finalmente los objetivos propuestos.

Lilian Benavides de Merchán

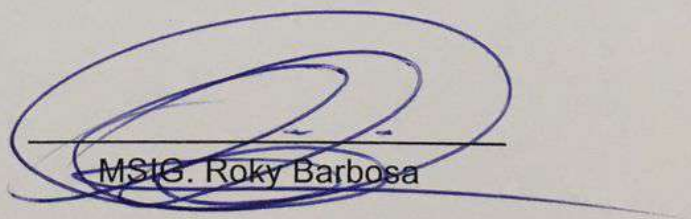
DEDICATORIA

Mi tesis la dedico a mi hija María Mercedes y a mi hijo por nacer, Daniel Enrique, para que conozcan desde ya, que el éxito está en la continuidad del esfuerzo de quien aspira a más y que no hay atajos. Que solo dependerá de ellos tener la perseverancia para alcanzar sus metas, pero que siempre podrán contar conmigo y con su papá cuando sientan que sus fuerzas están por desfallecer. A ellos mi todo mi esfuerzo y mi amor.

TRIBUNAL DE SUSTENTACIÓN

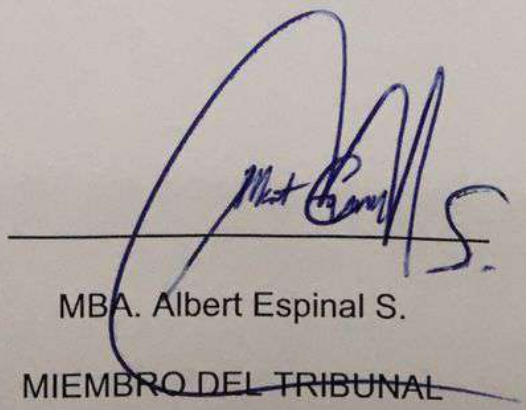
MSIG. Lenin Freire

DIRECTOR DE MSIA



MSIG. Roky Barbosa

DIRECTOR DE TESIS



MBA. Albert Espinal S.

MIEMBRO DEL TRIBUNAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la **Escuela Superior Politécnica del Litoral**"

Lilian Teresa Benavides Ostaiza

RESUMEN

En la actualidad, las empresas apuntan sus esfuerzos a lograr que sus sistemas de información puedan estar disponibles a los usuarios utilizando dispositivos móviles, lo que les obliga tener presente otros aspectos en pro de reforzar la seguridad de los datos, como por ejemplo, desarrollar mecanismos para el control del acceso, lo cual es el objetivo de la presente tesis.

En el capítulo uno, se analizará el problema que genera para las empresas el diversificar las vías de comunicación entre sus empleados, proveedores y clientes, y vamos a plantear la solución para algunas de las vulnerabilidades y dificultades a las que se ven expuestos los sistemas, la autenticación de los usuarios y la integración de diferentes servicios,

En el segundo capítulo, se dará un vistazo a las diferentes opciones que existen entre los tipos de dispositivos móviles, sus sistemas operativos, los componentes básicos para la seguridad y la arquitectura involucrada en un servicio de directorio.

En el tercer capítulo, se definirá de manera funcional nuestra solución, su alcance, sus condiciones y supuestos, incluso los riesgos involucrados.

En el cuarto capítulo, se realizará el diseño de la solución, exponiendo casos de uso, su arquitectura, el plan de pruebas, entre otros.

En el quinto capítulo, se revisará la implementación de la solución y se pondrá en práctica el plan de pruebas,

En el sexto capítulo, se presentará el comportamiento de la solución durante las pruebas y se analizarán los resultados obtenidos

Finalmente, se expondrán las conclusiones y recomendaciones resultantes del presente trabajo de tesis.

ÍNDICE GENERAL

ABREVIATURAS Y SIMBOLOGÍA

ADT	Android Developer Tools
API	Application Interface
APK	Android Application Package
CCWTR	Cisco Connected World Technology Report
CPU	Central Processing Unit
DEX	Dalvik Executable
DIO	Diagrama de Interacción de Objetos
DMZ	Demilitarized Zone
GUI	Graphical User Interface
IDC	International Data Corporation
IDE	Integrated Development Environment
IMEI	International Mobile Station Equipment
IMSI	International Mobile Subscriber Identity
iOS	Mobile Operating System de Apple
IP	Internet Protocol
IPC	Inter Process Communication
IPV4	Internet Protocol Version 4
ISO	International Organization for Standardization
JAR	Java Archive
JDK	Java Development Kit
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol

MB	Megabyte
MVC	Model–view–controller
ODBC	Open Database Connectivity
OS X	Sistema Operativo de Apple
OU	Unidad Organizativa
PID	Process ID
SIM	Subscriber Identity Module
SOAP	Simple Object Access Protocol
SQLite	Relational Database Management System para móviles
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UID	User ID
WSDL	Web Services Definition Language
XML	Extensible Markup Language

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Los usuarios de hoy requieren que los accesos a los servicios que las empresas ofrecen vayan acorde a su estilo de vida, que cada día exige mayor movilidad, esto debido al creciente uso de dispositivos tecnológicos, como teléfonos celulares o tablets, para tareas que van más allá de hacer una llamada o leer un libro.

Por tal motivo, las empresas están desarrollando sus sistemas de información de modo que puedan ser accedidos desde dispositivos móviles, sin embargo en muchas ocasiones, no se toma en cuenta la seguridad de los datos, lo cual a la larga podría ocasionar numerosas pérdidas; o por el contrario, los niveles de seguridad son tan altos que los usuarios terminan con aversión a utilizar el sistema por los múltiples controles que deben superar antes de poder acceder a los servicios informáticos. Esto a pesar de que en la actualidad muchas las empresas cuentan con una infraestructura básica de seguridad como la implementación de un servicio de directorio.

La investigación que se realiza en este trabajo de tesis ofrece la oportunidad de que las empresas cuenten con una solución que les permita que sus empleados, clientes, proveedores, etc., puedan acceder a los sistemas informáticos desde

dispositivos móviles, teniendo una vía de autenticación centralizada haciendo uso de los recursos con los que ya cuentan, como el servicio de directorio y su WebService, facilitando a los administradores el control del acceso y brindando a los usuarios una vía cómoda y a la vez segura.

Para efectos prácticos, se realizará este trabajo de tesis en la empresa LaMerced, a la cual se hará referencia por su nombre o como “la empresa” o “la organización” y que está dedicada a la comercialización de productos químicos para el agro, la misma que está contantemente mejorando su área tecnológica en búsqueda de brindar a sus clientes un mejor servicio, razón por la cual, ha solicitado a su departamento de tecnología, se permita el acceso a los sistemas informáticos desde dispositivos móviles.

CAPÍTULO 1

GENERALIDADES

El desarrollo del presente capítulo identifica, en esencia, la importancia de este trabajo de investigación haciendo hincapié en la necesidad de sostener un sistema corporativo en fuertes bases de seguridad para aplicaciones móviles. Aquí se expone el marco de referencia que rige la investigación bajo un conjunto de objetivos generales y específicos y que propone como meta el cumplimiento de la solución al problema.

1.1 Antecedentes

La empresa multinacional Cisco, pionera en tecnología e innovación, realizó un estudio de investigación en diferentes países de América Latina y Europa en el segundo semestre del año 2014 que permite conocer la tendencia en el uso de la tecnología en el trabajo.

Según Cisco Connected World Technology Report (CCWTR), en [1], existen dos grupos de empleados con cualidades diferenciadas por edad, los de Generación X, llamados así porque tiene su fecha de nacimiento entre el año 1960 y 1980, y los de Generación Y, llamados así porque tienen su fecha de nacimiento entre el año 1980 y 2000, también son llamados “Millenials”. De estos grupos de empleados, el 58% sostienen que la presencia de la tecnología es imprescindible en el trabajo y que se necesita al menos tres o cuatro dispositivos en el trabajo diario, es decir, los encuestados indican que tener una laptop solamente en el trabajo no es suficiente porque además se emplean, teléfonos inteligentes, tabletas digitales, y otros dispositivos de comunicación para realizar las tareas diarias que demanda el empleo, el estudio denomina “Supertasker” o Multitareas a este selecto grupo. Ambos grupos sostienen que para el año 2020 los teléfonos inteligentes y los dispositivos portables (o wearables) serán los dispositivos más importantes en el lugar de trabajo por ende los “Superstakers” serán los recursos humanos más solicitados en las entrevistas de trabajo.

Entre la Generación X y la Generación Y existen discrepancias significativas, el 60% de profesionales de la Generación X y el 81% de los profesionales de recursos humanos piensan que los empleados de la Generación Y son capaces de realizar las tareas más rápidas que los empleados de más edad que utilizan dispositivos móviles y aplicaciones. Además, 7 de cada 10 profesionales de recursos humanos creen que los empleados de Generación Y son capaces de realizar tareas más

rápidas si se les permite usar sus teléfonos inteligentes y las aplicaciones en vez de computador, laptop, etc.

El uso de Internet tiene mayor presencia en nuestro día a día, uno de los resultados más interesantes que devuelve la investigación realizada por Cisco, en [1], compara la necesidad de tener acceso a Internet versus contar con nuestros sentidos elementales, como el olfato, el gusto o el tacto. El 42% de los encuestados menciona que estarían dispuestos a perder el sentido del olfato para tener acceso a Internet, es decir, casi la mitad de los encuestados elegiría el acceso a Internet en lugar del sentido del olfato. Sin lugar a dudas la presencia de los teléfonos inteligentes han incrementado el uso del Internet debido a la comodidad que nos ofrecen para realizar tareas de toda índole y en cualquier momento y lugar, así otro resultado muy valioso de la investigación destaca que la Generación X y Y prefieren los teléfonos inteligentes en lugar de un televisor, y no parece sorpresa, al contrario los teléfonos inteligentes son especialistas en contenido multimedia y tiene muchas ventajas frente a un televisor tradicional.

Finalmente la dependencia que tiene los profesionales con los teléfonos inteligentes influyen en su comportamiento y hábitos, el 54% de los profesionales de Generación Y y el 38% de la Generación X tienen por costumbre, revisar sus teléfonos celulares, como lo primero en hacer al despertar. En las actividades diarias del trabajo, el 70% de los profesionales hacen uso de aplicaciones móviles, siendo el 27%

exclusivas para el trabajo y cerca del 60% reconoce utilizar al menos 10 aplicaciones diariamente.

Las estadísticas mencionadas, en [1], demuestran la tendencia en el uso de Internet, dispositivos móviles y aplicaciones para nuestras actividades diarias y en especial para aquellas que se desarrollan en nuestro lugar de trabajo.

1.2 Descripción del problema

Son cada vez más las empresas que ingresan al mercado con su aplicación móvil para teléfonos inteligentes. Algunas empresas exponen información de sus productos y servicios, otras permiten algunas transacciones sencillas, como realizar un pedido o reservar un asiento de cine, y otras permiten ejecutar transacciones complejas pero muy útiles como consultar saldo de cuenta o transferencias bancarias. También las aplicaciones móviles se implementan en las empresas como herramientas de los empleados, para atender un trámite de un cliente interno o para que los gerentes reciban un reporte mensual de ventas.

Mientras crece la demanda de aplicaciones móviles para ofrecer más servicios, también crecen las vulnerabilidades en las aplicaciones debido a malas prácticas de codificación y poco control de acceso a los datos.

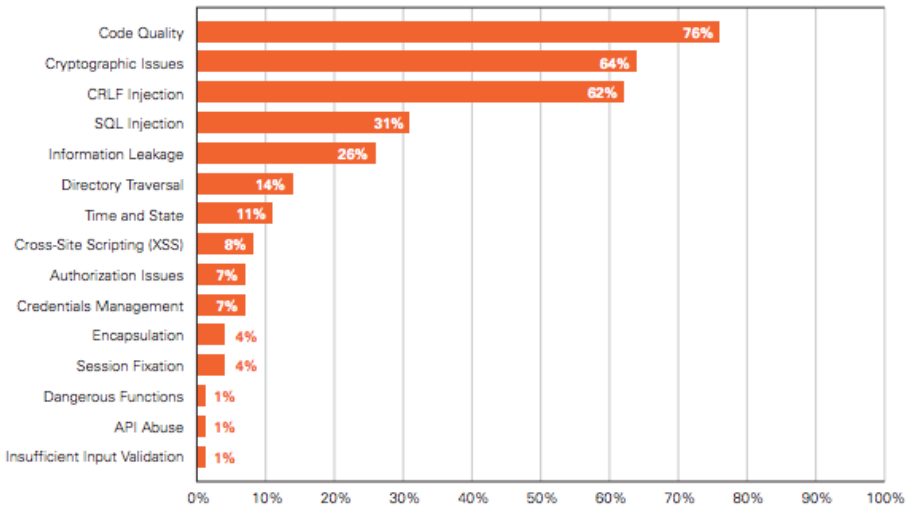


Figura 1.1. Vulnerabilidades en aplicaciones Android. [2]

La figura 1.1 muestra un resumen de las vulnerabilidades del sistema operativo móvil Android según un estudio publicado por la empresa VERACODE, en [1], especialista en seguridad informática.

Entre algunas causas de vulnerabilidad del sistema operativo Android se observa: Problemas de autorización y Administración de credenciales. Estas dos causas, las cuales suman 14%, representan las deficiencias en el acceso y manipulación de las credenciales o cuentas de usuarios. Esto ocurre porque la implementación de un proceso de autenticación es rígido y no emplea una autenticación centralizada, por ejemplo, usando un servicio de directorio en la organización. Muchas veces la autenticación se omite, se implementa parcialmente o simplemente se

crea un nuevo mecanismo de autenticación. Esto ocasiona que los procesos de administración de credenciales sean costosos en tiempo y recurso y poco seguro debido a que existen diferentes cuentas de usuario por cada sistema que utilice la empresa. Un usuario que tenga varias credenciales de acceso para diferentes sistemas de una misma empresa ocasiona un impacto negativo en la usabilidad de la herramienta porque lo considera una carga operativa dentro sus actividades diarias.

Existe una solución empresarial para la administración de usuarios y credenciales denominada servicio de directorio, sin embargo no se tiene un módulo que permita integrar una aplicación móvil a este servicio de forma sencilla y directa. Un servicio de directorio además de centralizar la gestión de credenciales también permite formar grupos de usuarios y definir acciones permitidas. Aprovechar al máximo la funcionalidad de un servicio de directorio permitirá a la organización ahorrar recursos y fortalecer los niveles de seguridad de sus colaboradores y sistemas.

La empresa LaMerced, cuenta con un servicio de directorio, sin embargo, la autenticación a los sistemas informáticos, se mantienen de manera autónoma.

Bajo lo anterior expuesto, el requerimiento de LaMerced de permitir el acceso a los sistemas informáticos desde dispositivos móviles, ha permitido identificar los siguientes problemas:

- No existe gestión centralizada de credenciales.
- Pérdida de control sobre los accesos que posee cada usuario.
- Exceso de carga operativa en la administración de usuarios por no ser utilizado el servicio de directorio para la autenticación en las aplicaciones.

1.3 Solución propuesta

Integrar las aplicaciones móviles con el sistema de servicios de directorio es el principal objetivo de este trabajo de titulación. La solución que se plantea es el desarrollo de un módulo de programación para el sistema operativo Android que brinde acceso a las funciones principales de un servicio de directorio con el propósito de controlar el acceso a una red corporativa mediante el uso de credenciales únicas.

Se ha seleccionado al sistema operativo Android como base para el desarrollo debido a su flexibilidad en codificar y reutilizar librerías de código abierto. Otras de las razones por las cuales se ha seleccionado, es porque tiene una participación en el mercado mundial de más del 80%, según la fuente publicada por IDC, International Data Corporation, en [3]. Ver figura 1.2.

Top Five Smartphone Operating Systems, Worldwide Shipments, and Market Share, 2014Q2 (Units in Millions)

Operating System	2Q14 Shipment Volume	2Q14 Market Share	2Q13 Shipment Volume	2Q13 Market Share	2Q14/2Q13 Growth
Android	255.3	84.7%	191.5	79.6%	33.3%
iOS	35.2	11.7%	31.2	13.0%	12.7%
Windows Phone	7.4	2.5%	8.2	3.4%	-9.4%
BlackBerry	1.5	0.5%	6.7	2.8%	-78.0%
Others	1.9	0.6%	2.9	1.2%	-32.2%
Total	**301.3**	**100%**	**240.5**	**100%**	**25.3%**

Figura 1.2. Participación de Android en el mercado global, en [3]

Esto quiere decir que un módulo desarrollado en Android tendrá un mayor impacto en más dispositivos del mercado que si se lo desarrolla en otros sistemas operativos como Blackberry o Windows Phone que apenas alcanzan juntos una participación del 3% aproximadamente.

Con el desarrollo de éste módulo, la empresa LaMerced podrá dar una solución a la problemática identificada en la sección anterior.

- Implementación centralizada de credenciales.
- Mayor control sobre los accesos que posee cada usuario.
- Fortalecer el uso del servicio de directorio para la autenticación en las aplicaciones.

1.4 Objetivo General

Implementar el módulo de autenticación para aplicaciones móviles corporativas que utilice un sistema de servicio de directorio.

1.5 Objetivos Específicos

- Comprender las necesidades del segmento corporativo y como las aplicaciones móviles permiten innovar su desarrollo.
- Identificar la arquitectura de los sistemas operativos móviles actuales y cómo desarrollar aplicaciones dentro de este entorno con un alto nivel de seguridad sobre los datos.
- Definir la funcionalidad mínima requerida para integrar aplicaciones móviles con un sistema de servicios de directorio.
- Utilizar las mejores técnicas de análisis y diseño de aplicaciones que cumplan los requerimientos del sistema.
- Implementar el módulo de integración entre aplicaciones móviles y un sistema de servicio de directorio.
- Evaluar los resultados obtenidos para garantizar que los requerimientos del sistema están acorde a lo solicitado.

1.6 Metodología

Para cumplir con el objetivo general de esta investigación la metodología que se va a emplear para la implementación del módulo informático será la Metodología en Cascada.

La Metodología en Cascada es la propuesta tradicional para el desarrollo de software cuidando la secuencia de cada etapa durante la fabricación del software. Esta investigación se ajusta a la propuesta planteada debido a su alcance específico y al nivel de experiencia del personal humano que desarrolla el sistema.

CAPÍTULO 2

MARCO TEÓRICO

Los dispositivos móviles desde su primera aparición siempre fueron íconos de innovación y avance tecnológico. En nuestro tiempo, los dispositivos móviles, tienen capacidades técnicas comparables a un computador de escritorio y gracias a esta ventaja ha sido posible la evolución de los sistemas operativos móviles. Este capítulo presenta las características de hardware y software de los dispositivos móviles que los convierten en teléfonos inteligentes y hacen posible un universo de aplicaciones móviles.

2.1 Características físicas de los dispositivos móviles

Los dispositivos móviles están presentes en todo lugar para donde miramos, desde un teléfono móvil que usamos principalmente para hacer llamadas, un televisor que utilizamos para entretenimiento o el

sistema de navegación que posee nuestro vehículo hasta lentes inteligentes con capacidades extraordinarias de conexión a Internet y procesamiento avanzado de comandos de voz. Muchos de estos innovadores dispositivos alcanzan su potencial gracias a los componentes de hardware que conforman su estructura.

Con el pasar de los años el teléfono celular, representante principal de los modernos dispositivos móviles, ha evolucionado muy rápidamente gracias a la evolución de sus componentes de hardware. Estos componentes mejoran cada año en diversas áreas, como el tamaño, la capacidad de almacenamiento o resolución, el consumo de energía, la durabilidad, el soporte a diferentes temperaturas o entornos de ambiente, etc. Estas características de hardware que posee los dispositivos móviles expande la capacidad del dispositivo de ofrecer mucho más que su principal función, por ejemplo los teléfonos inteligentes cuya función básica es hablar por teléfono, pero su atractivo principal es la resolución de pantalla, la capacidad de procesamiento, la memoria, la resolución de la cámara de fotos, etc.

A continuación revisaremos las características principales de un teléfono inteligente:

2.1.1 Capacidad de procesamiento

Desde hace más de 10 años ha existido los procesadores de 64 bits principalmente destinados para equipos de alta capacidad de procesamiento y por ende de costos muy elevados, estos aparatos eran destinados principalmente para el sector corporativo que podía realizar grandes inversiones. Hoy en día estos procesadores han disminuido en costos y están al alcance de economías más livianas, principalmente en el uso de teléfonos inteligentes.

Si la presencia de procesadores de 64 bits es más común en los teléfonos inteligentes, ¿cuál es la razón?, aquí algunas de las principales:

- Un procesador de 32 bits puede realizar cálculos de 0 hasta 4’294.967.295, pero un procesador de 64 bits puede realizar el doble de cálculos, es decir, 18’’’446.744’’073.709’551.615, esto da como resultado, que las operaciones alcanzarán cantidades mayores y los cálculos con cantidades inferiores serán más eficientes.
- Un procesador de 32 bits puede controlar un máximo de 4GB de memoria RAM, pero un procesador de 64 bits puede controlar hasta 16 Exabytes de memoria RAM. Esto significa que podemos tener más aplicaciones abiertas en el teléfono inteligente sin volver lento el funcionamiento.

- Con una mayor capacidad de cálculo, un procesador de 64 bits permite un cifrado más complejo debido a una longitud de clave mucho más larga, esto permite ofrecer un nivel de seguridad más alto en el teléfono inteligente y muy difícil de vulnerar.
- Con 64 bits se logra una mejora notable en el procesamiento de gráficos y videos en 4K, así como la grabación y la compresión. Los videojuegos también aumentan el realismo y los detalles sin causar retrasos ni refrescamiento lento.

2.1.2 Sensores

En el hogar, en el trabajo o en el colegio, la vida moderna de los usuarios exige que los teléfonos inteligentes identifiquen e interpreten el entorno que los rodea para una tener una comunicación más natural y espontánea.

Que un teléfono inteligente reconozca si es de día o de noche, o que permita indicar una ubicación geográfica o que identifique una huella dactilar permite una experiencia de usuario completa, llevando la tecnología a replicar capacidades humanas naturales. Aquí los sensores que la mayoría de los teléfonos inteligentes poseen:

- Sensor de luz, capacidad de un teléfono inteligente de medir la intensidad de la luz exterior. Este sensor sirve para ajustar de manera automática el brillo de la pantalla para ahorrar energía.
- Sensor de proximidad, capacidad de un teléfono inteligente de identificar superficies próximas. Este sensor es muy útil cuando se contesta una llamada telefónica y se quiere evitar activar una opción del teléfono accidentalmente con una parte de la cara.
- Giroscopio, este sensor utiliza la fuerza de la gravedad de la Tierra para medir la orientación del teléfono. Es empleado en juegos de realidad aumentada.
- Sensor de reconocimiento dactilar, el iPhone fue el primer teléfono inteligente en implementar un sensor que permita autenticar a una persona usando sus huellas dactilares. Este sensor está basado en un capacitor táctil que escanea la epidermis a una resolución de más de 500 puntos por pulgada, suficiente para obtener una imagen en alta definición de la huella dactilar.

2.2 Sistemas operativos móviles

La variedad de componentes de hardware que posee un teléfono inteligente y las diferentes interfaces de comunicación hacen

indispensable que exista un sistema operativo para una correcta interacción.

Como se analizó en la figura 1.2, los sistemas operativos predominantes del mercado mundial son Android y iOS, y esta es la razón por la cual se detalla a continuación su arquitectura y características principales.

2.2.1 Arquitectura de los sistemas operativos

Android es un sistema operativo creado en 2005 por una empresa independiente del mismo nombre, Android Inc. Años más tarde, la empresa sería adquirida por Google como estrategia para ingresar al mercado de tecnología móvil. Desde sus inicios, Android fue concebido bajo licencia de código abierto Apache, esto ha permitido que diferentes fabricantes de hardware móvil realicen adaptaciones personalizadas del sistema operativo y se comercialicen con sus dispositivos móviles. Android 1.5 es liberado el 30 de abril del 2009 bajo el código secreto “Cupcake”, de ahí en adelante las demás versiones que existen del sistema operativo siempre tienen su nombre secreto relacionado a un postre, la última versión anunciada de Android, 6.0 tiene por nombre “Marshmallow” o Malvavisco.

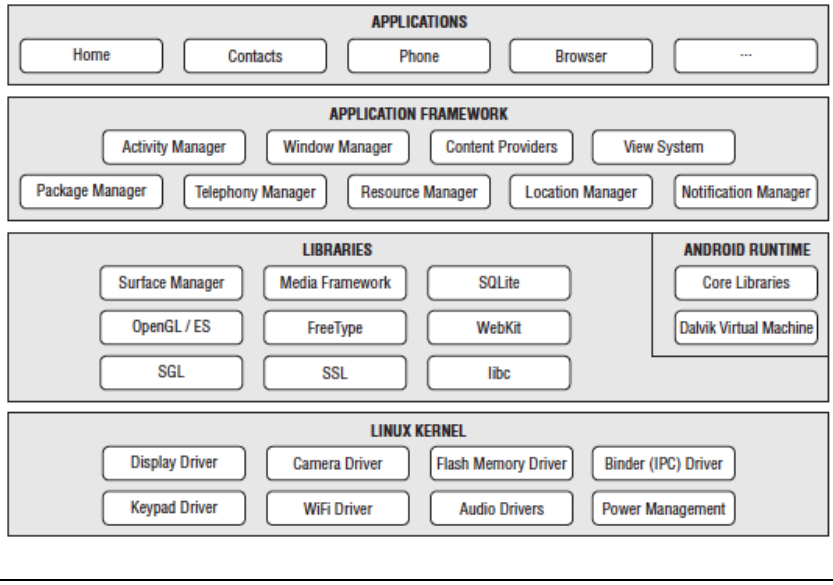


Figura 2.1. Arquitectura del sistema operativo Android. [4]

En la figura 2.1 se visualiza la arquitectura del sistema operativo Android, se identifica cuatro capas o niveles que tienen responsabilidades claras y definidas las cuales revisaremos a continuación:

Linux Kernel, Android está basado sobre Linux y la primera capa de su arquitectura representa la abstracción entre el hardware y el resto de capas. Aquí en el núcleo de Linux encontramos servicios de memoria y procesamiento, de seguridad, de protocolos de comunicación y sobre todo los programas o drivers para interactuar con el hardware del dispositivo.

Libraries / Android Runtime, esta capa segmentada en dos módulos, permite la ejecución de la aplicación accediendo a

librerías nativas en lenguaje de programación C y un entorno de ejecución similar a la máquina virtual de Java pero con adaptaciones propias para Android. Desde sus primera versiones hasta la versión 4.3.x (Jelly Bean), Android utiliza su máquina virtual Dalvik, la cual reúne dentro de sus principales características: la optimización de los recursos, siendo que Dalvik está optimizada para dispositivos con características de hardware limitadas, formato de compresión .dex (Dalvik executable) el cual comprime las clases .class y los recursos para un mejor uso de memoria. Dalvik implementa el acceso por registro en lugar de usar pilas, esto permite aprovechar al máximo el rendimiento de los dispositivos. Las librerías nativas incluyen un amplio conjunto de recursos para el manejo de: Soporte Multimedia para diferentes formatos de audio y video, Surface Manager para el manejo de gráficos en 2D o 3D, SQLite para el manejo de bases de datos, SSL para el soporte de servicios de encriptación y demás librerías básicas nativas del procesador.

Application Framework, posee un conjunto de librerías en alto nivel disponibles para el desarrollador. Esta capa escrita en lenguaje Java provee, entre sus principales servicios, la capacidad para manipular el lienzo de una pantalla, manejo de los ciclos de vida de la aplicación, proveedores de notificaciones, manejos de contenidos, etc.

Applications, la última capa que forma parte de la arquitectura de Android está formada por todas las aplicaciones que posee por defecto el sistema operativo y por aquellas que se instalen sobre él. Aquí encontramos aplicaciones como: Cámara de fotos, Localizador, Navegador Web, Redacción de SMS, etc.

iOS, es el sistema operativo del teléfono inteligente iPhone, causante indiscutible de la revolución de la tecnología móvil. En el año 2007, Apple anuncia en el MacWorld Conference & Expo su incursión en el mundo móvil con su producto innovador iPhone, desde entonces se han anunciado nueve versiones de sus sistema operativo iOS con diversas características que hacen un oponente difícil de superar a Android.

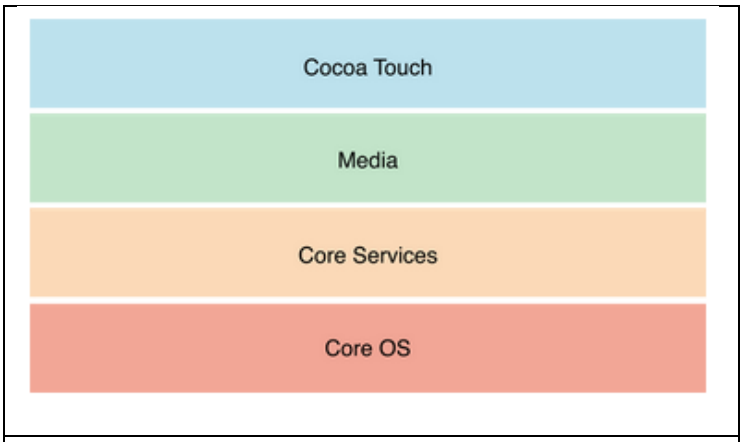


Figura 2.2. Arquitectura del sistema operativo iOS. [5]

En la figura 2.2 se observan las cuatro capas de la arquitectura del sistema operativo iOS donde las capas más altas contienen los servicios y tecnologías más importantes para el desarrollo de aplicaciones, y las capas más bajas controlan los servicios básicos.

Core OS, es la capa de más bajo nivel que provee interfaces basadas en Unix responsables del manejo de drivers que permite la interacción de con el hardware, también permite la administración de la memoria del sistema, multiprocesamiento, manejo de archivos, redes de comunicaciones, etc. Esta capa no se utiliza directamente desde las aplicaciones, las mismas acceden a través de otros frameworks de alto nivel.

Core Services, esta capa es la encargada de proveer y contener todos los servicios básicos y fundamentales del sistema operativo que usan todas las aplicaciones, como por ejemplo SQLite para almacenamiento de información.

Media Layer, esta capa está destinada para el uso de los servicios de gráficos, audio y video. A nivel de presentación gráfica, esta capa provee todas las librerías necesarias para construir aplicaciones enriquecidas en imágenes y animaciones en 2D y 3D, así como también un correcto uso del procesador para manipulación de gráficos. A nivel de audio, provee todos los formatos disponibles para manejo de archivos de audio y además

posee integración con la biblioteca iTunes. Finalmente a nivel de video proporcionar una amplia gama de formatos y niveles de calidad ofrecidos por la versión del dispositivo móvil.

Cocoa Touch, esta capa es una abstracción de la capa Cocoa de OS X, el sistema operativo para Mac, la cual se emplea de forma exclusiva para el desarrollo de aplicaciones iPhone y sus demás dispositivos como Ipad y Ipod Touch. Esta capa ofrece servicios de alto nivel que se emplea directamente por los desarrolladores en la construcción de las aplicaciones, como por ejemplo: manejo de tareas en paralelo, auto layout, Storyboards, notificaciones push, etc.

2.3 Componentes de desarrollo

Durante el diseño de una aplicación móvil es importante definir el comportamiento de la misma cuando atraviesa diversos momentos o estados durante la iteración con el usuario. Esta secuencia de eventos organiza el acceso a los recursos, en envío de mensajes al usuario, o la disponibilidad de la información. Cuando se desarrolla una aplicación para dispositivos móviles son muchos los aspectos que se deben tomar en cuenta y que son relevantes si queremos ofrecer una verdadera experiencia de usuario con acceso a los datos de forma segura, así es

importante conocer los componentes de desarrollo dependiendo del sistema operativo móvil en el que se trabaje.

2.3.1 Ciclo de vida

El ciclo de vida de un programa define los estados y flujo de interacción entre los mismos. En Android, el ciclo de vida de un programa está vinculado al ciclo de vida de una Actividad, una actividad en Android representa la interfaz gráfica y el comportamiento relacionado a la misma. Así tenemos que una aplicación en Android está conformada por un conjunto de actividades, entonces el ciclo de vida de una actividad es el siguiente:

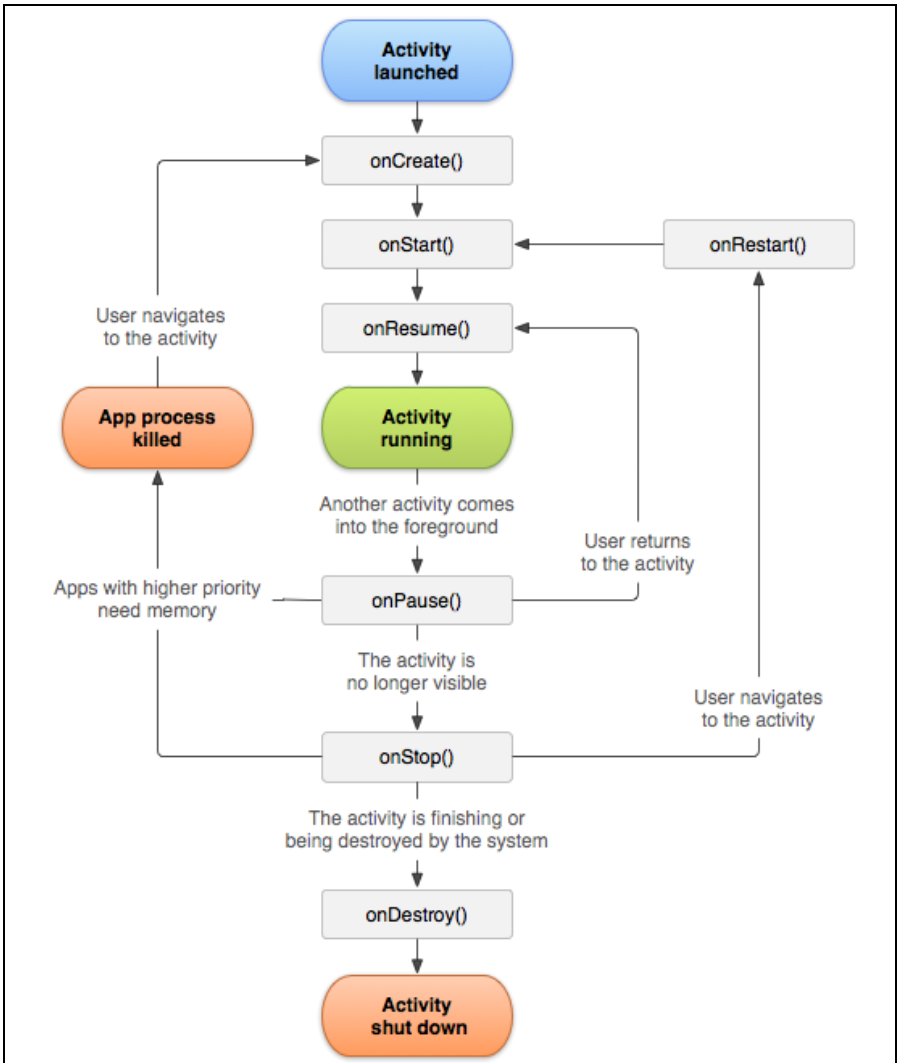


Figura 2.3. Ciclo de vida de una Actividad en Android. [6]

El ciclo de vida de una actividad en Android permite al desarrollador tener el control de la actividad en cada situación al que sea expuesta por el usuario. Por ejemplo: durante la interacción con un juego en el dispositivo podría aparecer una notificación emergente que informe sobre la llegada de un nuevo mensaje de texto, el usuario podría decidir revisar el mensaje e

incluso responder, si esto sucede el juego debería pausarse y detener toda comunicación con los servidores en línea si se tratase de un juego en red. Cuando el usuario regrese al juego el mismo podría reanudarse y retomar el juego justo desde el momento en que recibió el mensaje de texto. Toda esta experiencia de usuario es posible gracias a los diversos estados que tiene una actividad. En la figura 2.3 se observa los estados que una actividad puede tener y también se observa las rutinas o procedimientos que son invocados automáticamente entre estado y estado.

A continuación revisaremos el comportamiento de cada método:

- onCreate(), este método es invocado cuando la actividad es creada, aquí el sistema operativo asigna los recursos de memoria y procesamiento que se requiere. Se ejecuta una sola vez si la aplicación no se encuentra en memoria.
- onStart(), este método se invoca cuando la aplicación es visible al usuario.
- onResume(), este método se invoca cuando la aplicación empieza a interactuar con el usuario, es invocado inmediatamente después del método onStart().

- onPause(), este método es usado principalmente para almacenar de manera segura el estado de la transacción, detener hilos de animación o liberar recursos.
- onStop(), este método es invocado cuando la actividad ya no es visible al usuario, es decir, ya no está disponible.
- Otros métodos como onDestroy() y onRestar() permiten detener completamente la aplicación y liberar recursos o reanudar la aplicación desde el método onStart().

A diferencia de los diversos estados que Android nos entrega en una actividad, en iOS los estados son pocos, se presenta a continuación:

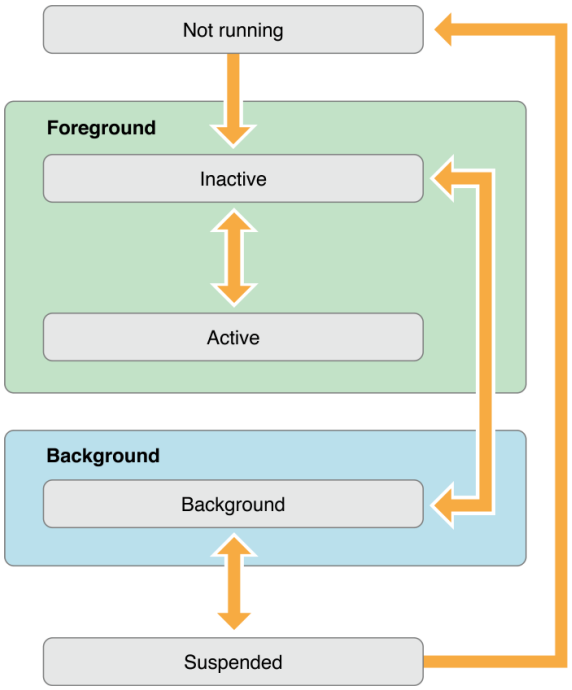


Figura 2.4. Ciclo de vida de una aplicación en iOS. [7]

El ciclo de vida de una aplicación en iOS cambia de estado de acuerdo a la interacción del usuario. A continuación revisaremos cada uno de los estados mostrados en la figura 2.4.:

- Not Running, la aplicación no ha sido iniciada o la misma fue terminada por el sistema.
- Inactive, este estado indica que la aplicación esta en primer plano pero que no posee interacción con el usuario.

- Active, la aplicación esta en primer plano y recibe eventos del usuario.
- Background, la aplicación está en ejecución y en segundo plano. Este estado se presenta para ejecutar rutinas sin que se espere por la participación del usuario.
- Suspended, la aplicación esta en segundo plano solamente, no está en ejecución. En este estado la aplicación se mantiene en memoria sin ejecución de ningún código.

2.3.2 Bases de datos móviles

Las aplicaciones para dispositivos móviles inteligentes en la actualidad presentan características muy robustas altamente comparables a aplicaciones de escritorio, es así como tenemos inclusive bases de datos móviles que ofrece persistencia en el almacenamiento de información. Las bases de datos móviles nacieron por la necesidad de almacenar la información ingresada por el usuario en la aplicación de manera persistente y local sin tener que depender de una conexión a una red. Tomando en cuenta las limitaciones en hardware que poseen los teléfonos inteligentes existe una base de datos que se adapta a estas limitaciones, es SQLite.

A continuación, algunas de las razones de por qué usar SQLite [8]:

- Dispositivos de bajo rendimiento, SQLite es ideal para dispositivos de bajo rendimiento con escasos recursos de hardware y no requiere administración ni soporte experto.
- Formato de almacenamiento, SQLite es transparente al momento de almacenar la información al sistema de archivos, puesto que no implementa rutinas complejas como bases de datos tradicionales, se almacena como un archivo de texto simple.
- Alta transaccionalidad, SQLite posee gran capacidad de respuesta ante una alta demanda de transaccionalidad.

SQLite se distribuye bajo licencia de Public Domain que significa que es de libre acceso para el código fuente y la documentación. Los sistemas operativos Android e iOS utilizan SQLite en la persistencia de información de sus aplicaciones.

2.4 Componentes de seguridad

A través de los tópicos anteriores se aprendió la arquitectura de Android y la comunicación entre sus principales componentes logrando así que

las aplicaciones móviles se comporten lo más real posible a aplicaciones de escritorio. Sin embargo, no hay espacio a discusión cuando de la seguridad se habla, y precisamente en el sistema operativo Android es un tópico del cual hay mucho interés. Uno de los pilares fundamentales de este trabajo de investigación es la seguridad que envuelve a las aplicaciones móviles dentro del entorno del sistema operativo Android, el modelo de seguridad que trae consigo valiosas ventajas y confianza al momento de construir aplicaciones móviles. A continuación se presenta los aspectos más importantes del modelo de seguridad de Android.

Application Sandbox

Uno de los principios más básicos de seguridad en sistemas informáticos es el de más bajo privilegio. En Android las aplicaciones se instalan usando el principio del más bajo privilegio o mínimo privilegio haciendo que las aplicaciones se ejecuten en un sandbox (o arenero en español), es decir, cada aplicación tiene acceso a sus propios recursos por defecto y tiene acceso restringido a los recursos de otras aplicaciones excepto que se otorgue el permiso correspondiente.

Este complejo mecanismo de seguridad viene implementado desde el kernel del sistema operativo Android, el cual es basado en Linux. El proceso consiste en asignar a cada aplicación, luego de ser instalada, un identificador único de usuario denominado UID (llamado también App ID),

a diferencia del esquema tradicional en Linux donde el UID hace referencia al usuario que establece la comunicación (User ID), en Android el usuario es implícito por defecto y el UID se asigna a la aplicación que se instala diferenciándola de las demás.

Permisos

Mediante el UID las aplicaciones no comparten recursos de forma implícita, sin embargo hay ocasiones en que se requiere usar recursos compartidos de otras aplicaciones o simplemente se desea acceder a recursos del sistema operativo, como el directorio telefónico, la cámara de fotos, la tarjeta de memoria, etc. Basados en el principio de mínimo privilegio, las aplicaciones sólo tienen acceso a los recursos propios y es necesario declarar en las propiedades de la aplicación los permisos adicionales que requiere, así cuando se instale la aplicación en el sistema operativo recibirá los permisos solicitados o caso contrario si no se pueden otorgar, la aplicación no se instalará. Los permisos son otorgados a través del archivo de configuración de aplicación AndroidManifest.xml, y según el nivel de protección que posee el recurso solicitado, el sistema emplea el procedimiento para asignar el permiso o no asignar el permiso. A continuación los cuatro niveles posibles de asignación:

- Normal, el más básico de los permisos, permite a la aplicación utilizar ciertos recursos aislados con bajo riesgo para otras aplicaciones, el sistema o usuario. Estos permisos se otorgan automáticamente a la aplicación cuando lo solicite sin aprobación explícita del usuario, por ejemplo: SET_WALLPAPER es la capacidad de cambiar el fondo de pantalla mediante la aplicación que obtiene el permiso.
- Dangerous, el siguiente nivel de permisos que habilita a la aplicación a utilizar recursos, como información privada o control del dispositivo, que pueden comprometer negativamente al usuario, por ejemplo: INTERNET es un permiso catalogado dentro de este nivel porque permite al dispositivo móvil acceder a Internet pudiendo compartir información con terceras personas ajenas a la organización.
- Signature, permiso que se concede siempre que la aplicación solicitante esté firmada digitalmente con el mismo certificado de la aplicación que definió dicho permiso. Si los certificados coinciden la aplicación recibe los privilegios automáticamente sin solicitar autorización al usuario explícitamente.
- Signature/System, representan los permisos más restrictivos y sólo son concedidos a aplicaciones propias del sistema. Es posible que estos permisos se asignen a otras aplicaciones siempre que estén firmadas con el certificado de las aplicaciones propias del sistema.

Binder

Las aplicaciones se ejecutan en un entorno restringido, gobernado por los principios de mínimo privilegio, la única forma de comunicación entre aplicaciones es mediante mecanismos de IPC (Inter Process Communication), función básica de sistema operativo Linux, cuyo principal objetivo es comunicar dos o más procesos para intercambio de mensajes, sincronización, uso de memoria compartida, etc. A pesar de que Android ofrece los procesos de IPC tradicionales (basados en sockets, señales y demás) es altamente recomendado que la comunicación de las aplicaciones se realice usando Binders.

Binder es un nuevo mecanismo IPC exclusivo para Android que permite efectuar las comunicaciones entre aplicaciones más ágilmente usando una implementación más eficiente haciendo uso de la memoria del kernel del sistema operativo compartida por todos los procesos. Binder utiliza una arquitectura de componentes distribuida basada en interfaces abstractas y una nueva implementación de RCP escrita en C++ para una alta productividad.

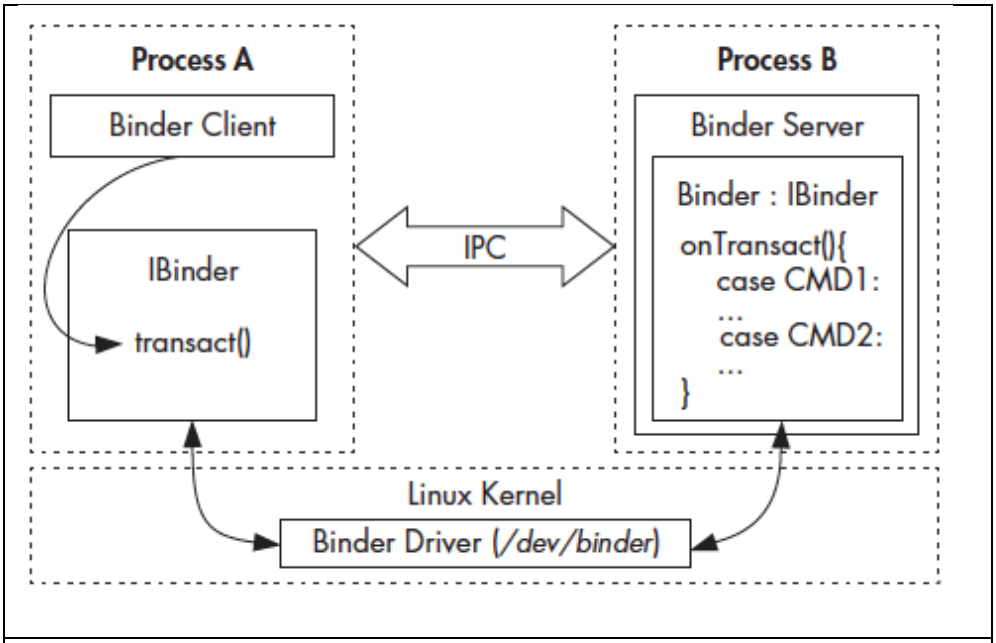


Figura 2.5. Binder IPC, en [10]

En la figura 2.5 se observa que la interface Binder driver, alojado en el espacio de dirección del kernel, es el proceso central que controla los mensajes de acceso a recursos que provienen de procesos de las aplicaciones, esto permite que se garantice que el UID y PID (Process ID) de los procesos no se pierdan en la comunicación y al contrario sirvan para determinar dinámicamente si tienen acceso o no a API del sistema vía IPC.

Code Signing

En Android, todas las aplicaciones, sean estas del sistema o de usuario, tienen el código fuente certificado digitalmente por el desarrollador. En

los desarrollos tradicionales de Java, el código fuente viene empaquetado en un formato JAR (Java Archive) y se utiliza este formato para firmar digitalmente, en Android se tiene una extensión del formato JAR denominado APK (Android Application Package) y utiliza el mismo para firmar el código fuente lo cual permite controlar que las actualizaciones de una aplicación móvil viene de una misma fuente, es decir, vienen del mismo desarrollador. El proceso consiste en comparar el certificado digital que reside en la aplicación instalada en el dispositivo móvil con el certificado digital que viene en la nueva aplicación móvil por actualización.

2.4.1 Métodos de autenticación

La autenticación consiste en determinar que el usuario que está intentando un acceso a un sistema, sea efectivamente quien dice ser, y que efectivamente cuente con el nivel de permisos apropiados para las tareas a las que está intentando realizar.

Para lograr este objetivo, contamos con tres categorías para reconocer al usuario:

- “Algo que conoce” como una contraseña o frase, sin que la haya compartido a otra persona.

- “Algo que posee” como las tarjetas de coordenadas, usb token, una simcard.
- “Algo que es” refiriéndose a una característica física del usuario, huellas digitales, retina, su voz.

En un sistema de autenticación, estas categorías pueden usarse independientes o una combinación de ellas.

2.4.2 Normas de seguridad en desarrollo de aplicaciones móviles

La seguridad en el desarrollo de aplicaciones móviles está influenciada directamente en la plataforma en que se desarrolla, sin embargo hay algunas directrices que se pueden generalizar sin importar el sistema operativo que usemos.

Comunicación TSL/SSL

Es uno de los esquemas básicos y seguros en el transporte de mensajes entre dispositivos. Ambos TSL (Transport Layer Securituy) y SSL (Secure Sockets Layer) permiten enviar y recibir mensajes bajo un canal seguro, donde la información no viaja en texto plano, pero no se debe descuidar el aseguramiento de la

disponibilidad ni la integridad de la información, estos son aspectos claves que complementan la utilización de TLS/SSL.

Programación segura

Otro aspecto importante dentro de las normas de seguridad del desarrollo de aplicaciones móviles es la programación o codificación segura. Sin importar el lenguaje de programación que se use, la buena codificación al momento de escribir el programa será un pilar fundamental en evitar la aparición de vulnerabilidades que puedan ser aprovechadas por un intruso. Aquí algunas recomendaciones:

- Uso de librerías, cuando se codifica un programa y se necesita una librería externa, se puede hacer uso de la librería siempre que se conozca su procedencia y sea de un proveedor seguro. Por otro lado se recomienda usar explícitamente la funcionalidad requerida y no dejar abierta toda la funcionalidad de la librería externa, por ejemplo: si se usa una librería para enviar notificaciones por correo electrónico, usar explícitamente el protocolo deseado, y los demás protocolos que vienen en la librería inhabilitarlos o no instalarlos.

- Entorno de desarrollo actualizado, en la etapa de implementación de un software se recomienda hacer uso de los IDE (Integrated Development Environment) certificados en el lenguaje de programación y que provengan de fuente segura. Adicional se debe mantener actualizada la herramienta de desarrollo con todos los parches de seguridad aplicados.
- Manejo de variables, en la codificación se debe tener control total sobre todos los tipos de variables en memoria que se creen, lo correcto es emplear solamente las necesarias y liberarlas de memoria cuando no se requiera.
- Codificación exacta, se debe codificar de manera exacta, de manera precisa y evitar ambigüedades o múltiples resultados que pueden ser aprovechados para un Code Injection.

Modelo de seguridad del S.O.

Cada sistema operativo provee mecanismos de seguridad básicos que deben seguirse. En Android, el control de la seguridad se basa principalmente en el aislamiento de recursos compartidos, cada aplicación dentro del sistema operativo debe solicitar permisos explícitamente para acceder a un recurso compartido. Por ejemplo: si una aplicación móvil requiere hacer

uso del recurso de Cámara de fotos, se debe configurar explícitamente el permiso de acceso a la Cámara de fotos, de lo contrario la aplicación no puede interactuar con la cámara. Es importante además estar pendiente de las últimas actualizaciones en seguridad del sistema operativo así como del marco de trabajo, así por ejemplo si se determina que un API será reemplazado por fallas en seguridad demostradas, se debe inmediatamente cambiar el código fuente que hace uso de ese API en la aplicación móvil.

Marco de trabajo estable y seguro

Para el desarrollo de las aplicaciones en general se debe emplear marcos de trabajo o Frameworks de amplia trayectoria y soporte, usualmente estos modelos de trabajo están siempre actualizados en términos de seguridad y controlan cuidadosamente sus cambios de versión. La inexistencia de un marco de trabajo o de uno que no tiene suficiente soporte aumenta la probabilidad de un código vulnerable y sensible a ataques.

Almacenamiento de información sensible

Es altamente inseguro almacenar información sensible, como credenciales de acceso, claves de cuentas bancarias, registros médicos, etc. en texto plano, sin encriptación. Esta información se verá comprometida si el dispositivo móvil es hurtado o si es intervenido por un hacker. Los sistemas operativos como Android o iOS ofrecen APIs para encriptación de información sin necesidad de utilizar módulos externos.

Certificado de firma de código

Las aplicaciones móviles tienen certificado de firma de código que identifica al desarrollador o a una entidad que son autores y responsables de la aplicación y de su codificación. El proceso de certificación es obligatorio para alojar la aplicación móvil en las tiendas virtuales como Play Store para Android o App Store para iOS. En algunos casos el certificado de firma de código permite habilitar accesos restringidos a recurso del sistema si no se tuviera un certificado.

2.4.3 Registro y control de intentos de acceso

De acuerdo a la norma ISO/27001, un control de seguridad debe considerar que las acciones de una entidad se pueden rastrear

hasta una única identidad (trazabilidad) y tener la capacidad de mostrar que una acción o evento tuvo lugar, sin que dicha acción o evento pueda ser negada posteriormente (no repudio), para tal efecto se debe implementar un registro en donde se detalle quién, cuando, desde que dispositivo y si el intento de acceso fue exitoso o no.

2.5 Servicio de directorio

Un directorio es una base de datos que contiene información sobre los objetos de una red como usuarios y recursos con sus respectivos atributos, organizados de una manera lógica y jerárquica. Un servicio de directorio es la aplicación o conjunto de aplicaciones que nos permite almacenar, organizar el directorio y gestionar el acceso de los usuarios a los recursos de nuestra red.

Un servicio de directorio resulta indispensable, cuando tenemos una infraestructura con muchas aplicaciones y usuarios, porque nos facilita la administración dado que podemos reutilizar la información, obteniendo integridad y eficiencia de los datos, y mayor flexibilidad y escalabilidad.

El protocolo que permite el acceso a un servicio de directorio es LDAP (Lightweight Directory Access Protocol - Protocolo Ligero de Acceso a Directorios). Existen varias implementaciones de este protocolo, pero algo que todas tienen en común es la estructura lógica, la cual se

compone de elementos intangibles como objetos, dominios, árboles y bosques.

Objetos: es el bloque de construcción básico, un conjunto de atributos que representa un recurso de la red. Los atributos del objeto son características de objetos del directorio. Los objetos se pueden organizar en clases, que son agrupaciones lógicas de objetos. Los usuarios, grupos y equipos son ejemplos de clases de objeto diferentes. El nivel más bajo de objetos se denomina **hoja** (un usuario, un equipo) y no pueden contener otros objetos. Los objetos hoja, se pueden colocar dentro del objeto **contenedor**, lo que ayuda a simplificar la organización del directorio. El objeto contenedor más común es la Unidad Organizativa (OU).

Dominio: se usa para agrupar objetos, con el fin de reflejar la red de una organización, y representa un límite de seguridad. Toda hoja o contenedor, puede existir solamente dentro de un dominio.

Árbol: agrupación de dominios relacionados para compartir recursos globales, que comparten una definición formal de los objetos, denominado esquema.

Bosque: agrupación de árboles, que no requieren tener el mismo esquema de denominación, operan de manera independiente y pueden comunicándose entre sí.

Entre las implementaciones LDAP más destacadas tenemos:

Active Directory

Es la implementación de servicio de directorio de Microsoft, desde Windows 2000, los datos son jerárquicos, replicados y extensibles [15].

OpenLDAP

Es una implementación conforme a LDAPv3, de código abierto, resultante de un esfuerzo colaborativo para crear una suit de aplicaciones y herramientas de LDAP, desarrollado por el proyecto OpenLDAP, el cual es administrado por una comunidad de voluntarios [16].

OpenLDAP puede ser utilizada en diversas plataformas como Linux, Microsoft Windows, Apple MacOS/X, Sun Solaris, IBM, entre otros, esta compatibilidad lo hace ideal para trabajar en ambientes heterogéneos.

2.5.1 Arquitectura y componentes

Como pudimos revisar en el punto anterior, las diferentes implementaciones están basadas en LDAP, razón por la cual comparten su estructura lógica:

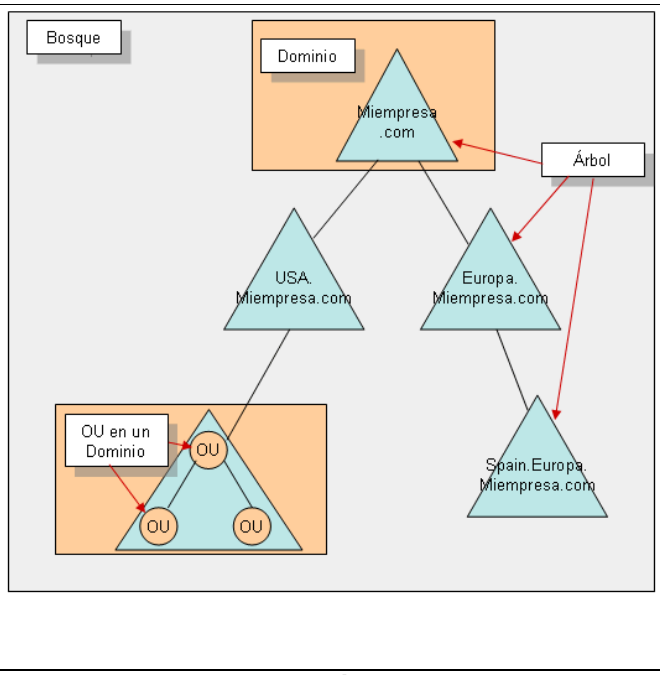


Figura 2.6. Estructura lógica de LDAP, en [11]

Las implementaciones de LDAP difieren entre sí, por sus elementos o componentes. Los elementos de Active Directory son:

Controladores de Dominio, los cuales son servidores ejecutando Windows Server y el Directorio Activo, el cual se compone de 3 particiones [17]:

- Contexto de Nombres de dominio, la cual contiene usuarios, grupos, unidades organizativas.
- Esquema, contiene clases y definiciones de atributos, incluso se puede agregar clases propias o atributos.
- Configuración, que contiene la topología del bosque (esquema de conexión de los sitios).

Sitios del Directorio Activo, que son grupos de equipos conectados y que pertenecen al dominio principal.

OpenLDAP tiene tres componentes principales:

- **Slapd**, demonio LDAP autónomo, en donde se utiliza una directiva de configuración syncrepl para que el demonio Slapd adicionalmente ejecute la replicación de directorios.
- **Biblioteca de rutinas** para soporte del protocolo LDAP.
- **Utilidades**, herramientas como ldapsearch, ldapadd, ldapdelete, entre otros.

CAPÍTULO 3

DESCRIPCIÓN FUNCIONAL

Con el propósito de ser exactos en atender todos los requerimientos que reúne el módulo de autenticación para aplicaciones móviles, se describe en este capítulo la parte funcional, incluyendo los requerimientos de seguridad, que debe satisfacer el presente trabajo de tesis, así también las condiciones y supuestos que asegura la viabilidad del proyecto y que permite el enfoque en el verdadero problema. Al final se revisará los riesgos más importantes que podrían afectar el proyecto pero con alternativas que permita mitigarlos.

3.1 Alcance y restricciones del sistema

Para cumplir los objetivos del sistema es necesario definir el alcance y restricciones del mismo, esto servirá para definir los entregables y

garantizar que el desarrollo está acorde con lo definido sin adicionar o quitar funcionalidad de la que se requiere.

El alcance y los límites del sistema han sido establecidos en base a las necesidades de seguridad en la autenticación de aplicaciones móviles. También debe reunir exigentes pruebas de calidad y un procedimiento que guíe a la empresa LaMerced a su correcta implementación. Todos estos aspectos se revisarán a continuación:

3.1.1 Alcance y límites del sistema

Con el propósito de dimensionar la funcionalidad con que debe contar el módulo de autenticación para dispositivos móviles utilizando un servicio de directorio, se tiene el siguiente conjunto de requerimientos:

- Autenticación centralizada, se utilizará el esquema de autenticación mediante el servicio de directorio ya establecido en LaMerced, en donde se añadirá a los objetos los atributos necesarios para cumplir los propósitos del presente trabajo de investigación, incluyendo los necesarios para identificar al dispositivo móvil desde el cual se intenta el acceso.

- Autenticación segura por Internet o Intranet, el servicio de autenticación debe estar disponible a los usuarios de la desde la red interna o desde Internet, por medio de un canal seguro y utilizando el WebService ya establecido en la organización, al cual se le añadirán los atributos requeridos por el módulo a desarrollar y su correspondiente comunicación con el servicio de directorio.
- El registro de los nuevos atributos es responsabilidad de la organización previa definición y configuración en los sistemas anexos que lo requieran.
- Módulo de autenticación para teléfonos inteligentes, se debe implementar un módulo o librería compatible con Android que facilite la autenticación del usuario desde una aplicación móvil de la empresa hacia el servicio de directorio de la misma, haciendo uso del WebService de la organización. Las aplicaciones móviles corporativas desarrolladas en Android podrán integrar en sus procesos de validación de usuario las librerías de autenticación, considerando esta gestión como un proceso de actualización de versión en las aplicaciones y a responsabilidad de LaMerced. El módulo de autenticación debe reunir las mejores prácticas en codificación segura

para evitar o minimizar las vulnerabilidades de código fuente.

- Registro de accesos, todo intento de ingreso al sistema por el nuevo módulo de autenticación debe registrar un evento en un repositorio definido. Todos los eventos generados, sean estos de éxito o error deben almacenarse para un consulta posterior.

3.1.2 Condiciones y supuestos

Para cumplir con la funcionalidad definida en el alcance es importante la acción sobre algunas condiciones y supuestos que existen. A continuación se repasa las condiciones y supuestos que se han determinado para el cumplimiento del presente trabajo de investigación:

- La organización cuenta con un servicio de directorio que utiliza el protocolo LDAP y con un WebService compatible con la versión del lenguaje de programación utilizada.
- LaMerced cuenta con una infraestructura de red segura, con hardening de seguridad en sus servidores y sus servicios se encuentran debidamente configurados para que sólo se permita el acceso esperado.

- La base de datos del servicio de directorio permite adicionar nuevos atributos en el perfil de los usuarios y existe un administrador que realiza esta operación. En la fase inicial de la puesta en producción al administrador debe registrar a cada usuario, los nuevos atributos para que puedan autenticarse sin dificultad.
- Las aplicaciones móviles de la organización tendrán como sistema operativo Android y deberán actualizar su versión en la tienda virtual de Play Store una vez que han integrado el módulo de autenticación.
- Una vez acreditado el acceso mediante el módulo de autenticación, los permisos que disponga el usuario a las opciones de los sistemas de LaMerced son responsabilidad de la misma y no se verán afectados con la integración del módulo de autenticación.
- La organización cuenta con políticas y procedimientos documentadas e implementadas para la gestión de la seguridad.
- Los dispositivos móviles deben ser originales de fábrica, sin ningún tipo de alteración de hardware o software por terceros.

3.1.3 Riesgos

En el ciclo de vida de un proyecto siempre están presentes los riesgos, los cuales identifican factores negativos que pueden alterar la ejecución normal de un proyecto.

El impacto potencial de que se haga caso omiso de un riesgo dentro del proceso de autenticación, van desde daño de la imagen de la organización, pérdidas financieras, publicación no autorizada de información sensible, daños personales, hasta consecuencias civiles o penales.

Para el presente análisis de riesgo, LaMerced ha delegado un grupo de personas de diferentes áreas, la responsabilidad de identificar las amenazas y vulnerabilidades asociadas al proceso de autenticación, clasificándolos y, de ser necesario, estableciendo los puntos de control para una pronta gestión con el fin de mitigar el riesgo, esto acorde a la cláusula 4, sobre la Evaluación y Tratamiento de Riesgo de ISO 27002:2013 [19]. Para mayor detalle del análisis realizado, revisar el ANEXO A.

Se hará referencia a los delegados como el “grupo evaluador de riesgos”, los cuales han identificado las siguientes amenazas:

- No reconocer a un usuario legítimo como tal: Un usuario que debe tener acceso al sistema, no se le concede el acceso desde su dispositivo móvil.
- Aceptar como usuario legítimo a quien no lo es: Un usuario que realiza un intento al sistema, se le concede el acceso a pesar de no contar con privilegios para el mismo.
- Olvido de credenciales de autenticación: Un usuario legítimo del sistema, no recuerda los mecanismos de autenticación que debe proporcionar para lograr el acceso al sistema.
- Robo de credenciales: Un tercero ha logrado tener acceso a las credenciales o dispositivos móviles de un usuario legítimo.
- Ataque de fuerza bruta: Un tercero trata de forzar el ingreso al sistema, realizando varios intentos hasta lograr su cometido.
- Ausencia del canal de comunicación: No se obtiene un medio de conexión entre el dispositivo móvil y el servidor de aplicaciones de la empresa.

Valoración de las amenazas:

El grupo evaluador de riesgos, ha clasificado cada evento de riesgo de acuerdo al tipo de pérdida que ocasionaría si se materializa y le ha asignado a cada uno una probabilidad de que ocurra dicho evento, tal como vemos en la tabla 1.

Tipo de Pérdida	**Evento de riesgo**	**Riesgo Inherente**	
		Probabilidad	**Impacto**
Pérdidas financieras	Robo de credenciales	4	4
Publicación no autorizada de información sensible	Aceptar como legítimo a un usuario que no lo es	2	4
Indisponibilidad del servicio	Ausencia del canal de comunicación	1	2
Pérdida de imagen	No reconocer a un usuario legítimo como tal	1	3
Discontinuidad del	Ataque de	4	3

negocio	fuerza bruta		
Incumplimiento de la misión empresarial	Olvido de credenciales de autenticación	2	1

Tabla 1. Valoración de riesgos

De la Figura 3.1, el grupo evaluador concluye que se debe implementar controles a los eventos de robo de credenciales, ataque de fuerza bruta y aceptar como legítimo a un usuario que no lo es. Para los demás eventos, se va a asumir el riesgo.

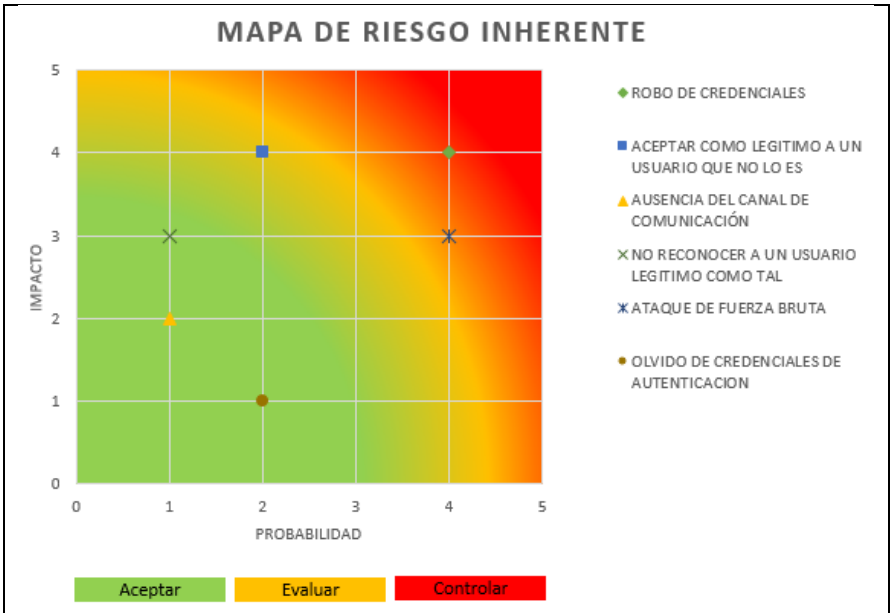


Figura 3.1. Probabiliad vs impacto de eventos. Riesgo inherente

Tratamiento del Riesgo

El grupo evaluador de riesgos, haciendo uso de ISO/IEC 27002, ha decidido que los eventos se tratarán de la siguiente manera:

- Robo de credenciales

    - **Registro de Usuario:** Debe existir un procedimiento formal para el registro y de-registro de usuario para otorgar y revocar el acceso a todos los sistemas y servicios de información de LaMerced. La definición de este procedimiento está a cargo de la organización.

    - **Identificación del equipo en las redes:** La identificación automática del equipo se debiera considerar como un medio para autenticar las conexiones de ubicaciones y equipos específicos.

    - **Cierre de una sesión por inactividad:** Las sesiones inactivas debieran ser cerradas después de un periodo de inactividad definido. El tiempo de inactividad máximo permitido debe estar estipulado formalmente en los procedimientos de seguridad de LaMerced.

- Ataque de fuerza bruta

    - **Validación de la input data:** Se debiera validar la input data para las aplicaciones para asegurar que esta data sea correcta y apropiada.

- Aceptar como usuario legítimo a quien no lo es.

    - **Retiro de los derechos de acceso:** Los derechos de acceso de todos los usuarios empleados, contratistas y terceras personas a la información y los medios de procesamiento de información debieran ser retirados a la terminación de su empleo, contrato o acuerdo, o debieran ser reajustados de acuerdo al cambio. Este procedimiento debe ser parte de los existentes en LaMerced.

    - **Restricción del acceso a la información:** El acceso de los usuarios y el personal de soporte a la información y las funciones del sistema de la aplicación debiera limitarse en concordancia con la política de control de acceso definida previamente en la empresa.

Una vez aplicado los controles a los eventos de riesgo, se obtiene un riesgo residual, el cual el grupo evaluador de riesgos ha analizado de acuerdo a la Figura 3.2 y a los cuales, se ha decido no implementar controles futuros, dado que los eventos de riesgos no fueron catalogados como altos o extremos.

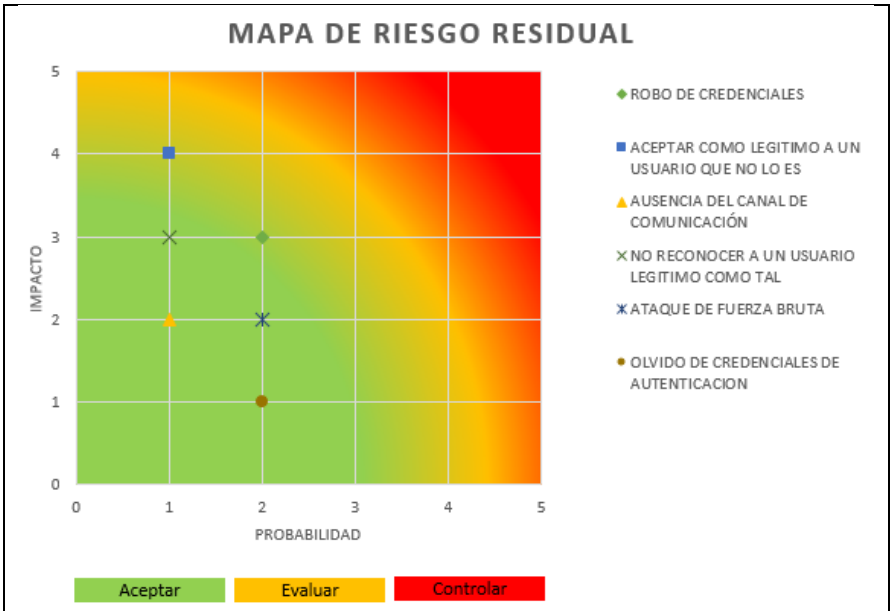


Figura 3.2. Probabiliad vs impacto de eventos. Riesgo residual.

3.2 Descripción funcional

A continuación se detallan las definiciones funcionales que deben considerarse para la implementación del módulo de autenticación:

3.2.1 Arquitectura de módulo de autenticación

Las aplicaciones móviles de la organización contarán con la capacidad de autenticación hacia el servicio de directorio de confianza. La capacidad de autenticación se consigue gracias al módulo o librería para Android que permite un acoplamiento entre las actividades principales de la aplicación y las funciones que se exponen a través de las librerías. En la figura 3.3 se observa la representación de las principales entidades de la aplicación móvil Android así como también de las entidades que son alojadas en el servidor de aplicaciones de lado de la organización.

Dentro de la aplicación móvil se observa a la Actividad Principal, quien es responsable de la orquestación de los mensajes que deben transmitirse entre el Modulo de Autenticación y el subproceso Manejador de Conexiones. La comunicación entre las entidades se deberá realizar tomando de referencia el concepto de desarrollo, Modelo-Vista-Controlador, modelo de trabajo en Android.

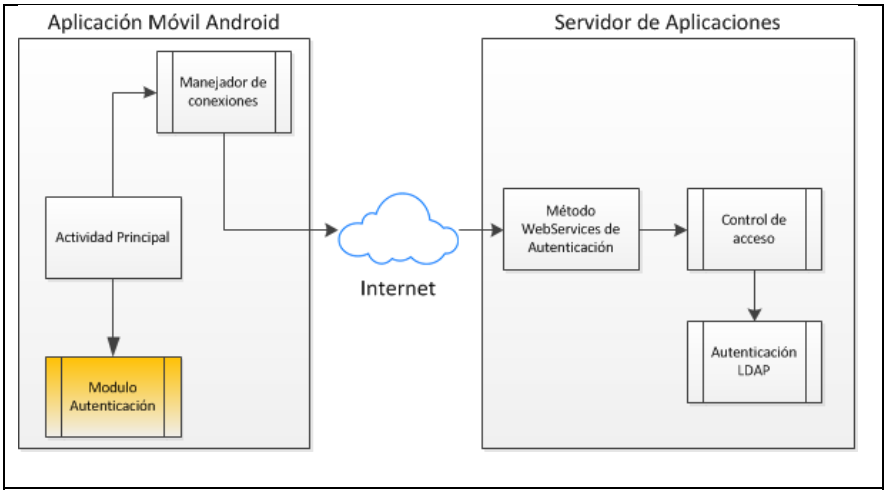


Figura 3.3. Arquitectura de modulo de autenticación

En la figura 3.1 también se representan las entidades de lado del servidor de aplicaciones, sin embargo, esto es referencial puesto que la implementación del Webservice que expone la autenticación sobre LDAP está fuera de alcance de este trabajo de investigación.

3.2.2 Características y versiones mínimas de servicio de directorio

El protocolo LDAP que debe utilizar el servicio de directorio es LDAP versión 3.

Como vemos en las figuras 3.4 y 3.5, Linux no solo es el sistema operativo más utilizado, sino que la proyección de crecimiento de su uso para los próximos años, es significativamente mayor en comparación a otros sistemas operativos.

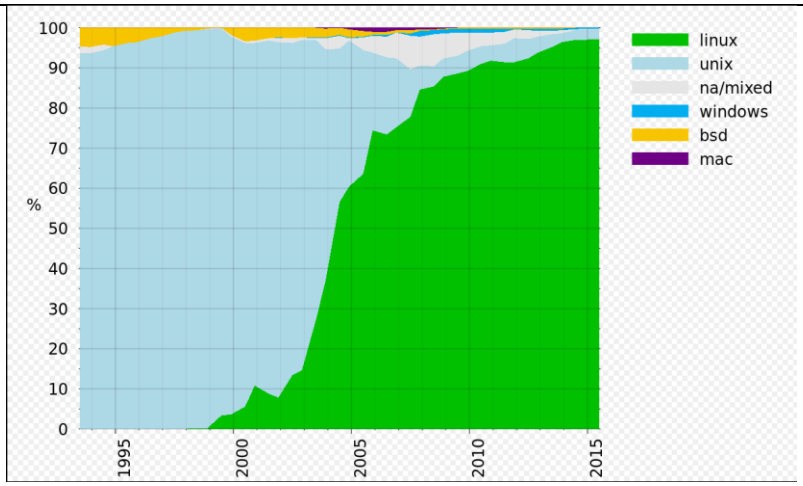


Figura 3.4. Top 500 Sistemas Operativos en supercomputadoras, en [20]

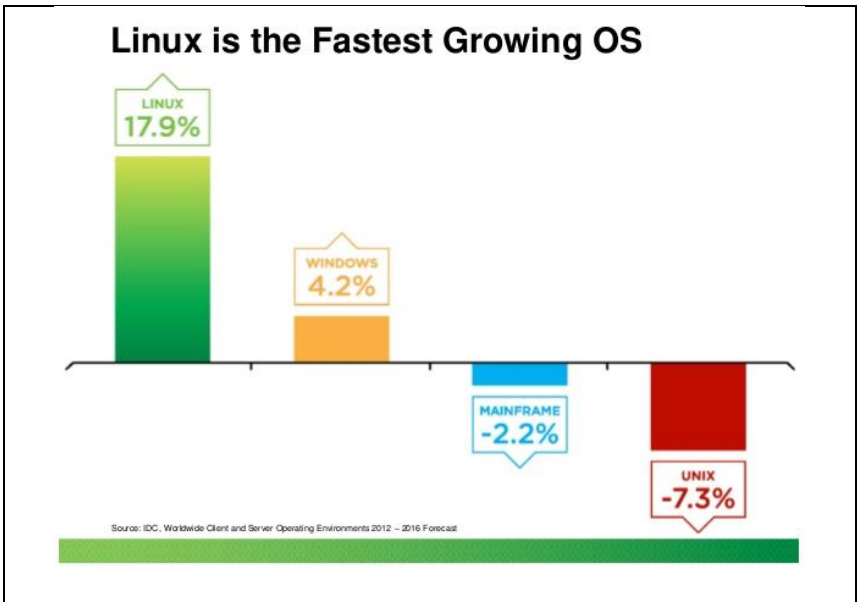


Figura 3.5. Top 500 Sistemas Operativos en supercomputadoras, en [21]

Por tal motivo, para la realización del presente proyecto, vamos a utilizar un ambiente Linux, y la implementación LDAPv3 que viene en todas sus distribuciones, OpenLDAP en la versión 2.4.x. y el servicio debe estar publicado al internet mediante WSDL 2.0.

3.2.3 Versión de sistema operativo móvil y compatibilidad

El módulo de autenticación debe ser desarrollado bajo el sistema operativo Android usando como versión mínima la 4.1.x, Jelly Bean. En la figura 3.6 se puede observar que las versiones más utilizadas son: Android Jelly Bean (4.1.x, 4.2.x, 4.3), Android KitKat (4.4) y Android Lollipop (5.x), por lo cual es conveniente utilizar como versión base Jelly Bean, es decir, la versión 4.1.x.

Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	3.8%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	3.4%
4.1.x	Jelly Bean	16	11.4%
4.2.x		17	14.5%
4.3		18	4.3%
4.4	KitKat	19	38.9%
5.0	Lollipop	21	15.6%
5.1		22	7.9%

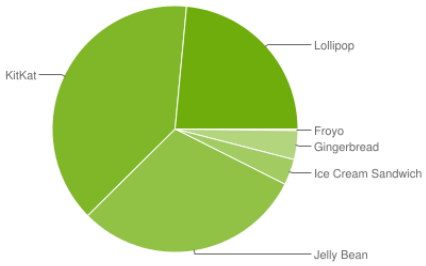


Figura 3.6. Versiones de Android más utilizadas, en [18]

Gracias a la capacidad de compatibilidad de Android, las versiones posteriores a la 4.1.x son, por defecto, compatibles y no debe existir problema alguno en utilizar el módulo de autenticación con versiones superiores como KitKat o Lollipop.

3.2.4 Definición de escenarios de autenticación

- Invocación correcta al módulo de autenticación con los atributos seguros del dispositivo móvil.
- Creación de mensaje de requerimiento de acuerdo al contrato WebServices definido.
- Descomposición del mensaje de respuesta de acuerdo al contrato WebServices definido.
- Validación de IMSI
- Validación de usuario y contraseña
- Validación de uso obligatorio de canal TSL/SSL

3.2.5 Definición de características técnicas de los dispositivos móviles

Los dispositivos móviles deben reunir las características mínimas requeridas para la instalación y correcta ejecución de las aplicaciones móviles junto al módulo de autenticación.

Para una alta capacidad de procesamiento y una alta eficiencia en la ejecución de algoritmos de encriptación, propios de los API de criptografía que ofrece Android, es necesario que se emplee solo dispositivos móviles con arquitectura de procesador de 64 bits.

Para un buen manejo de permisos de aplicación se recomienda tener las últimas actualizaciones del sistema operativo, tomando como base la versión Android Jelly Bean.

Los dispositivos móviles deben disponer de conectividad segura sobre Internet, es decir, deben tener soporte de SSL/TLS nativo y la validación de certificados de seguridad.

Los dispositivos móviles deben ser auténticos, confiables, originales de fábrica y sin ninguna alteración física o lógica, o manipulado por terceras personas que puedan poner en riesgo la integridad del dispositivo y de los sistemas a los que accede.

3.2.6 Niveles mínimos de seguridad de la aplicación

Con el fin de lograr un nivel de seguridad preventivo, el módulo de autenticación va a ser sometido a un análisis de código estático, en donde se compruebe que el desarrollo ha seguido las normas

básicas y las políticas de seguridad establecidas por LaMerced, para minimizar las vulnerabilidades.

3.2.7 Gestor de notificaciones de eventos en línea

Con el fin de obtener un nivel de seguridad reactivo, el módulo debe ser capaz de enviar una alerta al sistema para que sea posible notificar tanto al administrador como al usuario cuando se presente algún incidente de seguridad.

3.2.8 Consulta de eventos exitosos y fallidos

Con el fin de poder cumplir con la característica de seguridad de trazabilidad, el módulo debe permitir realizar auditorías sobre los eventos de autenticación realizados.

3.2.9 Definición de procedimientos y políticas

Dado que LaMerced gestiona la seguridad de la información con políticas y procedimientos establecidos, se debe seguir los lineamientos dictados por la misma, sugiriendo las modificaciones necesarias para el correcto funcionamiento y utilización del

módulo de autenticación y que no debieran ser omitidos en los procedimientos.

La política existente debe solicitar de manera explícita, en su procedimiento de registros de nuevos usuarios, los siguientes atributos:

- Usuario/ID: Código de identificación único para cada usuario, asignado por el administrador.
- Contraseña: Código secreto, conocido de manera exclusiva por el usuario y regido bajo las políticas de gestión de claves secretas de la organización.
- Dirección de correo electrónico
- Operadora de telefonía móvil
- Número celular
- Código IMSI: Código de identificación único para cada dispositivo móvil, integrado en la tarjeta SIM.
- Estado: Situación actual en la que se encuentra el ID del usuario (Activo, Bloqueado, Eliminado, etc) de acuerdo a las políticas para el control de revisión de los derechos de acceso de usuarios, implementado en la organización.

Los atributos mencionados, deberán estar actualizados y disponibles para ser usados en otros procedimientos.

3.2.9.1 Definición de procedimientos y políticas de acceso

La política de acceso debe solicitar de manera explícita, en su procedimiento de autenticación de usuarios, los siguientes atributos: Usuario, Contraseña.

La política de acceso debe solicitar de manera opcional, en su procedimiento de autenticación de usuarios, los siguientes atributos: Código IMSI

3.2.9.2 Definición de procedimientos y políticas para deshabilitar acceso e invalidación de la aplicación

El módulo debe estar sujeto al procedimiento para el Acceso Seguro al Sistema, implementado en la organización; por ejemplo, el número de intentos infructuosos permitidos, tiempo forzoso de espera antes de permitir nuevos intentos, entre otros.

3.2.9.3 Definición de procedimientos de seguridad cuando el acceso es vulnerado

El módulo debe estar sujeto al procedimiento de Gestión de Incidentes de Seguridad, implementado en la organización; comunicando de manera apropiada los eventos de seguridad y permitiendo identificar responsabilidades.

3.3 Requisitos de hardware y software

La implementación de la solución de autenticación hacia un servicio de directorio debe reunir los siguientes requerimientos mínimos de hardware y software.

Hardware

El módulo de autenticación debe emplear como único medio para el ingreso al servicio de directorio, un dispositivo móvil inteligente, es decir, un teléfono, tableta o similar dispositivo que ofrezca las características técnicas descritas en la Tabla 2.

La aplicación móvil podrá hacer uso de la infraestructura de red disponible para establecer una conexión a los sistemas de directorio o

transaccionales y demás servicios que se requiera. No se exige hardware adicional al existente.

Software

El módulo de autenticación deberá construirse, como podemos observar en la Tabla 2, usando la biblioteca de APIs que ofrece la plataforma de Android, Jelly Bean. No se deberá emplear librerías de fabricantes externos o de código libre para así evitar vulnerabilidades en código fuente de terceros. Las APIs que se utilicen deben estar actualizadas a la fecha de despliegue en producción.

El software empleado en los demás servicios a la que la aplicación móvil se conecta, deben estar actualizados en la versión más estable y con los parches de seguridad al día.

Cantidad	Descripción	Característica
1	Procesador	64 bits
1	Sistema Operativo	Android Jelly Bean 4.1.2 API Level 16 o superior, sin alteraciones de terceros.
1	Chip	Contar con acceso a internet

Tabla 2. Requerimientos mínimos de Hardware y Software

3.4 Estimación de costos

En el Ecuador, el desarrollo de aplicaciones móviles es muy limitado, según una investigación realizada por Metro Ecuador, en [22], algunos de los obstáculos para los desarrolladores en el país son: la competencia desleal, un freelance cobra una décima parte por el desarrollo de una aplicación móvil, los elevados costos de implementación, una aplicación para una empresa que utilice bases de datos o conexiones a sistemas externos está en el rango de USD$ 3000 a USD$ 12000, y las empresas locales no pueden invertir esa cantidad; y por último el mercado es reducido para una aplicación móvil dedicada.

Considerando la situación actual del mercado, se tiene los siguientes costos para la implementación de la solución móvil de autenticación para servicios de directorios:

Descripción	**Costos**
Desarrollo de módulo de autenticación en Android para dispositivos móviles	$ 3000
Soporte de instalación, configuración, pruebas y capacitación (20 horas)	$ 400
Contrato anual por mantenimiento y mejoras (40 horas)	$ 800
Total	**$ 4200**

Tabla 3. Costos de implementación

Estos valores están en el rango inferior del desarrollo de una aplicación dedicada para una organización que utiliza acceso a sistemas externos, en este caso, un servicio de directorio. Los costos por hora para el soporte y mantenimiento son de $20.

3.5 Ventajas

Actualmente en el mercado externo la empresa UnboundID ofrece un API para conexiones a LDAP, UnboundID LDAP SDK bajo licencia GNU General Public License version 2 y GNU Lesser General Public License versión 2.1, [23]. Esta es la alternativa más cercana a la investigación de este trabajo de tesis. Sin embargo hay que considerar las siguientes ventajas al momento de evaluar que alternativa seleccionar:

Descripción	**Módulo de autenticación**	**UnboundID LDAP SDK**
API de autenticación	Disponible mediante WebServices	Disponible mediante LDAP v3
Protección de acceso mediante cortafuegos	Disponible, a través de WebServices	No existe
Soporte y mantenimiento local	Disponible	No existe soporte local
Integración a	Disponible	Requiere desarrollo

aplicación móvil dedicada		
Tiempo de implementación	Sí, tiempo mínimo, sólo APIs básicos de Android	Sí, pero tiempos altos, se requiere aprender API de UnboundID LDAP SDK
Actualizaciones de versión	Sí, mediante soporte local	No, depende de la empresa UnboundID

Tabla 4. Ventajas de módulo de autenticación versus UnboundID LDAP SDK

Según la tabla 3 de ventajas de módulo de autenticación versus UnboundID LDAP SDK, es más conveniente la implementación de este trabajo de investigación, considerando los criterios descritos. Uno de los aspectos más relevantes es la seguridad de acceso, considerando que el servicio de directorio es un activo critico de una organización, no debe ser expuesto directamente hacia una red externa, su acceso debe ser controlado y restringido mediante reglas de seguridad de un corta fuegos. UnboundID LDAP SDK no ofrece esta característica y aumenta la probabilidad de un acceso no autorizado, a diferencia de la implementación del módulo de autenticación que si lo considera.

CAPÍTULO 4

ANÁLISIS Y DISEÑO DEL SISTEMA

Este capítulo centra su enfoque en determinar los aspectos más importantes que se deben considerar para identificar un usuario, desde una aplicación móvil a un servicio de directorio. En el capítulo anterior se definió la funcionalidad que debe reunir la solución propuesta y la capacidad de integración con distintas aplicaciones móviles considerando las restricciones y supuestos, tomando la descripción funcional como entrada a la etapa de análisis, se plantea diferentes puntos de vista, usando las más adecuadas herramientas de ingeniería de software, para analizar con detalle cada situación que debe responder el módulo de autenticación; y cada diagrama, modelo o representación tiene por objetivo abarcar completamente el alcance definido.

4.1 Casos de uso y escenarios

Tradicionalmente los casos de uso siempre han sido una herramienta idónea en la descripción de pasos o actividades que deben realizar los usuarios o actores para completar un proceso. Dentro del análisis, es justificada la identificación de casos de uso y sus respectivos escenarios.

Dentro del módulo de autenticación se levantan los siguientes casos de uso y sus especificaciones, así también los escenarios correspondientes:

Casos de Uso	**Escenarios**
Apertura de aplicación móvil	Apertura exitosa de aplicación móvil
	Apertura fallida de aplicación móvil por incompatibilidad de módulo de autenticación
	Apertura fallida de aplicación móvil por bloqueos de accesos especiales de recursos
	Apertura fallida de aplicación móvil por falta de recursos
Autenticación de usuario	Autenticación exitosa de usuario
	Autenticación fallida de usuario por datos de entrada incorrectos
	Autenticación fallida de usuario

	porque usuario no existe
	Autenticación fallida de usuario por que el dispositivo móvil no se encuentra registrado
	Autenticación fallida de usuario por bloqueos de accesos especiales de recursos
	Autenticación fallida de usuario por error de conexión a Internet
	Autenticación fallida de usuario por error de conexión a servidores externos de la organización
	Autenticación fallida de usuario por error en certificado de seguridad
	Autenticación fallida de usuario por error de hardware

Tabla 5. Casos de uso y escenarios

Según la descripción que se muestra en la Tabla 4, los casos de uso de análisis son Apertura de aplicación móvil y Autenticación de usuario.

Apertura de aplicación móvil, este caso de uso describe el proceso de inicialización de una aplicación móvil cualquiera que emplea el módulo de autenticación. Este escenario ocurre en el momento indicado en que

el usuario busca en su dispositivo móvil la aplicación brindada por la empresa LaMerced y la ejecuta para iniciar sus actividades, pero no realiza ninguna entrada de datos, es simplemente la apertura o la operación de abrir una aplicación móvil. En ese instante, la aplicación realiza algunas rutinas automáticas propias del proceso de arranque y del entorno del sistema operativo Android.

Autenticación de usuario, el caso de uso más relevante del proyecto, su lógica de instrucciones permite que los datos privados de las credenciales de un usuario se validen y transformen para ser enviados al servicio de directorio.

A continuación se describe al detalle los casos de uso y sus escenarios más importantes.

Caso de uso: Apertura de aplicación móvil

Nombre del caso de uso: Apertura de aplicación móvil	**ID:** 100
Módulo: Inicialización de aplicación y de módulos integrados	
Actor(es): Usuario final	
Interesados: Usuario final, Jefe de atención a clientes, Jefe de Sistemas y Seguridad Informática	
Descripción: Proceso que es invocado por una acción del usuario final y consiste en la inicialización de la aplicación móvil con sus módulos integrados.	
Pasos realizados:	**Información de los pasos:**
1.- Usuario localiza ícono de aplicación móvil	El usuario busca la aplicación móvil mediante su ícono representativo
2.- Usuario invoca a la aplicación móvil	La aplicación móvil es iniciada cuando el usuario la selecciona dentro del menú de aplicaciones.
Precondiciones: La aplicación debe estar instalada y actualizada correctamente.	
Postcondiciones: El sistema operativo asigna a la aplicación móvil los recursos suficientes para que se inicie.	
Suposiciones: El dispositivo móvil responde adecuadamente a los eventos del usuario.	
Garantía de éxito: La aplicación móvil se inicia correctamente	
Prioridad: Media	**Riesgo:** Medio

Escenario: Apertura fallida de aplicación móvil por bloqueos de accesos especiales de recursos

<table>
<tr><td colspan="2">Nombre del escenario: Apertura fallida de aplicación móvil por bloqueos de accesos especiales de recursos</td><td>ID: 100.1</td></tr>
<tr><td colspan="3">Módulo: Inicialización de aplicación y de módulos integrados</td></tr>
<tr><td colspan="3">Actor(es): Usuario final</td></tr>
<tr><td colspan="3">Interesados: Usuario final, Jefe de atención a clientes, Jefe de Sistemas y Seguridad Informática.</td></tr>
<tr><td colspan="3">Descripción: Proceso que es invocado por una acción del usuario final y consiste en la inicialización de la aplicación móvil con sus módulos integrados. El resultado de la ejecución del proceso es fallido debido a falta de permisos de acceso a recursos especiales como: Acceso a INTERNET y READ_PHONE_STATE</td></tr>
<tr><td>Pasos realizados:</td><td colspan="2">Información de los pasos:</td></tr>
<tr><td>1.- Usuario localiza ícono de aplicación móvil</td><td colspan="2">El usuario busca la aplicación móvil mediante su ícono representativo</td></tr>
<tr><td>2.- Usuario invoca a la aplicación móvil</td><td colspan="2">La aplicación móvil es iniciada cuando el usuario la selecciona dentro del menú de aplicaciones.</td></tr>
<tr><td colspan="3">Precondiciones: La aplicación debe estar instalada y actualizada correctamente.</td></tr>
<tr><td colspan="3">Postcondiciones: El sistema operativo no asigna a la aplicación móvil los recursos suficientes y accesos para que se inicie.</td></tr>
<tr><td colspan="3">Suposiciones: Instalación completa pero configuración de accesos incompleta o nula.</td></tr>
<tr><td colspan="3">Garantía de éxito: La aplicación móvil no se inicia</td></tr>
<tr><td>Prioridad: Media</td><td colspan="2">Riesgo: Medio</td></tr>
</table>

Caso de uso: Autenticación de usuario

<table>
<tr><td colspan="2">Nombre del caso de uso: Autenticación de usuario</td><td>ID: 200</td></tr>
<tr><td colspan="3">Módulo: Módulo de autenticación</td></tr>
<tr><td colspan="3">Actor(es): Usuario final</td></tr>
<tr><td colspan="3">Interesados: Usuario final, Jefe de atención a clientes, Jefe de Sistemas y Seguridad Informática</td></tr>
<tr><td colspan="3">Descripción: La aplicación móvil, a través, del módulo de autenticación permitirá a los usuarios verificar su identidad frente al servicio de directorio.</td></tr>
<tr><td colspan="2">Pasos realizados:</td><td>Información de los pasos:</td></tr>
<tr><td colspan="2">1.- Usuario ingresa los valores de entrada</td><td>Se ingresa los datos de usuario y clave. También en segundo plano se obtiene el valor del IMSI de la simcard del dispositivo móvil.</td></tr>
<tr><td colspan="2">2.- Usuario presiona el botón de login</td><td>La aplicación móvil envía los datos ingresados a los servidores externos mediante la conexión a la red establecida.</td></tr>
<tr><td colspan="3">Precondiciones: La aplicación debe estar iniciada correctamente. Dentro del perfil de los usuarios se debe adicionar previamente un nuevo atributo para almacenar la IMSI.</td></tr>
<tr><td colspan="3">Postcondiciones: El sistema de directorio externo verifica la identidad del usuario y la aplicación móvil permite el ingreso al sistema de acuerdo al rol predeterminado del usuario.</td></tr>
<tr><td colspan="3">Suposiciones: El dispositivo móvil tiene acceso a una red WIFI o Celular para que se pueda conectar a los servidores externos.</td></tr>
<tr><td colspan="3">Garantía de éxito: El usuario recibe una respuesta satisfactoria de ingreso al sistema.</td></tr>
<tr><td colspan="2">Prioridad: Media</td><td>Riesgo: Medio</td></tr>
</table>

Escenario: Autenticación fallida de usuario por datos de entrada incorrectos.

<table>
<tr><td colspan="2">Nombre del escenario: Autenticación fallida de usuario por datos de entrada incorrectos</td><td>ID: 200.1</td></tr>
<tr><td colspan="3">Módulo: Módulo de autenticación</td></tr>
<tr><td colspan="3">Actor(es): Usuario final</td></tr>
<tr><td colspan="3">Interesados: Usuario final, Jefe de atención a clientes, Jefe de Sistemas y Seguridad Informática</td></tr>
<tr><td colspan="3">Descripción: La aplicación móvil, a través, del módulo de autenticación permitirá a los usuarios verificar su identidad frente al servicio de directorio. El escenario es fallido debido al ingreso de datos incorrectos.</td></tr>
<tr><td>Pasos realizados:</td><td colspan="2">Información de los pasos:</td></tr>
<tr><td>1.- Usuario ingresa los valores de entrada</td><td colspan="2">Se ingresa los datos de usuario y clave. En segundo plano se obtiene el valor del IMSI de la simcard del dispositivo móvil.</td></tr>
<tr><td>2.- Usuario presiona el botón de login</td><td colspan="2">La aplicación móvil envía los datos ingresados a los servidores externos mediante la conexión a la red establecida.</td></tr>
<tr><td colspan="3">Precondiciones: La aplicación debe estar iniciada correctamente. Dentro del perfil de los usuarios se debe adicionar previamente un nuevo atributo para almacenar la IMSI.</td></tr>
<tr><td colspan="3">Postcondiciones: El módulo de autenticación no permite el ingreso de datos incorrectos en formato y en valor, ambos se validan mediante rutinas de programación o mediante el servicio de directorio.</td></tr>
<tr><td colspan="3">Suposiciones: El usuario no ingreso correctamente los datos de nombre de usuario y contraseña.</td></tr>
<tr><td colspan="3">Garantía de éxito: El usuario recibe una respuesta de error en el ingreso</td></tr>
</table>

de sus credenciales.	
Prioridad: Media	**Riesgo:** Medio

Escenario: Autenticación fallida de usuario por error de conexión a servidores externos de la organización.

<table>
<tr><td colspan="2">Nombre del escenario: Autenticación fallida de usuario por error de conexión a servidores externos de la organización</td><td>ID: 200.2</td></tr>
<tr><td colspan="3">Módulo: Módulo de autenticación</td></tr>
<tr><td colspan="3">Actor(es): Usuario final</td></tr>
<tr><td colspan="3">Interesados: Usuario final, Jefe de atención a clientes, Jefe de Sistemas y Seguridad Informática</td></tr>
<tr><td colspan="3">Descripción: La aplicación móvil, a través, del módulo de autenticación permitirá a los usuarios verificar su identidad frente al servicio de directorio. El escenario es fallido debido a problemas de conectividad con los servicios web para utilizar el servicio de directorio.</td></tr>
<tr><td>Pasos realizados:</td><td colspan="2">Información de los pasos:</td></tr>
<tr><td>1.- Usuario ingresa los valores de entrada</td><td colspan="2">Se ingresa los datos de usuario y clave. También en segundo plano se obtiene el valor del IMSI de la simcard del dispositivo móvil.</td></tr>
<tr><td>2.- Usuario presiona el botón de login</td><td colspan="2">La aplicación móvil envía los datos ingresados a los servidores externos mediante la conexión a la red establecida.</td></tr>
<tr><td colspan="3">Precondiciones: La aplicación debe estar iniciada correctamente. Dentro del perfil de los usuarios se debe adicionar previamente un nuevo atributo para almacenar la IMSI.</td></tr>
<tr><td colspan="3">Postcondiciones: El módulo de autenticación no puede validar la identidad del usuario debido a problemas de conectividad al servicio de directorio.</td></tr>
<tr><td colspan="3">Suposiciones: Los errores de conectividad pueden ser por: Acceso a</td></tr>
</table>

redes restringido, servicio de datos celular insuficiente, baja señal de red, etc.	
Garantía de éxito: El usuario recibe una respuesta de error en el ingreso de sus credenciales debido a problemas de conexión.	
Prioridad: Media	**Riesgo:** Medio

4.2 Modelo de procesos

Usando un modelo de procesos se describe el flujo de comunicación que tienen los datos desde que son ingresados por el usuario desde la interfaz de la aplicación hasta la validación contra el repositorio de datos del servicio de directorio.

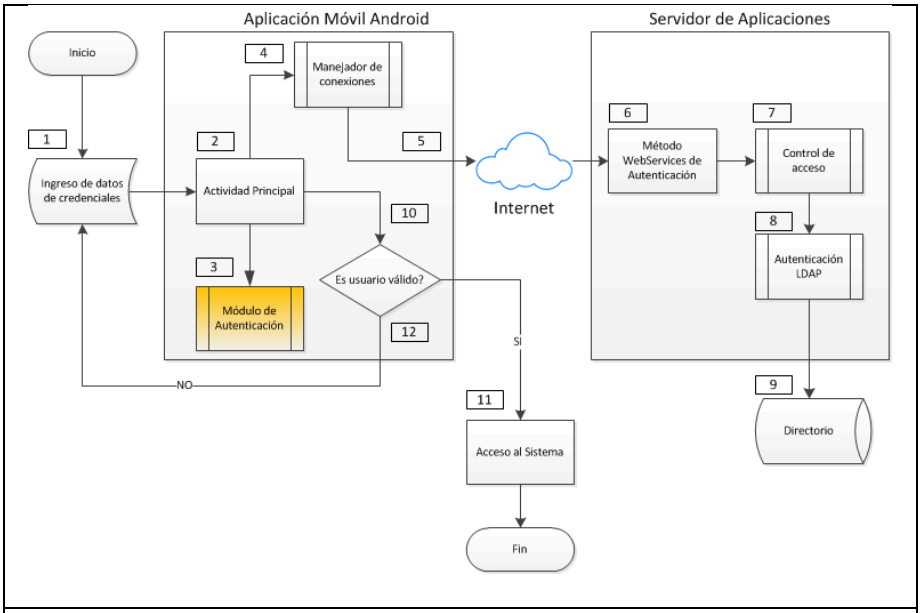


Figura 4.1. Modelo de procesos

En la figura 4.1 se observa la interacción entre la aplicación móvil, el módulo de autenticación, el servicio web y el servicio de directorio, todos comunicados y sincronizados para ofrecer una respuesta al usuario en línea. A continuación se describe los pasos del proceso:

- Ingreso de datos de credenciales, el usuario ingresa el nombre de usuario y la clave.
- La actividad principal recoge los valores de entrada y los envía al módulo de autenticación.
- El Módulo de Autenticación tiene como función principal validar los datos del usuario contra el Servicio de Directorio. El módulo crea el mensaje de requerimiento hacia el Manejador de Conexiones y también descompone e interpreta el resultado enviado por el Servicio de Directorio.
- El Manejador de Conexiones tiene como objetivo el transporte de mensajes en el protocolo http o https hacia servidores externos.
- Los mensajes son enviados por la red hacia los servidores externos.
- El WebServices recibe el requerimiento y direcciona la petición a las demás capas requeridas.
- Se aplican reglas de control de acceso
- El requerimiento se envía en el formato de consulta que interpreta el LDAP
- La consulta devuelve la información solicitada desde el repositorio de datos del Servicio de Directorio.
- El mensaje de respuesta del servicio de directorio es devuelvo a la Actividad Principal de la aplicación móvil. De acuerdo al

resultado se otorga acceso al sistema o se devuelve al usuario según el escenario.

- Si las credenciales son válidas se otorga acceso al sistema.
- Si las credenciales son inválidas se direcciona a la pantalla de inicio de sesión para que el usuario intente nuevamente.

4.3 Arquitectura de sistema

La arquitectura completa de la solución de autenticación comprende diferentes elementos de red, servidores y cortafuegos que correctamente conectados ofrecen un sistema seguro y confiable.

En la figura 4.2 se observa la arquitectura de referencia, es un modelo propuesto, tomando en cuenta la infraestructura de red actual de LaMerced, pero el mismo puede ser modificado acorde a las necesidades, recursos y nivel de protección que la organización espera. El acceso al sistema empieza desde el ingreso de credenciales de usuario desde el dispositivo móvil, sea este: teléfono inteligente, tableta o computador, pero la autenticación desde un computador esta fuera de alcance del presente proyecto.

Los datos del usuario son validados y transformados en un mensaje que puede ser enviado por Internet y que es interpretado por un WebServices. La seguridad del canal se protege mediante un cortafuego delante del WebServices, el cual recibe las peticiones desde Internet

pero solo resuelve las correctas y verificadas para el dominio de la organización. El WebServices se encuentra en la red DMZ, o red zona desmilitarizada.

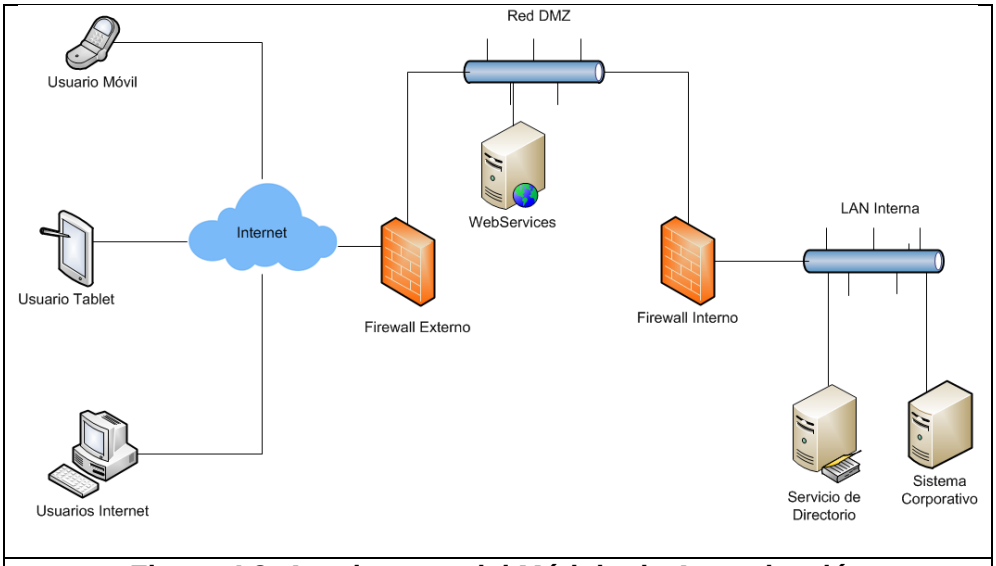


Figura 4.2. Arquitectura del Módulo de Autenticación

El Servicio de Directorio y el Sistema Corporativo residen en una red separa de la DMZ, esta red es la Red LAN Interna, de esta manera no se expone el Servicio de Directorio de la organización y se evita que sea atacado por algún intruso. Las peticiones validadas por el WebServices son atendidas por el Servicio de Directorio, y si el acceso es concedido, el usuario podrá realizar transacciones sobre el Sistema Corporativo.

4.4 Diagrama de interacción de objetos

El diagrama de interacción de objetos nos permite tener una representación visual de la comunicación entre las instancias de las clases. Este diagrama nos permite ver al detalle el ingreso de datos del usuario y como estos viajan entre clase y clase transformando esos datos en mensajes que se intercambian.

En la figura 4.3 se analiza el escenario de Autenticación Exitosa, se identifica las clases que intervienen en el flujo y los mensajes que intercambian. El ingreso de datos es capturado por la clase ActividadPrincipal, representación visual de la aplicación. Esta clase realiza validaciones de ingreso de datos en formato y contenido, luego de ejecutar sus rutinas, el mensaje sigue el flujo establecido comunicando a la clase ModuloAutenticacion, este es el núcleo del sistema, y tiene un conjunto de operaciones locales y llamadas a procesos remotos para conocer si el usuario tiene privilegios o no. Entre sus operaciones están:

- ValidarDatosEntrada, operación que permite ejecutar rutinas de validación de datos de entrada.
- recuperarImsi, operación que permite recuperar el IMSI de una simcard.
- crearMensaje, operación que reúne los datos del mensaje y crea el mensaje.
- transformarMensaje, operación que transforma un mensaje de acuerdo al formato requerido por el receptor.

- enviarRequerimiento, operación que envía el mensaje al receptor.

La clase ManejadorConexiones, es una clase tipo driver que direcciona los mensajes a un repositorio de datos, para este caso, la fuente de datos se accede mediante un WebServices. Posee las operaciones:

- validarCertificado, controla el emisor y receptor de certificado de seguridad.
- abrirConexion, permite abrir una conexión al WebServices externo.
- procesarRequerimiento, enviar y recupera respuesta del repositorio de datos.

La clase WebServiceLDAP permite resolver el requerimiento directamente en el repositorio de datos, en este caso, el servicio de directorio.

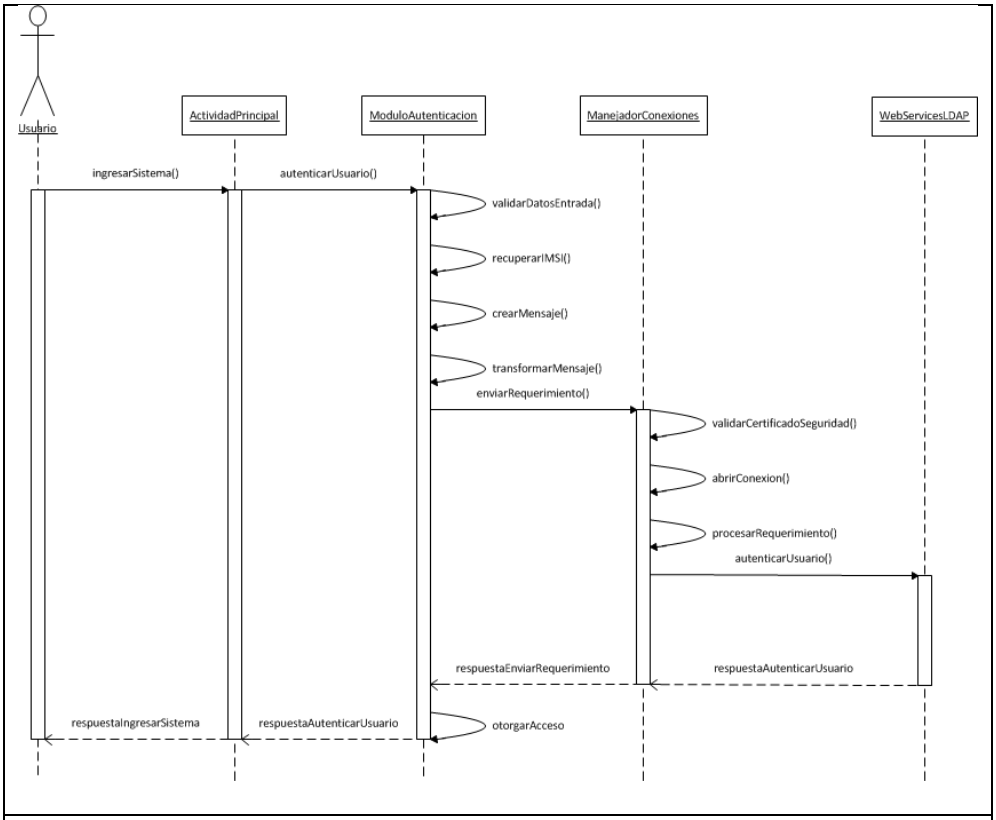


Figura 4.3. DIO de acceso exitoso

En la figura 4.4 se observa el mismo flujo de mensajes entre clases, pero representa el escenario de error cuando el usuario no es válido dentro del servicio de directorio. El módulo de autenticación es quien resuelve la respuesta de la clase anterior y si los datos de la aplicación móvil no cruzan con los datos del servicio de directorio no se otorga el acceso.

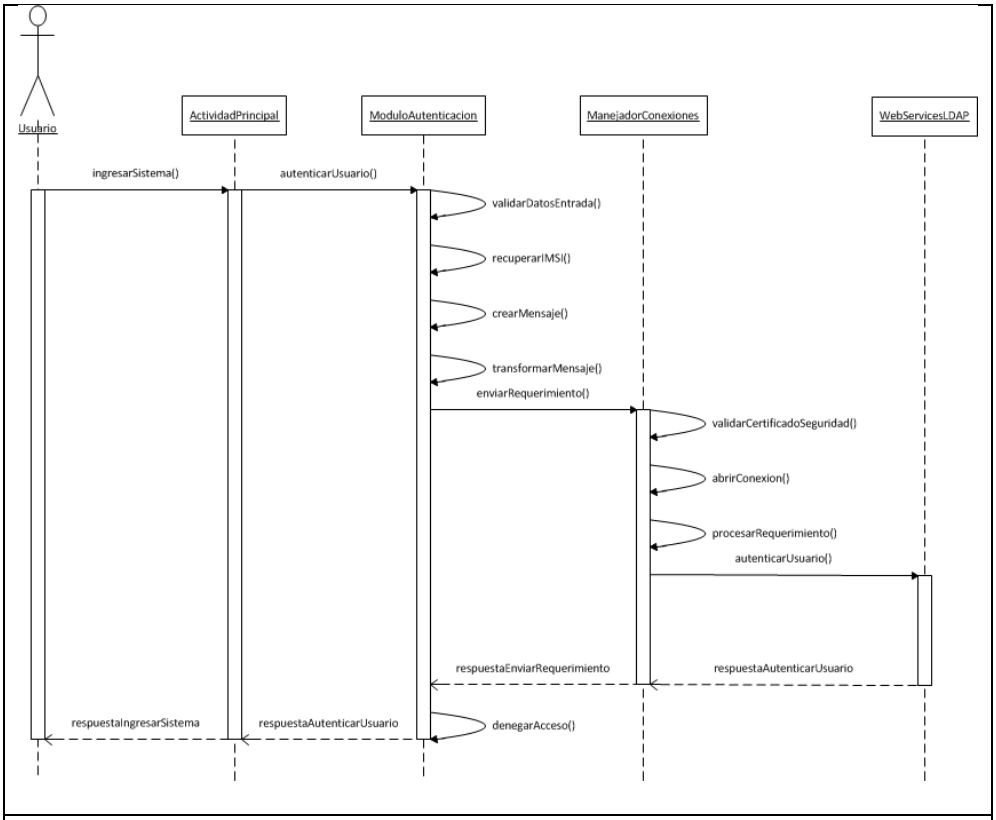


Figura 4.4. DIO de denegación de acceso

Otros diagramas son adjuntos al capítulo como error en conexión externa (Figura 4.5) y error en recuperación de IMSI (Figura 4.6).

Cuando la clase ManejadorConexiones no puede establecer una comunicación hacia servidores externos presenciamos un error de conexión externa, el mismo que detiene la autenticación indicando la causa del error.

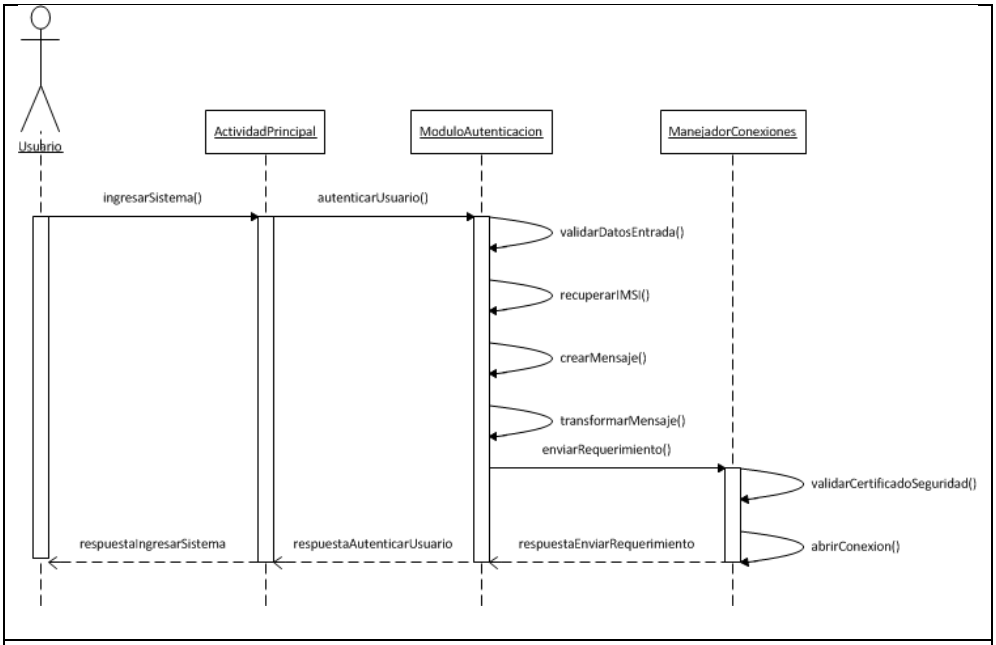


Figura 4.5. DIO de error en conexión externa

El módulo de autenticación tiene dentro de sus operaciones la recuperación del IMSI del dispositivo móvil conectado. Si no se efectúa la recuperación del IMSI se produce el siguiente flujo de mensajes (Figura 4.6) devolviendo el mensaje de error al cliente por inaccesibilidad del estado de la simcard y su información.

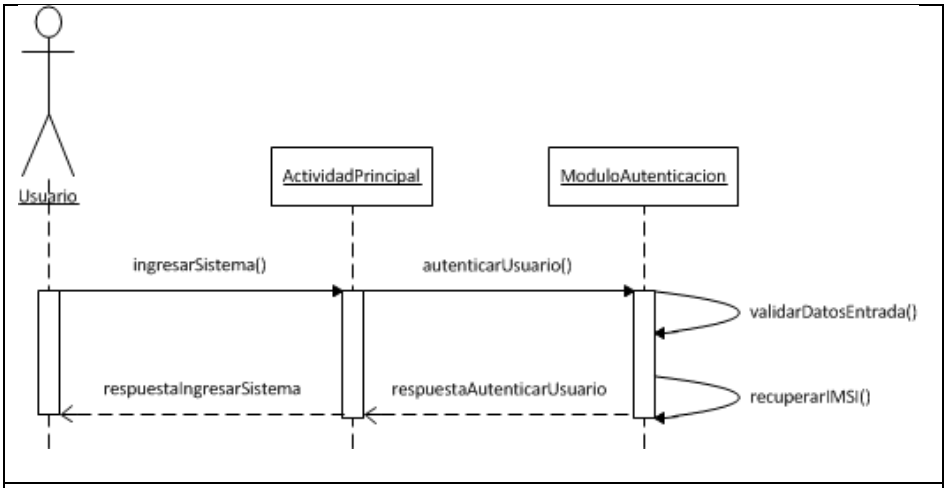


Figura 4.6. DIO de error en recuperación de IMSI

4.5 Diseño del plan de pruebas

El plan de pruebas permite identificar cómo se comporta el sistema ante diferentes escenarios, normalmente desde el punto de vista del usuario final. Es muy importante realizar un levantamiento de información respecto de los escenarios mínimos y requeridos que deben ejecutarse sobre el sistema y que deben garantizar el cumplimiento de los objetivos del mismo. Cada escenario detectado debe reunir un conjunto de atributos que direccionan el sentido de la prueba, esto es, datos de entrada, elementos del escenario, factores externos que afectan al escenario de prueba, resultados esperados, resultados obtenidos y condiciones y supuestos. No se puede omitir los criterios de aceptación que aprueban o rechazan el escenario de prueba planteado.

En el presente diseño del plan de pruebas, los elementos más importantes a considerar son:

- Objetivos: Se identifica los objetivos del conjunto de pruebas que se ejecutará.
- Participantes: Lista de participantes responsables de la ejecución y resultado de las pruebas, es importante que se indique el rol o función que cumplirá dentro de los escenarios de pruebas.
- Recursos: Lista de insumos que serán provistos para la ejecución de las pruebas.
- Sección, módulo o segmento: Define un conjunto de escenarios de pruebas agrupados por un propósito o factor en común. Por ejemplo: módulo de facturación, transacciones de crédito, reportes de inventario, ambiente de desarrollo, de producción, etc.
- Escenarios de prueba: Define conjunto de escenarios por cada caso de uso, con sus respectivos atributos, es decir, nombre y descripción, entradas, salidas, condiciones y supuestos, etc. No se debe contemplar planes de pruebas con exceso de escenarios que vuelve complejo su ejecución y control, es preferible planes de pruebas segmentados por objetivo, modulo o criterio de aceptación.
- Orden: Indica la secuencia o el orden en que debe ejecutarse los escenarios de pruebas.

- Criterios de aceptación: Define como interpretar los resultados de las pruebas y bajo que parámetros se acepta o rechaza el escenario.
- Conclusiones y recomendaciones: Sección que concluye el resultado del plan de pruebas y las recomendaciones a seguir para los próximos pasos.

CAPÍTULO 5

IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA

Siguiendo la metodología de desarrollo en cascada, luego de haber concluido la etapa de análisis y diseño del sistema se prosigue inmediatamente en la etapa de implementación del mismo, que consiste en materializar la solución planteada siguiendo los lineamientos definidos en los capítulos anteriores. Dentro de éste capítulo se exponen las herramientas de trabajo en el desarrollo de software tanto en la construcción como en la depuración y revisión del código seguro y al final se presenta un modelo de interfaz de usuario que puede servir como guía en la integración con el módulo de autenticación y los distintos escenarios de pruebas que se deben cumplir.

Las aplicaciones móviles que no implementen el módulo de autenticación, deberán regirse con el control de acceso autónomo para cada sistema, tal como lo han venido realizando. En la figura 5.1, se muestran dos sistemas distintos: el módulo de facturación, al cual se ingresa con una clave y un

usuario y el CRM, al cual se ingresa con la identificación y clave, donde "identificación" hace referencia al número de cédula del usuario.

Figura 5.1. Acceso a sistemas de LaMerced que no implementan el módulo de autenticación.

Es decir, para cada sistema, el usuario debe ingresar las credenciales correspondientes, como vemos en la figura 5.2.

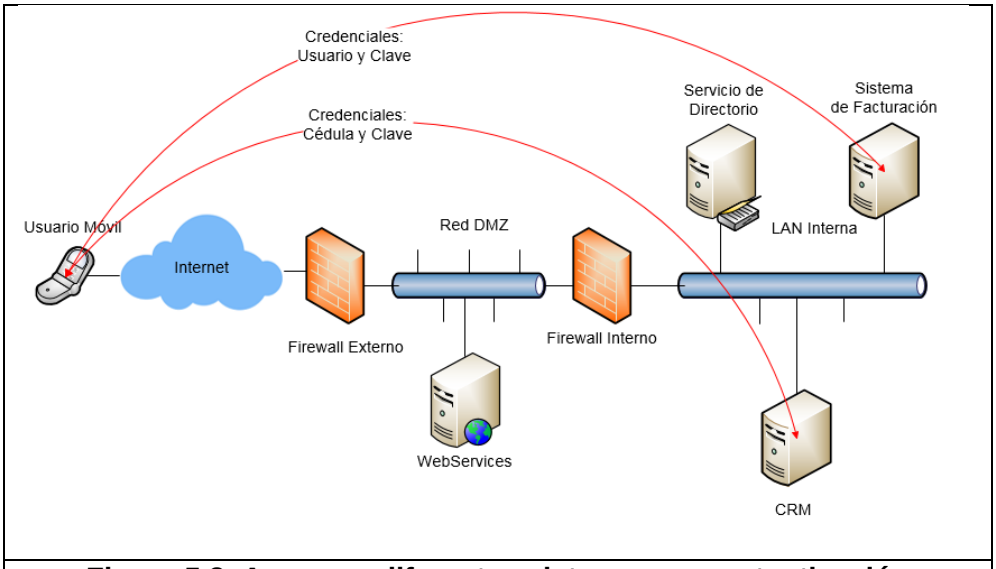


Figura 5.2. Acceso a diferentes sistemas con autenticación autónoma

Con la implementación del módulo, se utiliza un único medio de autenticación independientemente del sistema, ya que la gestión de credenciales se realiza de manera centralizada utilizando el servicio de directorio, como se muestra en la figura 5.3.

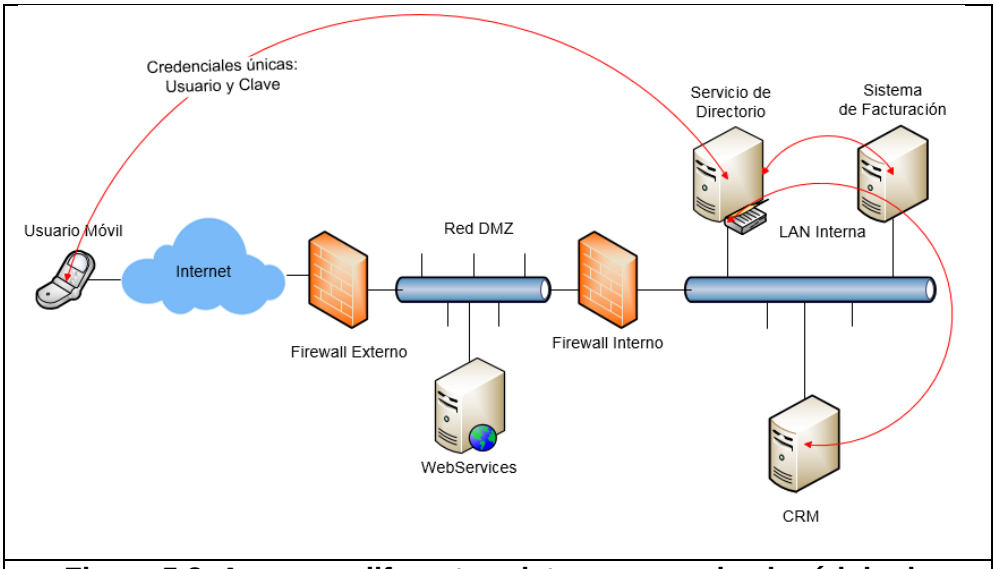


Figura 5.3. Acceso a diferentes sistemas usando el módulo de autenticación

5.1 Desarrollo de módulo de autenticación

Para la implementación del módulo de autenticación es necesario adecuar un ambiente de desarrollo estable y versátil que permita la codificación de una manera sencilla pero sin descuidar el control de errores y las advertencias en el mal uso del lenguaje de programación. Así, como IDE para el desarrollo del módulo de autenticación se ha seleccionado a Eclipse para desarrolladores Java versión Mars release 4.5.0, esta distribución es estable y puede ser descargada del siguiente link: http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/mars1, ver figura 5.4.

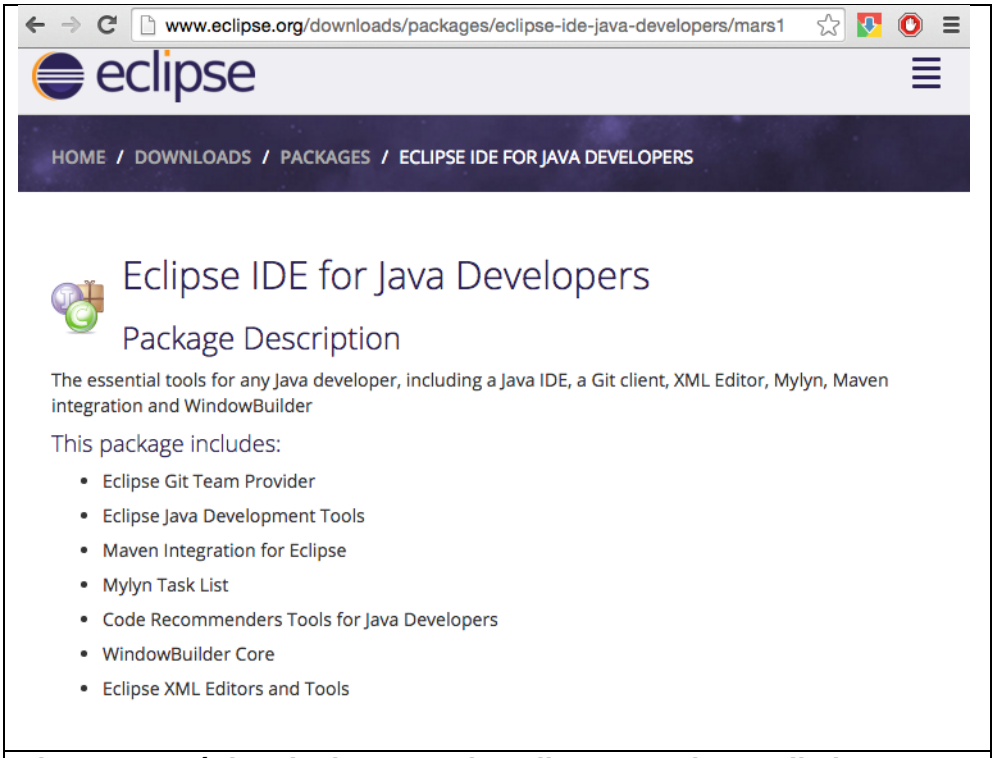


Figura 5.4. Página de descarga de Eclipse para desarrolladores Java

Como requisito de instalación se requiere instalar previamente el paquete de desarrollo en Java SE (Java Platform Stantard Edition), que instala un ambiente de desarrollo para Java y su respectiva Máquina Virtual, característica inseparable del lenguaje. Para una correcta interacción de Eclipse y Java, y posteriormente Android, se selecciona la versión mínima Java 8 con su última actualización.

La versión de Eclipse Mars no incluye el marco de trabajo para desarrollar aplicaciones en Android, ADT (Android Developer Tools), para adicionarle esta característica es necesario instalar un plugin o funcionalidad que permita tal capacidad. El sitio Web de Android para desarrolladores de aplicaciones móviles considera que para trabajar en

Eclipse se utilice el siguiente link de descarga: https://dl-ssl.google.com/android/eclipse/ que agrega la capacidad de desarrollar y emular aplicaciones móviles en Android en cualquiera de las versiones del sistema operativo que haya disponible. En la figura 5.2 se observa la pantalla de configuración de Eclipse para la instalación de ADT.

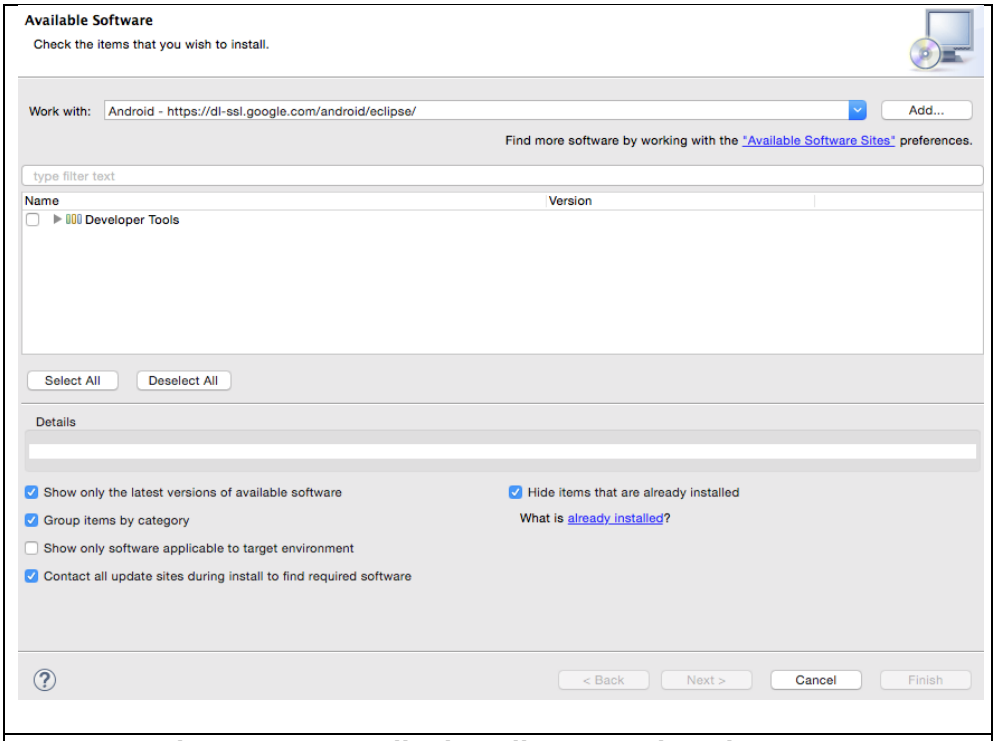


Figura 5.5. Pantalla de Eclipse para instalar ADT.

Dentro del análisis realizado en este trabajo de investigación se seleccionó como plataforma de desarrollo mínima Android 4.1.2 Jelly Bean (API Level 16), se requiere descargar ésta versión usando el componente Android SDK Manager instalado con el plugin. En la figura

5.6 se puede observar la pantalla de descarga de Android 4.1.2 y además otras versiones si fuera necesario.

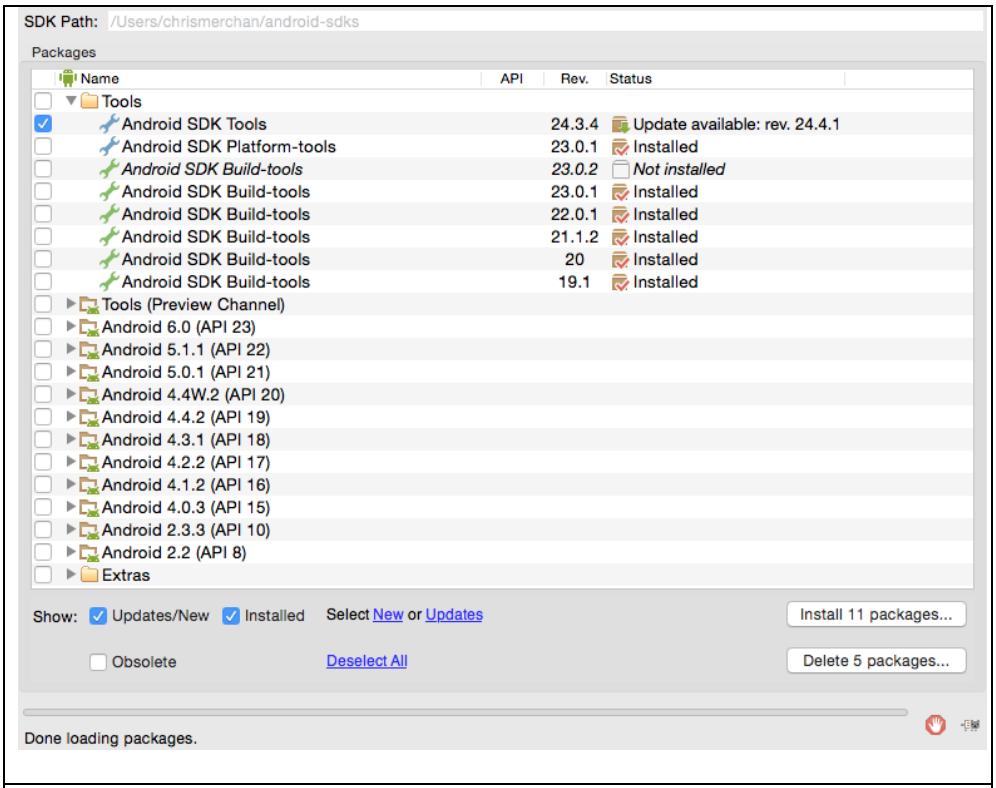


Figura 5.6. Android SDK Manager para instalación de plataformas.

En la siguiente tabla (Tabla 6) se resume las herramientas de desarrollo seleccionadas, la versión y su propósito.

Software	Versión Mínima	Propósito
Java SE	Java 8 update 60 o superior	Lenguaje de programación Java usado en la construcción de aplicaciones móviles Android
Eclipse IDE	Mars release 4.5.0	IDE para codificar en lenguaje de programación Java
Android Platform	Jelly Bean 4.1.2 API Level 16	Versión mínima de Android para el desarrollo del módulo de autenticación.

Tabla 6. Resumen de herramientas de software para desarrollo

Android tiene dos modalidades de desarrollo, Android Application Project o Android Library Project, la primera permite construir una aplicación Android completa que puede incluir código fuente, recursos gráficos o de multimedia, recursos de textos y recursos de permisos, configuración de aplicación y otros archivos que conforman la aplicación, todo esto es empaquetado en un archivo ejecutable de extensión .apk. Un Android Library Project permite construir una aplicación Android sin interfaz pero con código fuente o recursos que serán compartidos hacia la aplicación móvil que los utilice, esta aplicación, a diferencia de la primera, se empaqueta en un archivo de extensión .jar. El módulo de autenticación se construye bajo la modalidad Android Library Project.

En la Figura 5.7 se representa el modelo de clases Java que compone el módulo de autenticación y la interacción que realiza con las clases básicas del marco de trabajo de Android y de cualquier aplicación móvil que requiere autenticación de usuario. Las funciones que cumple cada clase y forman parte del módulo de autenticación son:

- CredencialSeguridad, clase que permite estructurar la información del usuario mediante los atributos de usuario, clave e imsi.
- GestorNotificaciones, clase que permite estructurar las notificaciones de eventos de éxito o error que genera el usuario. Los atributos son: código de mensaje, mensaje de evento y categoría que puede tomar los valores de Información, Advertencia o Crítico.
- ModuloAutenticacion, clase principal y núcleo del módulo de autenticación, tiene como principales operaciones: realizar la validación de los datos del usuario, crear la estructura del mensaje, transformar el mensaje para su transporte, notificar eventos, y otorgar o denegar acceso al usuario.

Las clases que se integran al módulo de autenticación son:

- MainActivity, clase que hereda de la súper clase Activity y que es responsable de la creación de la interfaz de autenticación y el manejo de eventos de usuario.
- ManejadorConexiones, clase que tiene por objetivo enviar un requerimiento a un Servicio Web, por convención el protocolo que se prefiere es SOAP.

Y finalmente las clases que se emplean del marco de trabajo de Android son:

- Activity, clase que interactúa con el usuario en la presentación gráfica y el control de eventos.
- Properties, clase que proporciona estructuras dinámicas para almacenamiento de parámetros de texto, se emplea para comunicar los parámetros de la cabecera SOAP.
- AsyncTask, clase asincrónica que levanta un hilo de ejecución en paralelo al principal para realizar peticiones http o https a un servidor.

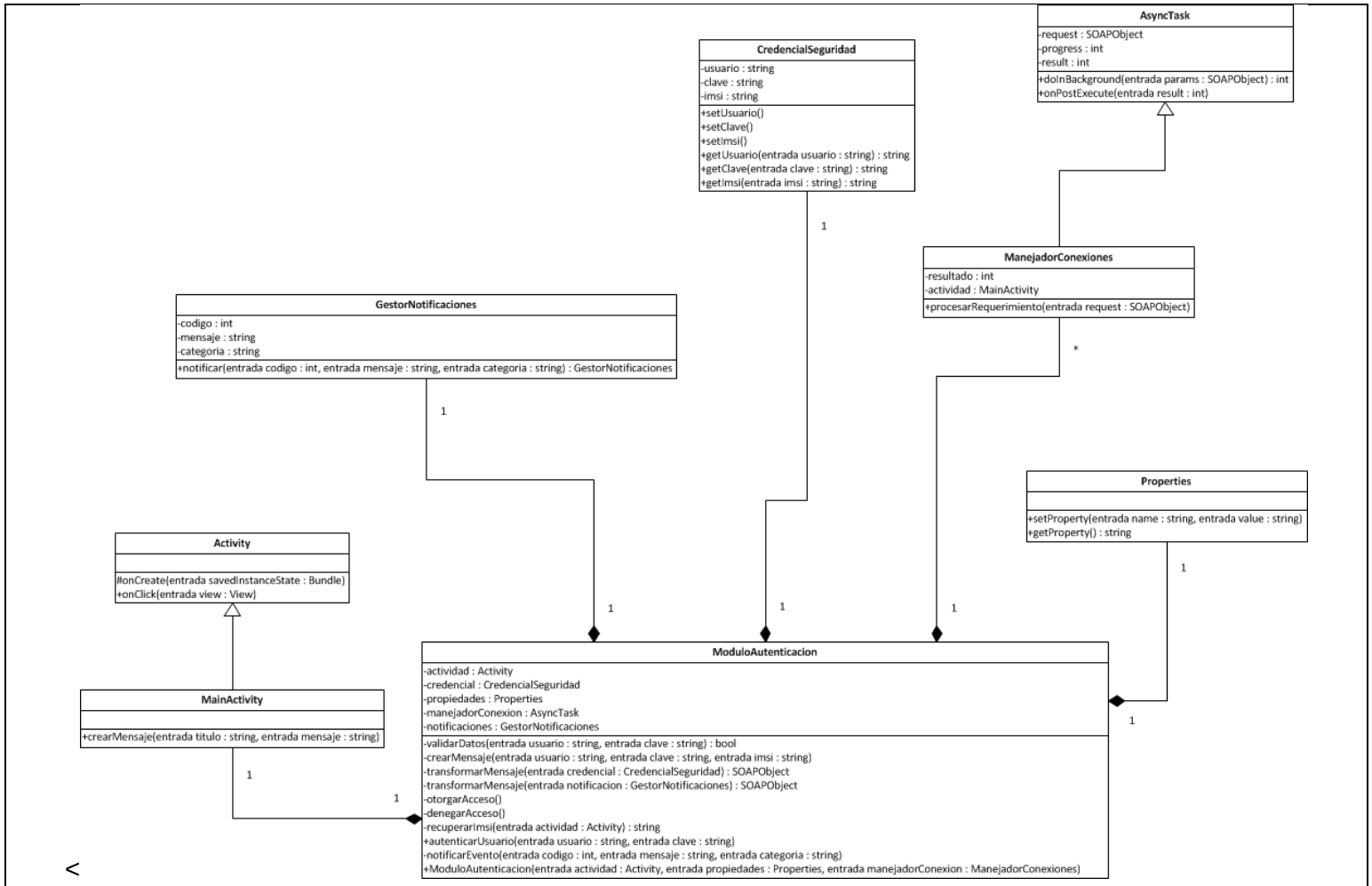


Figura 5.7. Modelo de clases Java

Cambios en el Servicio de Directorio

Como se describió en las secciones anteriores, el módulo requiere que el departamento de sistemas de LaMerced agregue nuevos atributos a su servicio de directorio, en la Figura 5.8 podemos la configuración del LDAP de la organización antes de modificar su configuración.

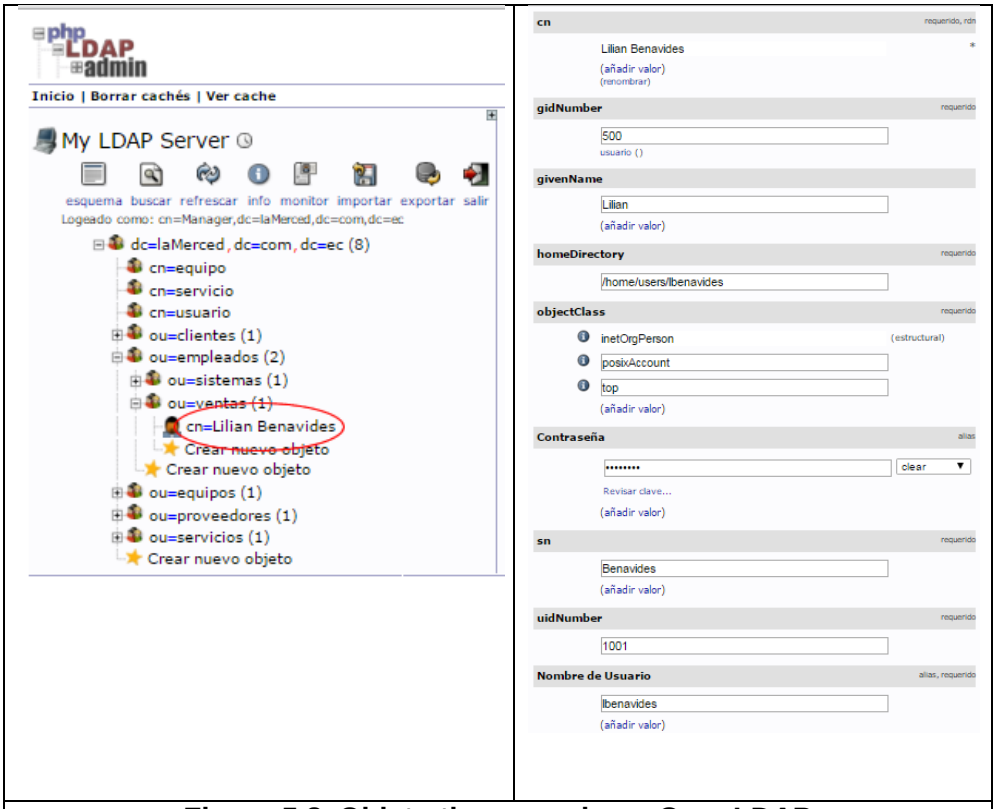


Figura 5.8. Objeto tipo usuario en OpenLDAP

Para añadir nuevos atributos en OpenLDAP, se ha creado una nueva clase. El esquema del Servicio de Directorios ha sido modificado para incluir esta nueva clase, tal como se muestra en la Figura 5.9.

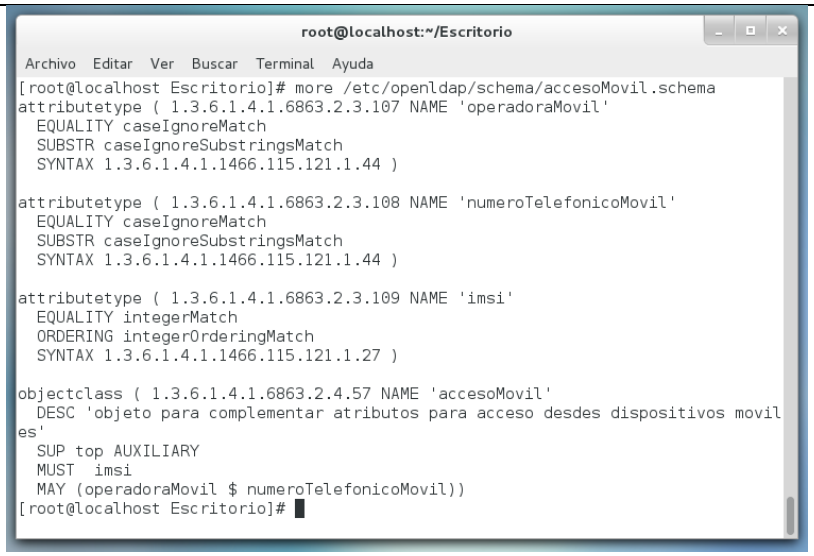


Figura 5.9. Configuración de clase con nuevos atributos en OpenLDAP

Una vez actualizado el esquema, se agrega la clase al objeto usuario que requiera los nuevos atributos, tal como se muestra en la Figura 5.10.

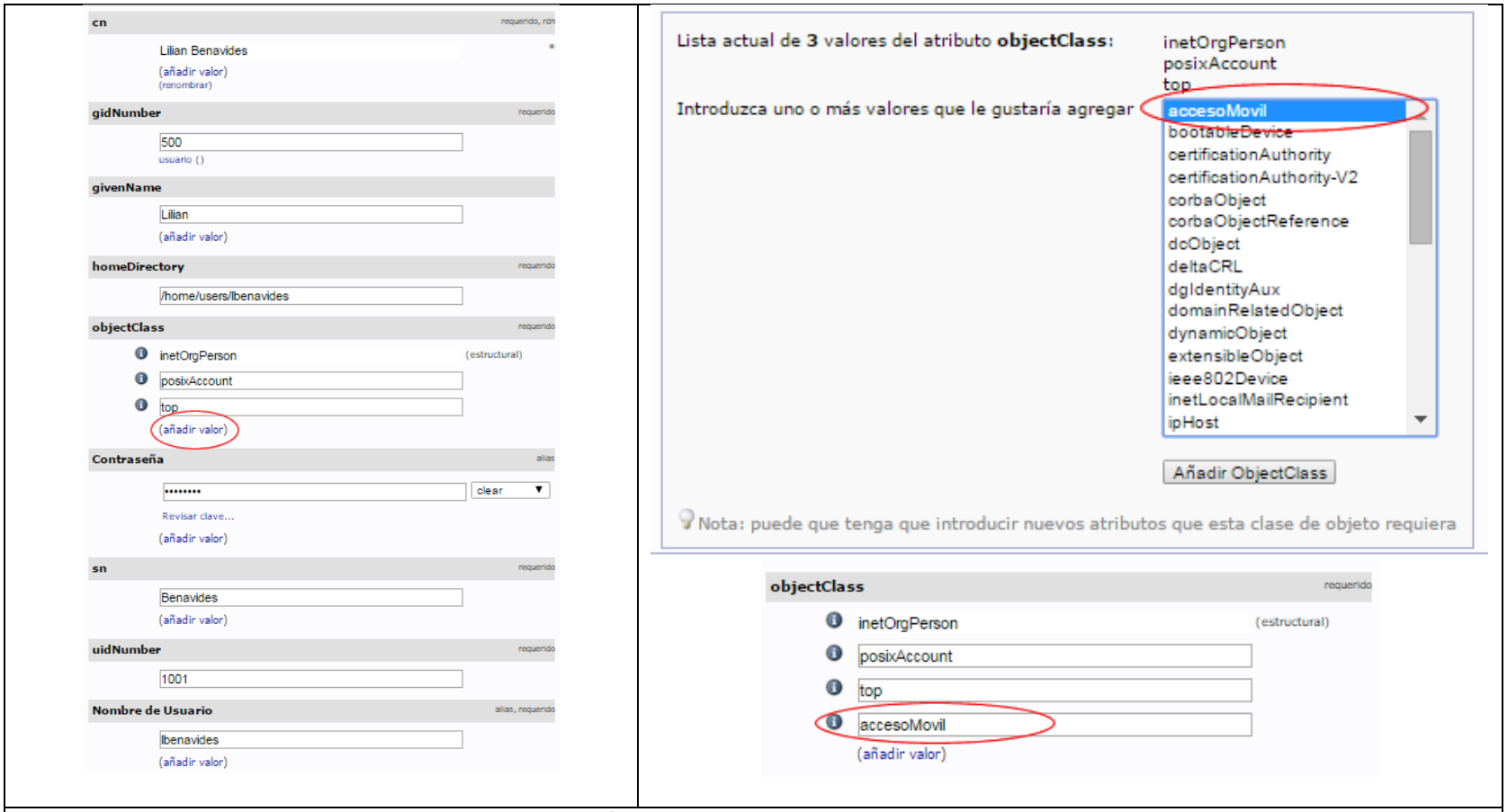


Figura 5.10. Configuración de la nueva clase a un usuario en OpenLDAP

Para finalizar, se añade el atributo al usuario que requiera la autenticación desde un dispositivo móvil y se le configura con los datos correspondientes a dicho equipo, tal como se muestra en la Figura 5.11.

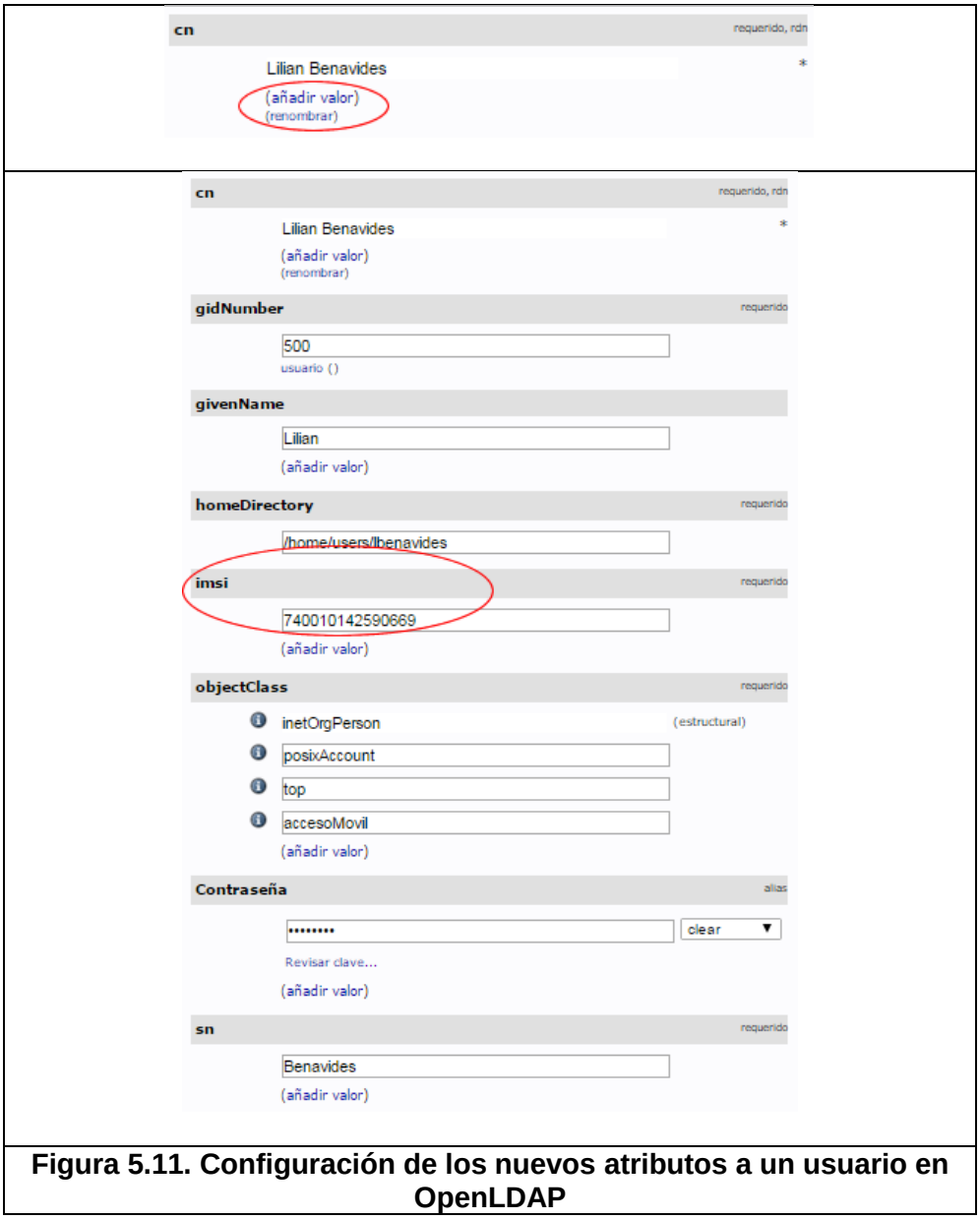


Figura 5.11. Configuración de los nuevos atributos a un usuario en OpenLDAP

Cambios en el WebService

Para evitar publicar el servicio de directorio, se hace uso del WebService de LaMerced, con el fin de tener acceso de manera segura a las credenciales de los usuarios, para esto, es necesario que en el WSDL se añadan nuevas operaciones correspondientes a la autenticación y registro de eventos, tal como se muestra en la Figura 5.12 y 5.13.

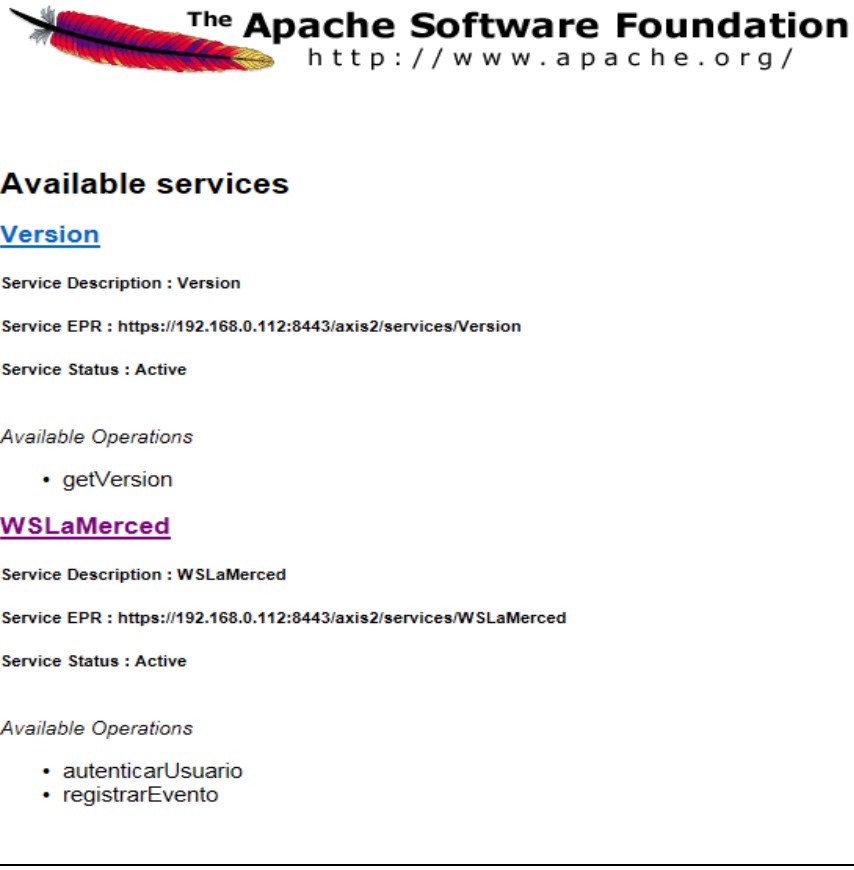
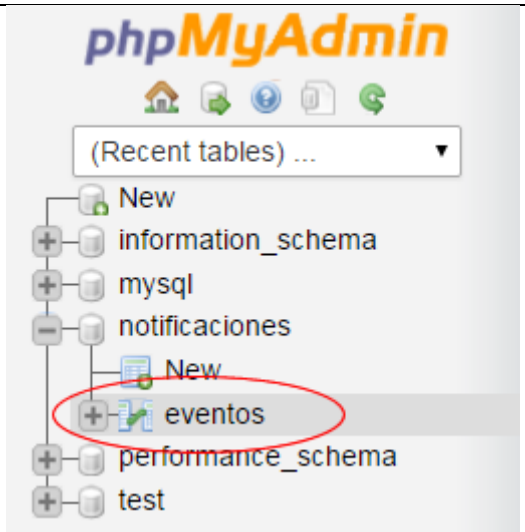


Figura 5.12. WebServices LaMerced

```
<?xml version="1.0" encoding="UTF-8"?>
- <wsdl:definitions targetNamespace="http://ec.com.lamerced/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:ns1="http://org.apache.axis2/xsd"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:tns="http://ec.com.lamerced/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:ns="http://ec.com.lamerced/xsd" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:documentation>WSLaMerced</wsdl:documentation>
  - <wsdl:types>
    - <xs:schema targetNamespace="http://ec.com.lamerced/xsd" elementFormDefault="qualified" attributeFormDefault="qualified">
      - <xs:element name="autenticarUsuario">
        - <xs:complexType>
          - <xs:sequence>
              <xs:element name="usuario" type="xs:string" nillable="true" minOccurs="0"/>
              <xs:element name="clave" type="xs:string" nillable="true" minOccurs="0"/>
              <xs:element name="imsi" type="xs:string" nillable="true" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      - <xs:element name="autenticarUsuarioResponse">
        - <xs:complexType>
          - <xs:sequence>
              <xs:element name="return" type="xs:int"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      - <xs:element name="registrarEvento">
        - <xs:complexType>
          - <xs:sequence>
              <xs:element name="codigo" type="xs:int"/>
              <xs:element name="descripcion" type="xs:string" nillable="true" minOccurs="0"/>
              <xs:element name="categoria" type="xs:string" nillable="true" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      - <xs:element name="registrarEventoResponse">
        - <xs:complexType>
          - <xs:sequence>
              <xs:element name="return" type="xs:int"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:schema>
    </wsdl:types>
```


Figura 5.13. Definición de las nuevas operaciones en el WebService

Para lograr la trazabilidad de los eventos que se presenten durante la utilización del módulo de autenticación, se ha creado un nuevo esquema en la base de datos existente, como se muestra en la figura 5.14.



+ Opciones

	id	codigo	descripcion	categoria	fecha
Editar Copiar Borrar	46	0	Autenticacion exitosa!	INFORMACION	2015-11-23
Editar Copiar Borrar	45	500	No se ha definido el uso del protocolo TLS/SSL	CRITICO	2015-11-23
Editar Copiar Borrar	42	0	Autenticacion exitosa!	INFORMACION	2015-11-13
Editar Copiar Borrar	39	0	Autenticacion exitosa!	INFORMACION	2015-11-11
Editar Copiar Borrar	38	0	Autenticacion exitosa!	INFORMACION	2015-11-11
Editar Copiar Borrar	37	500	Usuario o clave incorrecta	CRITICO	2015-11-10

Figura 5.14. Registro de eventos generados desde el módulo de autenticación.

5.1.1 Control de errores

Los errores del módulo de autenticación se controlan de acuerdo a la siguiente clasificación: error de usuario, error de aplicativo, error de sistema.

Los errores de usuario ocurren cuando se ingresa incorrectamente los valores de autenticación, esto es, el usuario y la clave de acceso. Existe el valor del Imsi, que no se ingresa pero que se captura de forma implícita de las propiedades de la simcard. En todos los escenarios se envía una notificación de evento al Servicio Web configurado en la aplicación, categorizado como Informativo o Advertencia y con códigos de error en el intervalo de 0 a 500.

Los errores de aplicativo no son generados de forma directa por el usuario, en su lugar son disparados por la aplicación ante escenario de falla. Algunas causas pueden ser: Configuración incorrecta de las propiedades para armar la cabecera SOAP que

se envía al Servicio Web, falta de permisos para acceder a Internet o al estado del teléfono, esto puede ocurrir si no se ha instalado correctamente la aplicación. Otro escenario es posible cuando el módulo de autenticación se emplea en versiones por debajo de Jelly Bean API Level 16, debido a que no soporta versiones inferiores a la 4.1.2. Ante estos escenarios se envía una notificación del evento al Servicio Web configurado en la aplicación, categorizado como Crítico y con códigos de error mayores a 500. Si no existe configurado el servidor de eventos no será posible reportar los escenarios de error.

Los errores de sistema pueden ocurrir cuando el sistema operativo Android está funcionando de manera inestable y produce un comportamiento inusual. Por ejemplo, conectividad nula a Internet o envío y recepción incompleta de los mensajes al Servicio Web, información de la simcard inaccesible por causas externas, falta de memoria para completar el requerimiento de la aplicación, etc. Ante estos escenarios se notifica mediante el Gestor de Notificaciones siempre y cuando sea posible, categorizado como Crítico y con códigos de error mayores a 500.

5.1.2 Gestor de notificaciones y reporte de acceso

Todas las notificaciones que se generen pueden ser enviadas a un servidor centralizado para su posterior análisis y revisión. Se debe realizar la configuración de los parámetros correctos como son: NAMESPACE_EVENT y METHOD_NAME_EVENT para que el módulo identifique la operación y pueda enviar los eventos. Los eventos deben tener la siguiente estructura:

- Código del evento, valor entero mayor a cero que indica un ID para el evento, por convención de 0 a 300 es Informativo, de 300 a 500 es Advertencia y mayores a 500 es Crítico.
- Mensaje del evento, valor que contiene el mensaje del evento a reportar.
- Categoría, valor que indica la clasificación del evento, puede tomar los valores de Información, Advertencia o Crítico.

Los eventos éxitos o de acceso también pueden ser reportados tomando en consideración los lineamientos descritos.

5.1.3 Interfaz de usuario

El módulo de autenticación no posee una interfaz de usuario, pues es una librería que se integra a una aplicación móvil existente, sin embargo existen requerimientos mínimos que debe reunir la aplicación móvil considerando la información que necesita el módulo de autenticación para su funcionamiento.

En la Figura 5.15 se muestra un ejemplo básico de una aplicación móvil que implementa autenticación con usuario y clave.

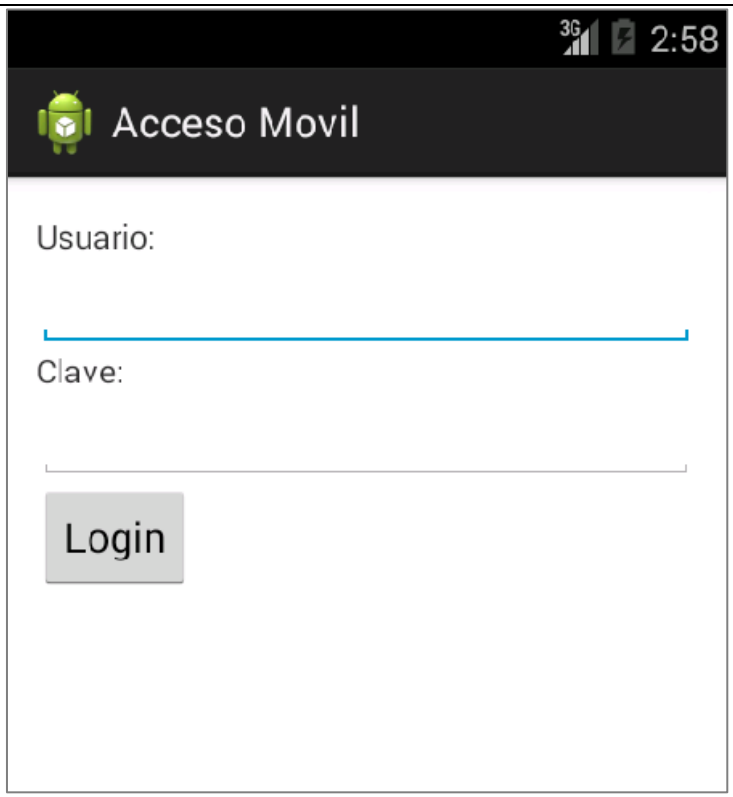


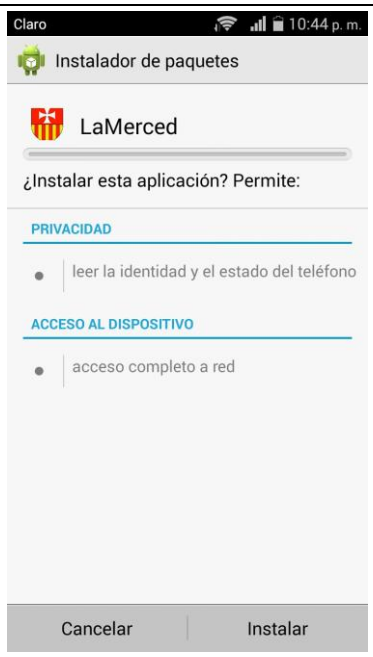
Figura 5.15. Interfaz de autenticación de usuario básica

Los campos deben ser tipo texto con la particularidad que el campo clave debe ser tipo PASSWORD para evitar mostrar el texto a terceras personas. El ingreso de los valores debe ser controlado mediante algún evento, como por ejemplo, al presionar el botón de Login o cuando se indique explícitamente la opción “Ir” del teclado.

Los mensajes de validación de datos se pueden mostrar usando un mensaje tradicional como AlertDialog o un mensaje más discreto usando Toast; o cualquier otro medio de preferencia, siempre que se muestre con claridad el mensaje al usuario.

Para efectos demostrativos del presente trabajo de tesis, se ha implementado una aplicación Android que utiliza el módulo de autenticación.

En la figura 5.16, se muestra el proceso de instalación, que incluye el listado de permisos que el app tendrá sobre el dispositivo.



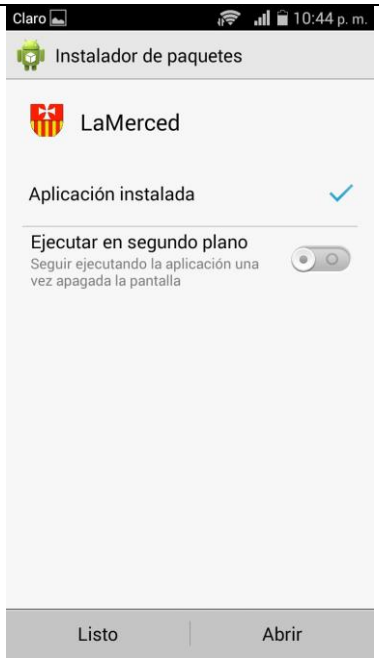


Figura 5.16. Instalación de Aplicación Android que implementa el módulo de autenticación.

Una vez instalada nuestra aplicación, se puede proceder a su apertura desde el icono creado en el dispositivo, como se muestra en la figura 5.17.

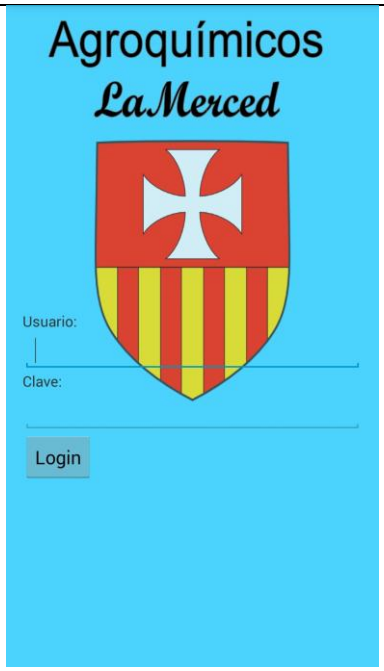
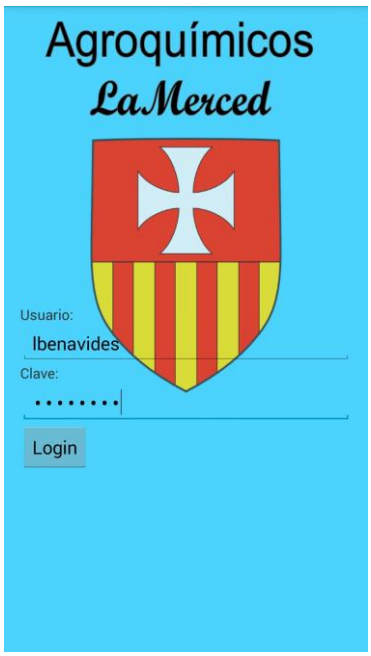


Figura 5.17. Apertura de Aplicación Android que implementa el módulo de autenticación.

En la figura 5.18, se puede observar el menú desplegable, que permite realizar la consulta del IMSI del dispositivo.

Figura 5.18. Consulta de IMSI usando Aplicación Android que implementa el módulo de autenticación.

Para realizar la autenticación, se debe ingresar el usuario y la contraseña; el IMSI es capturado internamente y enviado al LDAP para su validación junto con los otros datos.



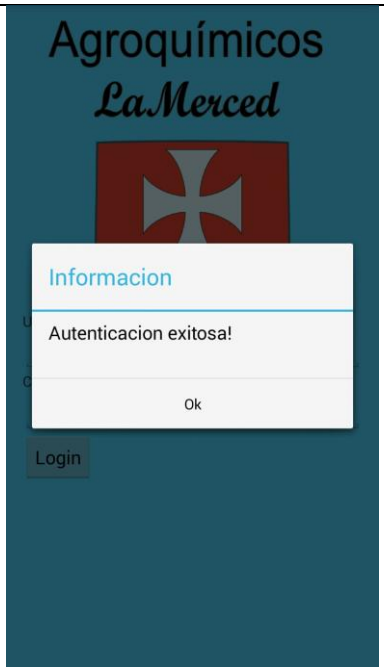


Figura 5.19. Login exitoso desde Aplicación Android que implementa el módulo de autenticación.

5.2 Escenarios de pruebas

Como estrategia en la ejecución de las pruebas se ha seleccionado los siguientes modelos de pruebas provistos en función de las validaciones del sistema, de los requerimientos funcionales, requerimientos de seguridad y de los usuarios.

Plan de pruebas con los requerimientos

La siguiente matriz de pruebas se plantea de acuerdo a los requerimientos funcionales definidos en el Capítulo 3, sección 3.1.1 Alcance y límites del sistema.

Requerimientos	¿Cumplió con el requerimiento?		Observaciones
	Si	No	
El módulo de autenticación se adapta a un modelo de autenticación centralizada usando un servicio de directorio que está disponible en la organización.			
El módulo de autenticación usa los atributos existentes del servicio de directorio de la organización identificados para la autenticación del usuario.			
El módulo de autenticación utiliza las opciones de configuración existentes del servicio de directorio para			

definir nuevos atributos válidos para la autenticación del usuario o de un dispositivo autorizado.			
El módulo de autenticación utiliza Servicios Web definidos por la organización para el acceso al servicio de directorio.			
El módulo de autenticación utiliza el canal seguro o encriptado SSL para establecer comunicación con el Servicio Web.			
El módulo de autenticación envía e interpreta los mensajes de los Servicios Web de la organización para la interacción con el servicio de directorio.			
El módulo de autenticación esta implementado en Android.			

El módulo de autenticación es modular e integrable a una aplicación Android.			
El módulo de autenticación valida las credenciales de un usuario definidas en la organización.			
El módulo de autenticación valida atributos definidos en la organización y asociados a los dispositivos móviles que posee un usuario.			
El módulo de autenticación contiene código seguro.			
El módulo de autenticación se integra un repositorio de datos para la notificación de eventos de autenticación exitosa o fallida.			

Plan de pruebas unitarias

Este plan de pruebas reúne las validaciones de las rutinas lógicas que se emplea en el sistema e indica las entradas y salidas.

Secuencia:	01

Código fuente:

```
private boolean validarDatos(String usuario, String clave){
        if (usuario != null && clave != null) {
                if (usuario.equals("") && clave.equals("")){
                        return false;
                }else{
                        return true;
                }
        }else{
                return false;
        }
}
```

Ingreso:

String usuario = null | “” | “lbenavides”;

String clave = null | “” | “password”;

Salida:

boolean = false | false | true;

Resultado:

Secuencia:	02

Código fuente:

```
private SoapObject transformarMensaje(CredencialSeguridad credencial){
        Properties propiedades = this.propiedades;

        SoapObject request = new
        SoapObject(propiedades.getProperty("NAMESPACE"),
                   propiedades.getProperty("METHOD_NAME"));

         request.addProperty("usuario", credencial.getUsuario());
         request.addProperty("clave", credencial.getClave());
         request.addProperty("imsi", credencial.getImsi());
         return request;
}
```

Ingreso:

CredencialSeguridad = credencial;

Salida:
SoapObject (usuario, clave, imsi)
Resultado:

Secuencia:	03
Código fuente:	

```
private SoapObject transformarMensaje(GestorNotificaciones notificacion){
        Properties propiedades = this.propiedades;

        SoapObject request = new
        SoapObject(propiedades.getProperty("NAMESPACE_EVENT"),
                   propiedades.getProperty("METHOD_NAME_EVENT"));

         request.addProperty("codigo", notificacion.getCodigo());
         request.addProperty("descripcion", notificacion.getMensaje());
         request.addProperty("categoria", notificacion.getCategoria());
         return request;
}
```

Ingreso:
GestorNotificaciones = notificacion;
Salida:
SoapObject (codigo, descripcion, categoria)
Resultado:

Secuencia:	04
Código fuente:	

```
private String recuperarIMSI(Activity actividad){

        String imsi = null;
        TelephonyManager telephonyManager = (TelephonyManager)
        this.actividad.getSystemService(Activity.TELEPHONY_SERVICE);
        imsi = telephonyManager.getSubscriberId();
```

```
        if (imsi != null){
                if (imsi.equals("")){
                        return null;
                }else{
                        return imsi;
                }
        }else{
                return null;
        }
}
```

Ingreso:

Actividad = actividad;

Salida:

String = imsi;

Resultado:

Secuencia:	05

Código fuente:

```
public void autenticarUsuario(String usuario, String clave) {
        String imsi = null;
        SoapObject requerimiento = null;
        if (validarDatos(usuario, clave)){
                imsi = recuperarIMSI(this.actividad);
                if (imsi != null){
                    crearMensaje(usuario, clave, imsi);
                    requerimiento = transformarMensaje(this.credencial);
                    this.tAuth.execute(requerimiento);
                }else{
                     mostrarDenegarAcceso("Autenticacion fallida!",
                                    "Error en la recuperacion del IMSI");
                }
        }else{
                mostrarDenegarAcceso("Autenticacion fallida!", "Usuario o
                                clave incorrecta");
        }
}
```

Ingreso:

String usuario = null | “” | “cmerchan”;

String clave = null | “” | “password”;

Salida:

Mensaje de aplicación con respuesta.
Resultado:

Plan de pruebas funcionales con usuarios

Las pruebas funcionales enfocadas en el usuario son diferenciadas por el rol o función que cumplen los usuarios dentro del proceso. Los siguientes escenarios se identifican por rol.

Rol: Usuario final del sistema

Ingreso al sistema desde teléfono móvil			
Descripción	**Resultado esperado**	**Resultado obtenido**	**Observaciones**
El usuario ingresa su nombre de usuario y clave único de la organización usando un dispositivo móvil previamente registrado.	Aplicación móvil notifica al usuario si otorga o deniega acceso de acuerdo a los datos ingresados		
El usuario no ingresa nombre de usuario y/o clave.	La aplicación móvil notifica al usuario que debe realizar el ingreso correcto de los datos.		
El usuario	La aplicación		

ingresa su nombre de usuario y clave desde un dispositivo no registrado	móvil notifica al usuario que su autenticación es fallida.		
El usuario ingresa su nombre de usuario y clave pero no se encuentra activo en el servicio de directorio.	La aplicación móvil notifica al usuario que su autenticación es fallida.		
El usuario ingresa su nombre de usuario y clave desde un dispositivo móvil perteneciente a otro usuario.	La aplicación móvil notifica al usuario que su autenticación es fallida.		

Rol: Administrador del sistema

Integración de módulo de autenticación con sistemas de la organización			
Descripción	**Resultado esperado**	**Resultado obtenido**	**Observaciones**
El módulo de autenticación se integra con el Servicio Web de la organización.	Mediante la integración entre sistemas se evidencia el envío y recepción de mensajes.		
El módulo de autenticación se integra con el Servicio Web de notificaciones de eventos.	Mediante la integración entre sistemas se evidencia el envío y recepción de mensajes.		
El módulo de	Las aplicaciones		

autenticación se integra con las aplicaciones móviles en Android superior a versiones Jelly Bean o Android 4.1.2	se integran con el módulo y se acoplan con transparencia.		
El módulo de autenticación se integra con las aplicaciones móviles en Android inferior a versiones Jelly Bean o Android 4.1.2	Posible inestabilidad o incompatibilidad en la integración de aplicaciones		
Bitácora de eventos generados por los intentos de acceso a través de los sistemas integrados.	Se registran los eventos en un repositorio centralizado de la organización.		
Las actualizaciones de las cuentas de usuarios en el servicio de directorio se reflejan inmediatamente en el proceso de autenticación.	Cambios en las cuentas de usuarios y atributos relacionados a la autenticación o dispositivos móviles, siempre que el Servicio Web lo detecte, están disponibles en línea para las aplicaciones móviles a través del módulo de autenticación.		

CAPÍTULO 6

ANÁLISIS DE RESULTADOS

Los planes de pruebas definidos en el capítulo anterior son evaluados en el presente capítulo y se detallan los resultados satisfactorios o fallidos junto a su respectiva justificación.

6.1 Resultado de las pruebas con los requerimientos

Requerimientos	¿Cumplió con el requerimiento?		Observaciones
	Si	No	
El módulo de autenticación se adapta a un modelo de autenticación centralizada usando un servicio de directorio que está	X		Para la implementación se utilizó el servicio de directorio OpenLDAP

disponible en la organización.			
El módulo de autenticación usa los atributos existentes del servicio de directorio de la organización identificados para la autenticación del usuario.	X		Previa definición de nombre de atributos usando en la autenticación.
El módulo de autenticación utiliza las opciones de configuración existentes del servicio de directorio para definir nuevos atributos válidos para la autenticación del usuario o de un dispositivo autorizado.	X		Previa definición de atributos adicionales asociados a los dispositivos móviles.
El módulo de autenticación utiliza Servicios Web definidos por la organización para el acceso al servicio de directorio.	X		Comunicación establecida bajo protocolo SOAP
El módulo de autenticación utiliza el canal seguro o	X		

encriptado SSL para establecer comunicación con el Servicio Web.			
El módulo de autenticación envía e interpreta los mensajes de los Servicios Web de la organización para la interacción con el servicio de directorio.	X		Previa definición de estructura de mensajes
El módulo de autenticación esta implementado en Android.	X		
El módulo de autenticación es modular e integrable a una aplicación Android.	X		Desde versiones de Android Jelly Bean.
El módulo de autenticación valida las credenciales de un usuario definidas en la organización.	X		
El módulo de autenticación valida atributos definidos en la organización y asociados a los dispositivos móviles	X		

que posee un usuario.			
El módulo de autenticación contiene código seguro.	X		Validación realizada con herramienta SonarQube (Ver Anexo B).
El módulo de autenticación se integra un repositorio de datos para la notificación de eventos de autenticación exitosa o fallida.	X		Previa definición de estructuras de mensajes.

6.2 Resultado de las pruebas funcionales unitarias

Secuencia:	01
Código fuente:	

```
private boolean validarDatos(String usuario, String clave){
        if (usuario != null && clave != null) {
                if (usuario.equals("") && clave.equals("")){
                        return false;
                }else{
                        return true;
                }
        }else{
                return false;
        }
}
```

Ingreso:
String usuario = null | “” | “lbenavides”; String clave = null | “” | “password”;
Salida:

boolean = false | false | true;
Resultado:
La función devuelve verdadero siempre que los valores de los Strings usuario y clave no sean incorrectos, es decir, nulos o vacíos.

Secuencia:	02
Código fuente:	

```
private SoapObject transformarMensaje(CredencialSeguridad credencial){
        Properties propiedades = this.propiedades;

        SoapObject request = new
        SoapObject(propiedades.getProperty("NAMESPACE"),
                   propiedades.getProperty("METHOD_NAME"));

         request.addProperty("usuario", credencial.getUsuario());
         request.addProperty("clave", credencial.getClave());
         request.addProperty("imsi", credencial.getImsi());
         return request;
}
```

Ingreso:
CredencialSeguridad = credencial;
Salida:
SoapObject (usuario, clave, imsi)
Resultado:
El resultado es una instancia de la Clase SoapObject con los valores del namespace y el método del Servicio Web. La instancia contiene los parámetros definidos de usuario, clave e IMSI.

Secuencia:	03
Código fuente:	

```
private SoapObject transformarMensaje(GestorNotificaciones notificacion){
```

```
        Properties propiedades = this.propiedades;

        SoapObject request = new
        SoapObject(propiedades.getProperty("NAMESPACE_EVENT"),
                   propiedades.getProperty("METHOD_NAME_EVENT"));

         request.addProperty("codigo", notificacion.getCodigo());
         request.addProperty("descripcion", notificacion.getMensaje());
         request.addProperty("categoria", notificacion.getCategoria());
         return request;
}
```

Ingreso:

GestorNotificaciones = notificacion;

Salida:

SoapObject (codigo, descripcion, categoria)

Resultado:

El resultado es una instancia de la Clase SoapObject con los valores del namespace y el método del Servicio Web. La instancia contiene los parámetros definidos de código, descripción y categoría.

Secuencia:	04

Código fuente:

```
private String recuperarIMSI(Activity actividad){

        String imsi = null;
        TelephonyManager telephonyManager = (TelephonyManager)
        this.actividad.getSystemService(Activity.TELEPHONY_SERVICE);
        imsi = telephonyManager.getSubscriberId();
        if (imsi != null){
                if (imsi.equals("")){
                        return null;
                }else{
                        return imsi;
                }
        }else{
                return null;
        }
}
```

Ingreso:

Actividad = actividad;

Salida:
String = imsi;
Resultado:
La función devuelve el IMSI asociado a la simcard alojada en el dispositivo móvil siempre que la aplicación tenga los permisos de sistema para acceder al estado del teléfono.

Secuencia:	05

Código fuente:

```
public void autenticarUsuario(String usuario, String clave) {
        String imsi = null;
        SoapObject requerimiento = null;
        if (validarDatos(usuario, clave)){
                imsi = recuperarIMSI(this.actividad);
                if (imsi != null){
                    crearMensaje(usuario, clave, imsi);
                    requerimiento = transformarMensaje(this.credencial);
                    this.tAuth.execute(requerimiento);
                }else{
                     mostrarDenegarAcceso("Autenticacion fallida!",
                                    "Error en la recuperacion del IMSI");
                }
        }else{
                mostrarDenegarAcceso("Autenticacion fallida!", "Usuario o
                                clave incorrecta");
        }
}
```

Ingreso:
String usuario = null | “” | “lbenavides”; String clave = null | “” | “password”;
Salida:
Mensaje de aplicación con respuesta.
Resultado:
La autenticación del usuario se realiza en otro hilo de ejecución y

devuelve la respuesta a la interfaz para la notificación al usuario.

6.3 Resultado de las pruebas con los usuarios

Rol: Usuario final del sistema

Ingreso al sistema desde teléfono móvil			
Descripción	**Resultado esperado**	**Resultado obtenido**	**Observaciones**
El usuario ingresa su nombre de usuario y clave único de la organización usando un dispositivo móvil previamente registrado.	Aplicación móvil notifica al usuario si otorga o deniega acceso de acuerdo a los datos ingresados	Mensaje de autenticación exitosa enviado al usuario	
El usuario no ingresa nombre de usuario y/o clave.	La aplicación móvil notifica al usuario que debe realizar el ingreso correcto de los datos.	Mensaje de autenticación fallida enviado al usuario	
El usuario ingresa su nombre de usuario y clave desde un dispositivo no registrado	La aplicación móvil notifica al usuario que su autenticación es fallida.	Mensaje de autenticación fallida enviado al usuario	
El usuario ingresa su nombre de usuario y clave pero no se encuentra activo en el servicio de directorio.	La aplicación móvil notifica al usuario que su autenticación es fallida.	Mensaje de autenticación fallida enviado al usuario	

El usuario ingresa su nombre de usuario y clave desde un dispositivo móvil perteneciente a otro usuario.	La aplicación móvil notifica al usuario que su autenticación es fallida.	Mensaje de autenticación fallida enviado al usuario	

Rol: Administrador del sistema

Integración de módulo de autenticación con sistemas de la organización			
Descripción	**Resultado esperado**	**Resultado obtenido**	**Observaciones**
El módulo de autenticación se integra con el Servicio Web de la organización.	Mediante la integración entre sistemas se evidencia el envío y recepción de mensajes.	Integración exitosa	
El módulo de autenticación se integra con el Servicio Web de notificaciones de eventos.	Mediante la integración entre sistemas se evidencia el envío y recepción de mensajes.	Integración exitosa	
El módulo de autenticación se integra con las aplicaciones móviles en Android superior a versiones Jelly Bean o Android 4.1.2	Las aplicaciones se integran con el modulo y se acoplan con transparencia.	Integración exitosa	
El módulo de autenticación se integra con las aplicaciones móviles en Android inferior a	Posible inestabilidad o incompatibilidad en la integración de aplicaciones	Sistemas inestables	

versiones Jelly Bean o Android 4.1.2			
Bitácora de eventos generados por los intentos de acceso a través de los sistemas integrados.	Se registran los eventos en un repositorio centralizado de la organización.	Consultas y reportes exitosos	
Las actualizaciones de las cuentas de usuarios en el servicio de directorio se reflejan inmediatamente en el proceso de autenticación.	Cambios en las cuentas de usuarios y atributos relacionados a la autenticación o dispositivos móviles, siempre que el Servicio Web lo detecte, están disponibles en línea para las aplicaciones móviles a través del módulo de autenticación.	Los cambios son reflejados en la próxima ejecución de la aplicación móvil.	

6.4 Análisis de resultados

Después de los resultados expuestos en los diferentes planes de pruebas se determina que el módulo de autenticación reúne los requerimientos funcionales y de seguridad solicitados y que cumple con los criterios de aceptación de usuario.

CONCLUSIONES Y RECOMENDACIONES

Al finalizar las etapas de análisis, diseño e implementación del módulo de autenticación, se presentan las siguientes conclusiones y recomendaciones para que sean consideradas en una futura implementación o mejora al sistema.

Conclusiones

1. El contar con módulo de autenticación que utilice los recursos del servicio de directorio, permite una gestión centralizada de credenciales para cada aplicativo con el que cuenta LaMerced, minimizando la carga operativa en la administración de usuarios.
2. La autenticación de usuarios hacia un servicio de directorio desde aplicaciones móviles es viable y abre nuevas vías en el control de seguridad y centralización de la información sensible.
3. El registro de sucesos en una transacción hacia un repositorio centralizado permite a los administradores y personal técnico de seguridad detectar

ataques informáticos y realizar un seguimiento de las acciones de los usuarios para determinar fallas de aplicación.

4. El uso de librerías propias de la plataforma Android, evitando recursos ofrecidos por terceros, ofrece una mayor seguridad en el desarrollo para dispositivos móviles.
5. El proceso de autenticación por medio del módulo desarrollado, permite a los usuarios finales de LaMerced, contar con un mecanismo de acceso con credenciales únicas para todos los sistemas informáticos de la organización, minimizando significativamente la carga cognitiva que representaba ingresar a cada sistema con credenciales distintas.

Recomendaciones

1. Realizar la implementación del módulo de autenticación para el sistema operativo iOS y así atender a futuro, a un grupo considerable de usuarios que usan dispositivos Apple.
2. Realizar actualizaciones periódicas al módulo de autenticación de acuerdo a las nuevas versiones que el sistema operativo Android vaya liberando.
3. Implementar en el gestor de notificaciones un camino redundante de conexión que permita el registro de eventos aun cuando no se tengan acceso al Servicio Web.
4. Incorporar nuevos atributos de seguridad de los dispositivos móviles cuando no se disponga de los valores de la IMSI, como el IMEI o Mac Address de la interfaz de red del dispositivo.

BIBLIOGRAFÍA

[1] CISCO, 2014 Connected World Technology Final Report, http://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise/connected-world-technology-report/cisco-2014-connected-world-technology-report.pdf, fecha de consulta 24 de junio del 2015.

[2] VERACODE, State of Software Security Report, http://www.veracode.com/sites/default/files/Resources/Reports/state-of-software-security-volume-5-report.pdf?mkt_tok=3RkMMJWWfF9wsRolv63IZKXonjHpfsX77OwrW662lMI%2F0ER3fOvrPUfGjI4FS8RqI%2BSLDwEYGJlv6SgFTbnFMbprzbgPUhA%3D, fecha de consulta 1 de julio de 2015.

[3] IDC, Worldwide Smartphone Shipments Edge Past 300 Million Units in the Second Quarter; Android and iOS Devices Account for 96% of the Global Market, According to IDC, http://www.idc.com/getdoc.jsp?containerId=prUS25037214, fecha de consulta 1 de julio de 2015.

[4] Wei-Meng Lee, Android application development, Wiley, 2011.

[5] Apple developer, iOS developer library, https://developer.apple.com/library/ios/navigation/, fecha de consulta 14 de septiembre de 2015.

[6] Activity, Android Developers, http://developer.android.com/reference/android/app/Activity.html, fecha de consulta 5 de octubre de 2015.

[7] The App Life Cycle, https://developer.apple.com/library/ios/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/TheAppLifeCycle/TheAppLifeCycle.html, fecha de consulta 5 de octubre de 2015.

[8] Appropriate uses for SQLite, https://www.sqlite.org/whentouse.html, fecha de consulta 5 de octubre de 2015.

[9] Romano Agustín V., Descripción y Análisis Formal del Modelo de Seguridad de Android, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional del Rosario, Argentina, 2014.

[10] Elenkov Nikolay, Android Security Internals, No Starch Press Inc., San Francisco USA, 2015.

[11] Elmasri R., Navathe S. B., Fundamentos de Sistemas de Bases de Datos, Pearson Educación S.A. 5ta. Ed., 2011.

[12] Kendall K., Kendall J., Análisis y Diseño de Sistemas, Prentice Hall 8ta. Ed., 2011.

[13] Dwivedi H., Clark C., Thiel D., Mobile Application Security, McGraw-Hill, 2010.

[14] ISO/IEC 27001:2013 Information technology — Security techniques — Information security management systems

[15] Introducción a Active Directory, https://support.microsoft.com/es-es/kb/196464, fecha de consulta 14 de octubre de 2015.

[16] OpenLDAP Project Overview, http://www.openldap.org/project/, fecha de consulta 14 de octubre de 2015.

[17] Active Directory Service Interfaces, https://msdn.microsoft.com/en-us/library/aa772170(v=vs.85).aspx, fecha de consulta 15 de octubre del 2015.

[18] Dashboard, http://developer.android.com/about/dashboards/index.html fecha de consulta 15 de octubre del 2015.

[19] Clausulas, Análisis y Tratamiento de Riesgos, ISO/IEC 27002:2013, fecha de consulta 15 de octubre del 2015.

[20] Top 500, Sistemas Operativos, Mainframe's, https://commons.wikimedia.org/wiki/File:Operating_systems_used_on_top_500_supercomputers.svg?uselang=es, fecha de consulta 18 de octubre del 2015.

[21] IDC, Proyección de crecimiento en ambientes para sistemas operativos, http://image.slidesharecdn.com/susesapbusinesswebinarpresentation-140331064326-phpapp02/95/suse-sap-businesswebinarpresentation-18-638.jpg?cb=1396248262, fecha de consulta 18 de octubre del 2015.

[22] Fierro Ana C., Apps locales en desarrollo, Diario Metro, Guayaqui-Ecuador. Agosto 2015.

[23] Commercial Edition Release Notes, https://docs.ldap.com/ldap-sdk/docs/commercial-edition/release-notes.html, fecha de consulta 20 de octubre de 2015.

ANEXO A

ANÁLISIS DE RIESGOS PARA LA IMPLEMENTACIÓN DE UN MÓDULO DE AUTENTICACIÓN PARA APLICACIONES MÓVILES DE LA EMPRESA LAMERCED

La empresa LaMerced, dedicada a la comercialización de productos químicos para el agro, consciente de la importancia de ir acorde a los avances tecnológicos y los riesgos que estos pueden implicar, ha decidido por medio de su junta directiva, delegar un grupo de colaboradores para la realización de una evaluación de riesgos asociados al desarrollo del proyecto de Implementación de un módulo de autenticación para aplicaciones móviles.

El grupo de evaluación de riesgos está conformado por Ing. Daniel Merchan, Presidente de la Junta Directiva; Msc. Mercedes Benavides, Subgerente de tecnología de Información; Ing. Christian Millán, Sugerente de Organización y Métodos.

Identificación de Riesgos

El grupo de evaluación de riesgos ha identificado las siguientes amenazas y vulnerabilidades:

- No reconocer a un usuario legítimo como tal: Un usuario que debe tener acceso al sistema, no se le concede el acceso desde su dispositivo móvil.
- Aceptar como usuario legítimo a quien no lo es: Un usuario que realiza un intento al sistema, se le concede el acceso a pesar de contar con privilegios para el mismo.
- Olvido de credenciales de autenticación: Un usuario legítimo del sistema, no recuerda los mecanismos de autenticación que debe proporcionar para lograr el acceso al sistema.
- Robo de credenciales: Un tercero ha logrado tener acceso a las credenciales o dispositivos móviles de un usuario legítimo.
- Ataque de fuerza bruta: Un tercero trata de forzar el ingreso al sistema, realizando varios intentos hasta lograr su cometido.
- Ausencia del canal de comunicación: No se obtiene un medio de conexión entre el dispositivo móvil y el servidor de aplicaciones de la empresa.

Valoración de las amenazas

De modo de poder cuantificar las amenazas y vulnerabilidades encontradas, se ha asociado el tipo de pérdida que ocasionaría cada una de ellas:

TIPO DE PERDIDA	**EVENTO DE RIESGO**
Pérdidas financieras	Robo de credenciales
Publicación no autorizada de información sensible	Aceptar como legitimo a un usuario que no lo es
Indisponibilidad del servicio	Ausencia del canal de comunicación
Pérdida de imagen	No reconocer a un usuario legitimo como tal
Discontinuidad del negocio	Ataque de fuerza bruta
Incumplimiento de la misión empresarial	Olvido de credenciales de autenticación

Una vez establecido el tipo de pérdida, el grupo de evaluación ha ponderado su probabilidad de ocurrencia y el impacto que ocasionaría el que se materialicen. Para esto, se ha tomado la siguiente escala:

PONDERACION	PROBABILIDAD	IMPACTO
1	Raro	Despreciable
2	Improbable	Bajo
3	Moderado	Medio
4	Probable	Muy Alto
5	Casi seguro	Extremo

La primera valoración se ha realizado sin considerar la implementación de control alguno, dando como resultado lo siguiente.

Tipo de Pérdida	**Evento de riesgo**	**No hay control Riesgo Inminente**	
		Probabilidad	**Impacto**
Pérdidas financieras	Robo de credenciales	4	4
Publicación no autorizada de información sensible	Aceptar como legítimo a un usuario que no lo es	2	4
Indisponibilidad del servicio	Ausencia del canal de comunicación	1	2
Pérdida de imagen	No reconocer a un usuario legítimo como tal	1	3

Discontinuidad del negocio	Ataque de fuerza bruta	4	3
Incumplimiento de la misión empresarial	Olvido de credenciales de autenticación	2	1

Para facilitar el análisis, se ha realizado el siguiente mapa de valoración de riesgo:

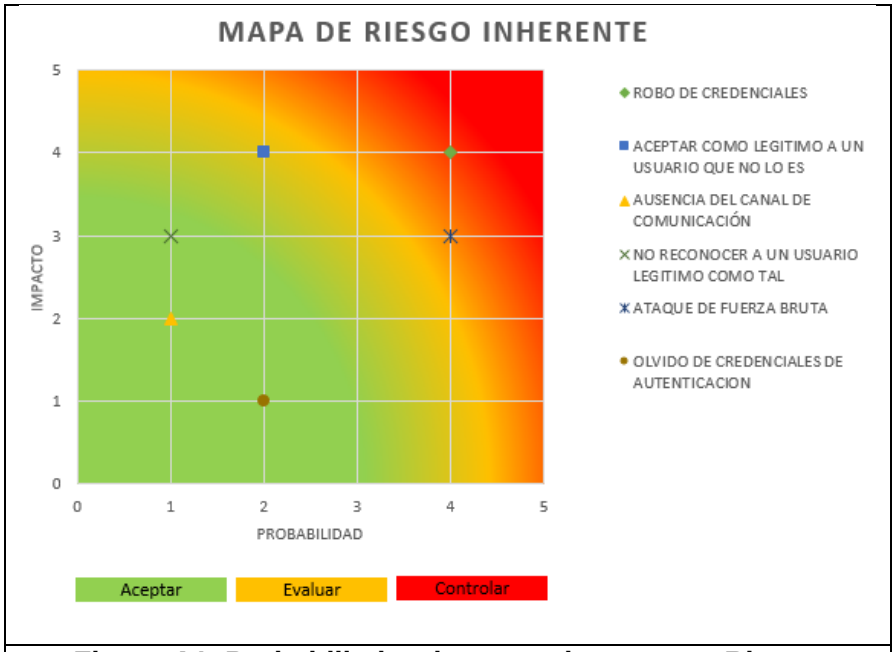


Figura A1. Probabiliad vs impacto de eventos. Riesgo inherente.

En base a este primer análisis, se ha determinado que las amenazas o vulnerabilidades que requieren un control, dado su nivel de riesgo inminente son: Robo de credenciales, Ataque de fuerza bruta y Aceptar como válido a un usuario que no lo es.

Tratamiento del Riesgo

Haciendo uso de los controles recomendados en ISO/IEC 27002, ha decidido que los eventos se tratarán de la siguiente manera:

- Robo de credenciales
  - **Registro de Usuario:** Debe existir un procedimiento formal para el registro y de-registro de usuario para otorgar y revocar el acceso a todos los sistemas y servicios de información de LaMerced. La definición de este procedimiento está a cargo de la organización.
  - **Identificación del equipo en las redes:** La identificación automática del equipo se debiera considerar como un medio para autenticar las conexiones de ubicaciones y equipos específicos.
  - **Cierre de una sesión por inactividad:** Las sesiones inactivas debieran ser cerradas después de un periodo de inactividad definido. El tiempo de inactividad máximo permitido debe estar estipulado formalmente en los procedimientos de seguridad de LaMerced.
- Ataque de fuerza bruta
  - **Validación de la input data:** Se debiera validar la input data para las aplicaciones para asegurar que esta data sea correcta y apropiada.
- Aceptar como usuario legítimo a quien no lo es.
  - **Retiro de los derechos de acceso:** Los derechos de acceso de todos los usuarios empleados, contratistas y terceras personas a la información y los medios de procesamiento de información debieran

ser retirados a la terminación de su empleo, contrato o acuerdo, o debieran ser reajustados de acuerdo al cambio. Este procedimiento debe ser parte de los existentes en LaMerced.

Restricción del acceso a la información: El acceso de los usuarios y el personal de soporte a la información y las funciones del sistema de la aplicación debiera limitarse en concordancia con la política de control de acceso definida previamente en la empresa.

Una vez aplicado los controles a los eventos de riesgo, se obtiene un riesgo remanente, el cual se ha analizado haciendo uso de la misma escala de ponderación anterior, obteniendo lo siguiente:

Tipo de Pérdida	**Evento de riesgo**	**Aplicando Controles Riesgo Residual**	
		Probabilidad	**Impacto**
Pérdidas financieras	Robo de credenciales	2	3
Publicación no autorizada de información sensible	Aceptar como legítimo a un usuario que no lo es	1	4
Indisponibilidad del servicio	Ausencia del canal de comunicación	1	2

Pérdida de imagen	No reconocer a un usuario legítimo como tal	1	3
Discontinuidad del negocio	Ataque de fuerza bruta	2	2
Incumplimiento de la misión empresarial	Olvido de credenciales de autenticación	2	1

De lo cual se ha obtenido el siguiente mapa de valoración de riesgos:

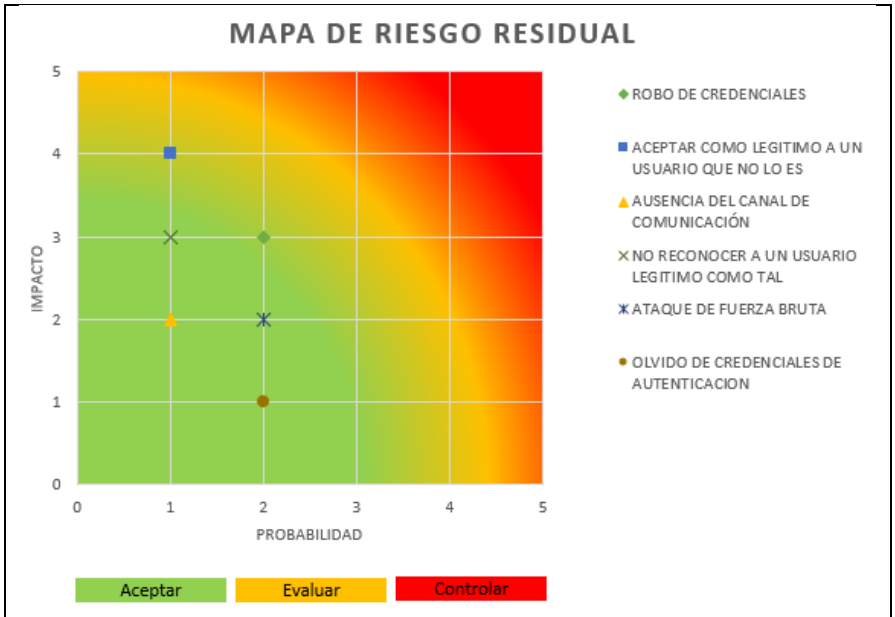


Figura A2. Probabiliad vs impacto de eventos. Riesgo residual.

En base a este resultado, se determina que con la aplicación de los controles a los eventos antes mencionados, es posible aceptar el riesgo y riesgo remanente.

Conclusiones

Del análisis realizado por el grupo evaluador, se concluye que el proyecto es completamente viable, ya que las amenazas encontradas, no representan un riesgo extremo para LaMerced; siempre y cuando se apliquen los controles sugeridos en el presente documento.

Declaración expresa

El grupo de evaluación de riesgos, asume la responsabilidad sobre el análisis realizado.

Para constancia firman:

Ing. Daniel Merchán
Presidente de la Junta Directiva

Msc. Mercedes Benavides
Subgerente de tecnología de Información

Ing. Christian Merchán M.
Sugerente de Organización y Métodos.

ANEXO B

ANÁLISIS DE CÓDIGO ESTÁTICO

El siguiente análisis de código estático, fue desarrollado utilizando la herramienta de análisis SonarQube, siguiendo los lineamientos mínimos esperados de acuerdo al procedimiento establecido y aprobado en LaMerced para el análisis de código estático para el desarrollo interno de software.

Archivo properties empleado para el análisis del proyecto:

```
# Required metadata
sonar.projectKey=org.sonarqube:java-sonar-runner-simple
sonar.projectName=Java :: ModuloAutenticacion :: SonarQube Runner
sonar.projectVersion=1.0

# Comma-separated paths to directories with sources (required)
```

```
sonar.sources=src

# Language
sonar.language=java

# Encoding of the source files
sonar.sourceEncoding=UTF-8
```

Ejecución de SonarQube sobre el proyecto:

```
C:\Users\Lilian\Documents\workspace\moduloautenticacion>c:\sonar-runner-2.4\bin\
sonar-runner.bat -e
c:\sonar-runner-2.4\bin\..
SonarQube Runner 2.4
Java 1.8.0_65 Oracle Corporation (64-bit)
Windows 7 6.1 amd64
INFO: Error stacktraces are turned on.
INFO: Runner configuration file: c:\sonar-runner-2.4\bin\..\conf\sonar-runner.pr
operties
INFO: Project configuration file: C:\Users\Lilian\Documents\workspace\moduloaute
nticacion\sonar-project.properties
INFO: Default locale: "en_US", source code encoding: "UTF-8"
INFO: Work directory: C:\Users\Lilian\Documents\workspace\moduloautenticacion\.\
.sonar
```

INFO: SonarQube Server 5.2

13:27:47.857 INFO - Load global repositories

13:27:48.158 INFO - Load global repositories (done) | time=307ms

13:27:48.180 INFO - User cache: C:\Users\Lilian\.sonar\cache

13:27:48.723 INFO - Load plugins index

13:27:48.741 INFO - Load plugins index (done) | time=18ms

13:27:49.010 INFO - Process project properties

13:27:49.298 INFO - Load project repositories

13:27:49.439 INFO - Load project repositories (done) | time=141ms

13:27:49.451 INFO - Apply project exclusions

13:27:50.476 INFO - Load quality profiles

13:27:50.521 INFO - Load quality profiles (done) | time=45ms

13:27:50.526 INFO - Load active rules

13:27:50.954 INFO - Load active rules (done) | time=428ms

13:27:50.979 WARN - SCM provider autodetection failed. No SCM provider claims t
o support this project. Please use sonar.scm.provider to define SCM of your proj
ect.

13:27:50.980 INFO - Publish mode

13:27:50.982 INFO - ---------- Scan Java :: ModuloAutenticacion :: SonarQub
e Runner

13:27:51.096 INFO - Language is forced to java

13:27:51.103 INFO - Load server rules

13:27:51.197 INFO - Load server rules (done) | time=94ms

13:27:51.266 INFO - Base dir: C:\Users\Lilian\Documents\workspace\moduloautenti

cacion

13:27:51.266 INFO - Working dir: C:\Users\Lilian\Documents\workspace\moduloautenticacion\.sonar

13:27:51.271 INFO - Source paths: src

13:27:51.271 INFO - Source encoding: UTF-8, default locale: en_US

13:27:51.272 INFO - Index files

13:27:51.315 INFO - 3 files indexed

13:27:51.319 INFO - Quality profile for java: Sonar way

13:27:51.382 INFO - Sensor JavaSquidSensor

13:27:52.210 INFO - Java Main Files AST scan...

13:27:52.214 INFO - 3 source files to be analyzed

13:27:53.236 INFO - Java Main Files AST scan done: 1026 ms

13:27:53.238 INFO - 3/3 source files have been analyzed

13:27:53.238 WARN - Java bytecode has not been made available to the analyzer. The org.sonar.java.bytecode.visitor.DependenciesVisitor@40fa8766, org.sonar.java.checks.UnusedPrivateMethodCheck@503d56b5 are disabled.

13:27:53.240 INFO - Java Test Files AST scan...

13:27:53.241 INFO - 0 source files to be analyzed

13:27:53.243 INFO - Java Test Files AST scan done: 3 ms

13:27:53.244 INFO - 0/0 source files have been analyzed

13:27:53.293 INFO - Sensor JavaSquidSensor (done) | time=1911ms

13:27:53.293 INFO - Sensor Lines Sensor

13:27:53.298 INFO - Sensor Lines Sensor (done) | time=5ms

13:27:53.300 INFO - Sensor QProfileSensor

13:27:53.310 INFO - Sensor QProfileSensor (done) | time=10ms

13:27:53.312 INFO - Sensor SurefireSensor

13:27:53.314 INFO - parsing C:\Users\Lilian\Documents\workspace\moduloautentica cion\target\surefire-reports

13:27:53.315 ERROR - Reports path not found or is not a directory: C:\Users\Lili an\Documents\workspace\moduloautenticacion\target\surefire-reports

13:27:53.317 INFO - Sensor SurefireSensor (done) | time=5ms

13:27:53.317 INFO - Sensor SCM Sensor

13:27:53.322 INFO - No SCM system was detected. You can use the 'sonar.scm.prov ider' property to explicitly specify it.

13:27:53.323 INFO - Sensor SCM Sensor (done) | time=6ms

13:27:53.324 INFO - Sensor Code Colorizer Sensor

13:27:53.326 INFO - Sensor Code Colorizer Sensor (done) | time=2ms

13:27:53.326 INFO - Sensor CPD Sensor

13:27:53.326 INFO - JavaCpdEngine is used for java

13:27:53.396 INFO - Sensor CPD Sensor (done) | time=70ms

13:27:53.520 INFO - Analysis reports generated in 112ms, dir size=26 KB

13:27:53.555 INFO - Analysis reports compressed in 35ms, zip size=13 KB

13:27:53.643 INFO - Analysis reports sent to server in 87ms

13:27:53.645 INFO - ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/d

ashboard/index/org.sonarqube:java-sonar-runner-simple

13:27:53.646 INFO - Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report.

13:27:53.646 INFO - More about the report processing at http://localhost:9000/api/ce/task?id=AVD3zdaWmPMIjypRWwN7

INFO: --

INFO: EXECUTION SUCCESS

INFO: --

Total time: 6.737s

Final Memory: 9M/144M

INFO: --

C:\Users\Lilian\Documents\workspace\moduloautenticacion>

Resultados obtenidos en el análisis:

La calidad del proyecto, cumple con los lineamientos establecidos en LaMerced, como podemos observar en la Figura B1

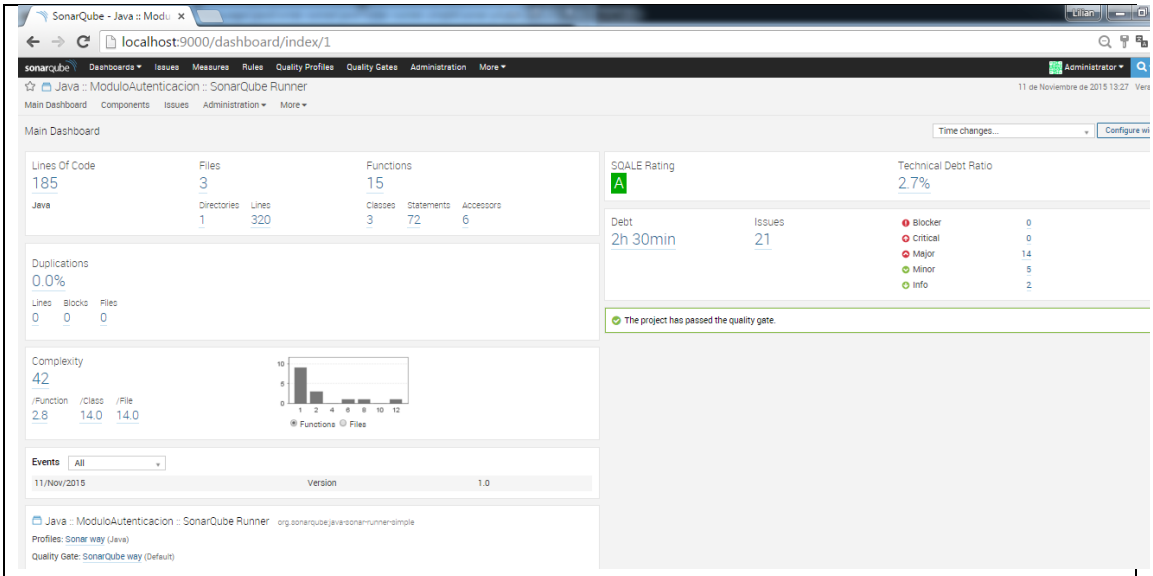


Figura B1. Dashboard de SonarQube para el proyecto Módulo de autenticación

Las novedades encontradas en el código que se sugieren ser corregidas en cada una de las clases del proyecto, se las detalla en las figuras B2, B3 y B4.

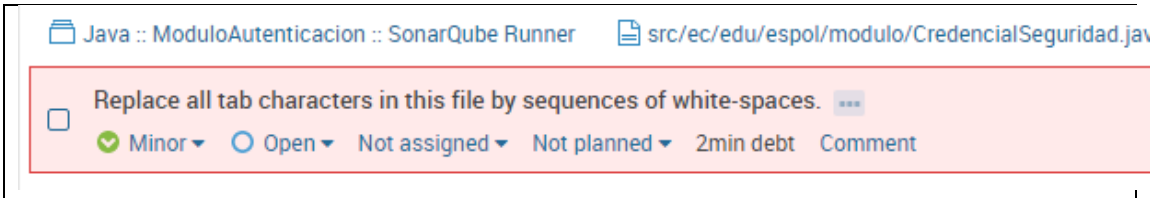


Figura B2. Novedades en el código de la clase CredencialSeguridad

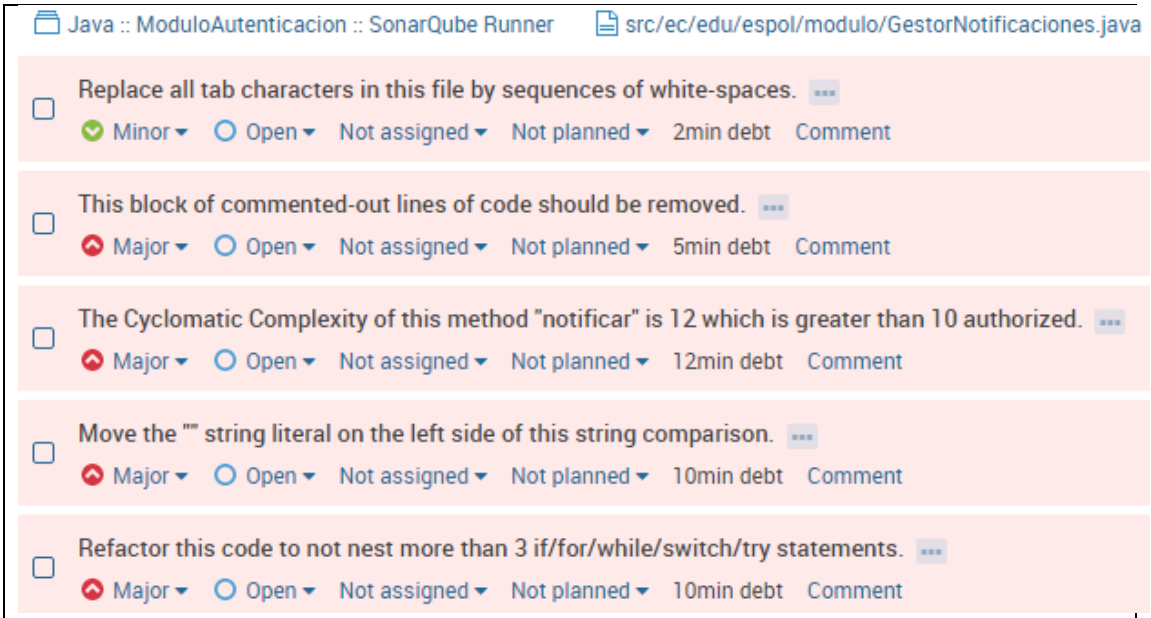


Figura B3. Novedades en el código de la clase GestorNotificaciones

Figura B4. Novedades en el código de la clase MóduloAutenticación