



UNIVERSIDAD AUTONOMA DEL ESTADO DE MEXICO

CENTRO UNIVERSITARIO UAEM TEXCOCO

“CONSULTA DE EXPEDIENTE MEDICO EN LINEA A
TRAVES DE LAS PLATAFORMAS JAVA Y ANDROID”

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACIÓN

PRESENTA: SALVADOR CASTRO GÓMEZ

DIRECTOR: Dr. En C. OZIEL LUGO ESPINOSA

REVISORES:

Dr. En Ed. JOEL AYALA DE LA VEGA

Dr. En C. ALFONSO ZARCO HIDALGO

CONTENIDO

AGRADECIMIENTOS	3
INDICE DE FIGURAS	6
INDICE DE TABLAS	7
RESUMEN	8
INTRODUCCIÓN	9
JUSTIFICACIÓN	10
OBJETIVO PRINCIPAL.....	10
Objetivos Específicos	10
Capítulo 1.	11
INGENIERÍA DE SOFTWARE	11
1.1. Metodología SCRUM	11
1.2. Los Elementos de Scrum	12
1.3. Requisitos en el desarrollo ágil	13
1.3.1. Requisitos y visión del producto.....	14
1.3.2. Pila del producto: los requisitos del cliente	16
1.3.3. Formato de la pila del producto	17
1.3.4. Pila del Sprint	18
1.3.5. El incremento.....	18
1.4. Plataformas de desarrollo.....	19
1.4.1. Android	19
1.4.2 Java	20
1.4.3. Los Inicios de Android y Java	22
1.4.4. Android VS Otras Plataformas	23
1.5 MySQL	25
1.5.1 Limitaciones	26
1.6. SQLite.....	27
CAPÍTULO 2.	29
DISEÑO Y ANALISIS DEL SISTEMA	29
2.1. Establecimiento de requerimientos	30
2.1.1. Principales Problemas.....	32
2.2. Análisis del Sistema.....	34
2.2.1. Diagrama de flujo de información	34
2.2.2. Diagrama de flujo de procesos.....	37
2.2.2.1. Ventajas del Diagrama de Flujo.....	38
2.3. Diseño del sistema	43
2.3.1. Modelo de base de datos	44
2.3.1.1. Modelo de datos entidad-relación.....	45
2.3.1.2. Modelo de base de datos relacional	46
2.3.2. Modelo de Casos de Uso	47
2.3.2.1. Representación de un modelo de caso de uso	48
2.3.2.2. Concepción de las relaciones entre casos de uso	48
2.3.2.2.1. Inclusión	48
2.3.2.2.2. Exclusión	49
2.3.3. Modelo de Clases.....	50
2.3.4. Modelo de secuencia.....	53
2.3.4.1. Objetos.....	53
2.3.4.2. Mensaje.....	54
2.3.4.3. Tiempo.....	54
CAPÍTULO 3.	57
IMPLEMENTACIÓN DEL SISTEMA	57
3.1. Implementación de la base de datos.....	57
3.2. Implementación de HW y SW	57
3.3. Implementación de la interfaz del SCEML.....	58
CAPÍTULO 4.	70
PRUEBAS Y EVALUACIÓN DEL SISTEMA	70
4.1. Imágenes de prueba	70
4.2. Evaluación de tiempo de respuesta.....	77

4.3. Ejemplos de otro sistema de expediente en el mercado.....	80
4.4. Tabla comparativa de ventajas y desventajas de ambas plataformas Java y Android.....	81
5. CONCLUSIONES.....	83
5.1. Oportunidades de Mejora.....	84
Bibliografía.....	85
ANEXOS.....	86
Manual de Usuario.....	86
1. OBJETO DEL DOCUMENTO.....	89
2. OBJETIVOS.....	89
3. MANUAL DE USUARIO JAVA.....	89
3.1. Pantalla Inicial.....	89
3.2. Iniciar Sesión.....	89
3.3. Exportar PDF.....	90
3.4. Registrarse.....	91
4. MANUAL DE USUARIO ADMINISTRADOR.....	92
4.1. Alta de un expediente.....	94
4.2. Buscar expediente.....	95
4.3. Editar expediente.....	96
4.4. Datos asociados.....	97
5. MANUAL DE USUARIO ANDROID.....	97
5.1. Pantalla Inicial.....	97
5.2. Iniciar sesión.....	98
5.3. Registrarse.....	99
5.4. Muestra Expediente.....	101

INDICE DE FIGURAS

Figura 1 Elementos de SCRUM	12
Figura 2 Ámbitos de los requisitos	13
Figura 3 Requisitos de la gestión predictiva vs requisitos de la gestión ágil.	14
Figura 4 Descripciones de los requisitos en la gestión predictiva y la gestión ágil.	16
Figura 5 Diagrama de flujo general para el SCEML	35
Figura 6 Diagrama de flujo del proceso registrarse	36
Figura 7 Diagrama de flujo del proceso registrar expediente	37
Figura 8 Flujo principal del SCEML	41
Figura 9 Proceso principal del SCEML	41
Figura 10 Diagrama de flujo de proceso de registrar expediente.....	43
Figura 11 Modelo entidad – relación SCEML.....	46
Figura 12 Modelo Relacional del SCEML	47
Figura 13 Diagrama de Casos de Uso para el Administrador del SCEML.....	49
Figura 14 Diagrama de Casos de Uso para el Usuario del SCEML	50
Figura 15 Ejemplo de Clase.....	51
Figura 16 Modelo de Clases del SCEML	52
Figura 17 Diagrama de Clases para SCEML Android.....	53
Figura 18 Tipos de mensajes en el diagrama de secuencia.	54
Figura 19 Diagrama de Secuencia (Usuario) SCEML.	55
Figura 20 Diagrama de Secuencia (Administrador) SCEML.	56
Figura 21 Interfaz de Inicio de Sesión SCEML.	58
Figura 22 Datos para el registro de un usuario	59
Figura 23 Datos para el registro de un expediente medico	59
Figura 24 Mensaje sin edición de expediente	63
Figura 25 Interfaz de inicio de sesión en Android	64
Figura 26 Interfaz de registro de usuario para consultar expedientes.	65
Figura 27 Interfaz del expediente en Android	66

INDICE DE TABLAS

Tabla 1 Comparativa de las principales plataformas móviles.....	24
Tabla 2 Ejemplo de una pila de producto	32
Tabla 3 Problemas a los que da solución el SCEML.....	33
Tabla 4 Simbología de un diagrama de flujo	39
Tabla 5 Comparación de tiempos de respuesta entre JAVA y ANDROID.....	79
Tabla 6 Comparativa de ventajas y desventajas entre la plataforma Java y Android	81

RESUMEN

Para muchas organizaciones la actualización de tecnología es una inversión necesaria que tiene como objetivo hacer cada vez más eficientes sus procesos para lograr ser competitivos. Sin embargo en muchas ocasiones el proceso de actualización puede convertirse en un proceso sumamente complicado y laborioso dando como resultado que muchos proyectos se cancelen, terminen incompletos o se concluyan deficientemente con costos demasiado elevados.

Esta tesis es una investigación e implementación de un software realizada en servicios de salud en la Ciudad de México, donde se analizó el impacto de los factores técnicos, administrativos y de comportamiento en un proceso de consulta de expedientes médicos con el fin de encontrar elementos que ayuden a desarrollar un esquema estratégico que facilite la implementación, reducción de costos de operación y un alto nivel productivo.

Los análisis indicaron que se da mayor importancia a los factores técnicos pero que se descuidan los administrativos provocando que factores de comportamiento sean afectados y por lo tanto el desempeño laboral sea muy deficiente. A partir de esto, se plantea un esquema que elimine esta tendencia proponiendo mejoras, como por ejemplo basar el proceso de actualización en un sistema de calidad que promueva la mejora continua, haciendo uso de la administración estratégica, por lo que, el esfuerzo de esta investigación, es ofrecer una propuesta que facilite y sea altamente productivo para un proyecto de actualización con sistemas informáticos para la mejora de procesos del sistema de salud.

INTRODUCCIÓN

En años recientes, ha surgido un crecimiento considerable de la matrícula del 2008 a 2012 se tiene que el número de afiliados al sistema de salud incremento de 5, 111,556 a 5, 805,762 respectivamente lo cual nos dice que se incrementó en 694,206 el número de afiliados en un lapso de 5 años (INEGI, 2014) en las diferentes instituciones del sector salud en la ciudad de México. Lo anterior se debe a que el gobierno otorga seguros, planes, vales entre otros métodos para brindar a la población servicio de salud gratuito, con el fin de apoyar a la sociedad que no tiene los recursos económicos para acudir con un médico particular en caso de Enfermedad.

Por otra parte, se puede mencionar que se requiere del diagnóstico no solo de uno sino de varios médicos especializados para detectar ciertas enfermedades que hayan surgido a lo largo de la vida de los pacientes. Por lo anterior, se generan expedientes médicos cuyo contenido puede variar desde la información demográfica, alérgica, medicamentos prescritos, hasta el historial clínico, social o sexual, etc. Con el objetivo de facilitar la tarea al médico para realizar un diagnóstico acertado basándose en el expediente médico, que puede ser muy largo y tedioso de elaborar, pues se redacta a mano, este proyecto de tesis propone modernizar el sistema de revisión de expedientes médicos a través del uso de Internet y la tecnología computacional y de esta manera se puedan consultar los expedientes médicos en línea. Todo este furor por innovar la captura de expedientes que van más allá del uso del lápiz y papel y el registro monótono de cada dato que debe contener, genera, en respuesta, estos tipos de expedientes electrónicos. De esta manera la población en general podrá consultar su expediente eficazmente, al instante, desde cualquier dispositivo móvil o computadora que cuente con acceso a Internet.

Este estudio tiene como finalidad proponer el desarrollo e implementación de una aplicación en la plataforma llamada Android y Java las cuales poseen una gran cantidad de dispositivos móviles y computadoras personales, en las que se tendrá un “expediente médico en línea” al instante con tan solo pulsar ciertos botones de acuerdo con el usuario, contraseña y número de folio del expediente que se está solicitando, así mismo, obtendrá la información por medio de una base de datos alojada en un servidor en línea.

JUSTIFICACIÓN

Una de las necesidades más apremiantes en las instituciones del sector salud es el implemento de las tecnologías computacionales en la consulta constante de los expedientes médicos extensos y complejos para agilizarla. En la sociedad mexicana existen pocas instituciones públicas y privadas que cuentan con un software de este tipo dirigido tanto a los pacientes como a los médicos, el presupuesto que se le otorga a las instituciones del sector salud se utilizan para favorecer otras cuestiones tales como ascensos, promociones, congresos, entre otros y no se canalizan para apoyar este tipo de innovaciones en la calidad y eficiencia del servicio médico. El uso del papel, las impresoras, las tintas, etc., son factores que afectan al medio ambiente, además que hacen tedioso la realización de los expedientes en las distintas instituciones del sector salud.

A medida que la tecnología avanza se pueden realizar proyectos como el que se propone en este estudio con la finalidad de ayudar al médico, al paciente, a la institución y, ¿por qué no?, hasta el medio ambiente. Las computadoras con las que cuentan las instituciones del sector salud podrían ser utilizadas como escritorio para consultas con tan solo tener la aplicación instalada. Con esto lograríamos reducir tiempo y costo al obtener mayor y mejor producción de la información.

OBJETIVO PRINCIPAL

Desarrollo e implementación de una aplicación en las plataformas Java y Android en la versión 4.0 para la consulta de expedientes médicos electrónicos accesibles tanto para la población como para las distintas instituciones del sector Salud.

Objetivos Específicos

- ✓ Realizar el código de la aplicación en el entorno de desarrollo integrado (IDE) multiplataforma llamado Eclipse para Android y en Netbeans para Java.

Evaluar la eficiencia y buen funcionamiento de la aplicación realizada.

Capítulo 1.

INGENIERÍA DE SOFTWARE

Para el proceso de ingeniería del Sistema de Consulta de Expedientes Médicos en Línea (SCEML) se han empleado metodologías y herramientas modernas, que permiten garantizar un producto de calidad que responda a las necesidades de las instituciones de salud. A continuación se enumeran dichas metodologías:

- Para el proceso de ingeniería del software se ha empleado la metodología SCRUM. Scrum es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

Las herramientas que se utilizarán a lo largo del proceso son:

- Como plataformas de desarrollo se empleará Android 4.0 y Java, ambientes visuales que permiten la interacción con diferentes manejadores de base de datos.
- Como plataformas de desarrollo de las bases de datos se empleara SQLite y MySQL, ambientes visuales que permiten el fácil manejo de las bases de datos y la interacción que puede tener con los lenguajes de programación.

1.1. Metodología SCRUM

Scrum es un proceso en el que se aplican de manera cotidiana un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener así el mejor resultado de un proyecto (Juan Palacio, 1.4.0 – Enero de 2011).

En Scrum se realizan entregas parciales y regulares del producto final, es decir, el software, priorizadas las cuales define el cliente, para ello se lleva a cabo una mesa de trabajo donde se definen entre el cliente y todo el equipo dichos requerimientos y a su vez se etiquetan por nivel de complejidad y prioridad para las entregas definidas por el beneficio que aportan al

receptor del proyecto. Por ello, Scrum está especializado para proyectos en ambientes complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

1.2. Los Elementos de Scrum

Los elementos centrales de ciclo ágil Scrum se muestran en la Figura 1 y son:

- Pila del producto (Product Backlog): Lista de funcionalidades que necesita el cliente.
- Pila del sprint (Sprint Backlog): Lista de tareas que se realizan en un sprint (iteración).
- Incremento: Parte del sistema desarrollada en un sprint (iteración).

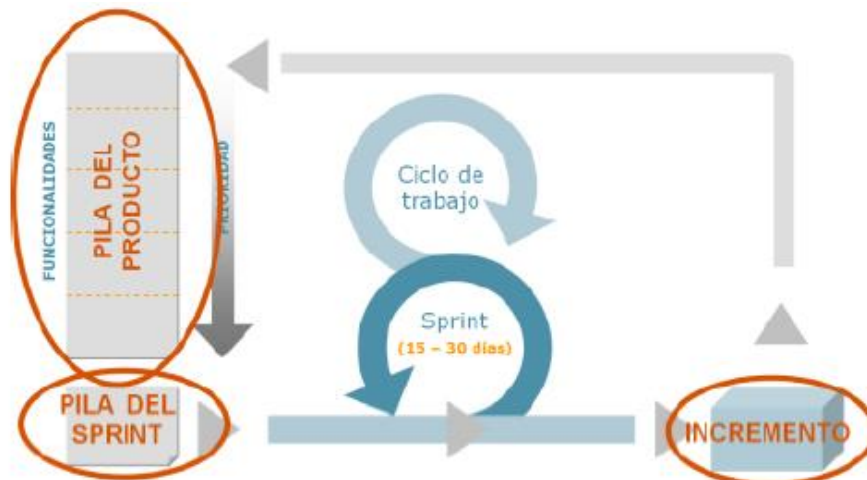


Figura 1 Elementos de SCRUM

Este tema describe estos tres elementos. Los dos primeros forman los requisitos del sistema, y el tercero es valor que se le entrega al cliente al final de cada iteración.

Cada incremento es una parte del producto completamente terminada y operativa.

No se deben considerar como incrementos: prototipos, módulos o subrutinas pendientes de pruebas o de integración.

1.3. Requisitos en el desarrollo ágil

La ingeniería de software clásica diferencia dos áreas de requisitos:

1. Requisitos del sistema
2. Requisitos del software

Los requisitos del sistema forman parte del proceso de adquisición, y por tanto es responsabilidad del cliente la definición del problema y de las funcionalidades que debe aportar la solución. En la Figura 2 se muestran los ámbitos de los requisitos.

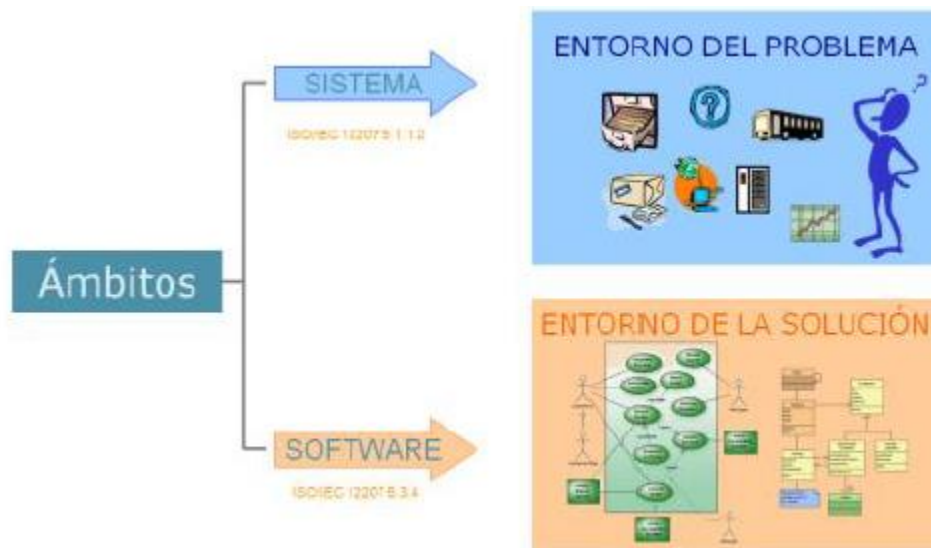


Figura 2 Ámbitos de los requisitos

No importa si se trata de gestión tradicional o ágil. La descripción del sistema es responsabilidad del cliente, aunque se aborda de forma diferente en cada caso como se muestra en la Figura 3 los requisitos de gestión productiva vs requisitos de la gestión ágil:

- En los requisitos del sistema suelen especificarse en documentos formales; mientras que en los proyectos ágiles toman la forma de pila del producto o lista de historias de usuario.

- Los requisitos del sistema formales se especifican de forma completa y cerrada al inicio del proyecto; sin embargo una pila del producto es un documento vivo, que evoluciona durante el desarrollo.
- Los requisitos del sistema los desarrolla una persona o equipo especializado en ingeniería de requisitos a través del proceso de obtención con el cliente. En Scrum la visión del cliente es conocida por todo el equipo y la pila del producto se realiza y evoluciona de forma continua con los aportes de todo el equipo.

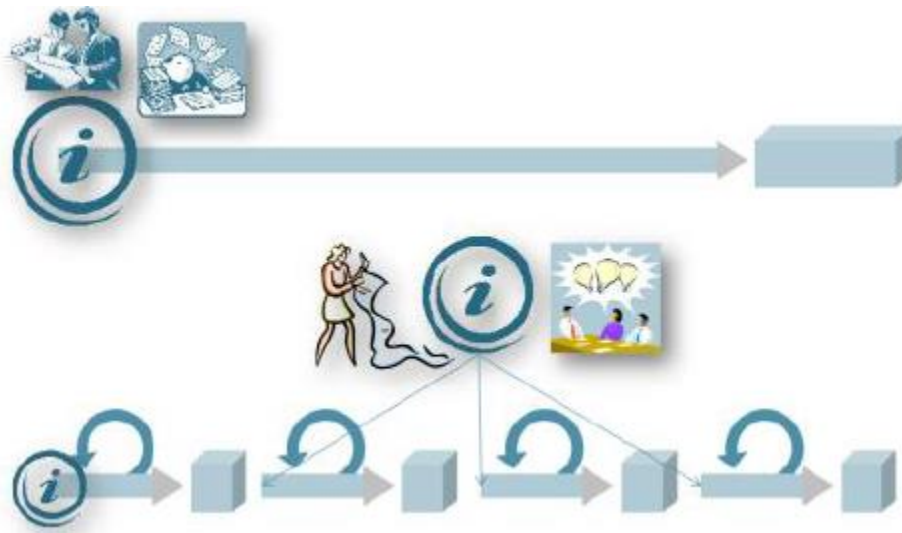


Figura 3 Requisitos de la gestión predictiva vs requisitos de la gestión ágil.

Pero la responsabilidad es del cliente, del “propietario del producto” en el caso de Scrum, que debe decidir que se incluye en la pila del producto, y el orden de prioridad.

1.3.1. Requisitos y visión del producto

Scrum, aplicado al software, emplea dos formatos para registrar los requisitos:

1. Pila del producto (Product Backlog).
2. Pila del sprint (Sprint Backlog).

La pila del producto se sitúa en el área de necesidades del negocio desde el punto de vista del cliente. Es el área en que la ingeniería del software tradicional cubre los requisitos del sistema.

La pila del sprint cubre la especificación de los requisitos de software necesarios para dar respuesta a las funcionalidades esperadas por el cliente.

Estas listas no tienen por qué cumplir con un determinado “formato scrum-estándar”. Pueden, y deben, aceptar la forma más adecuada al sistema. En la Figura 4 se muestra como es el desarrollo ágil y el predictivo.

Requisitos del Sistema (pila del producto):

- Las funcionalidades que aportan dan forma a una visión del producto definida y dada a conocer a todo el equipo.
- Las funcionalidades están individualmente definidas, organizadas por prioridades y pre-estimadas.
- Están realizados y gestionados por el cliente.

Requisitos del software (pila del sprint):

- Tienen todas las tareas necesidades para realizar el incremento de un sprint.
- El equipo ha estimado el esfuerzo de cada tarea.
- El equipo ha asignado cada tarea a un miembro
- Las duraciones estimadas de las tareas no son ni inferiores, ni superiores a los límites definidos en el equipo.

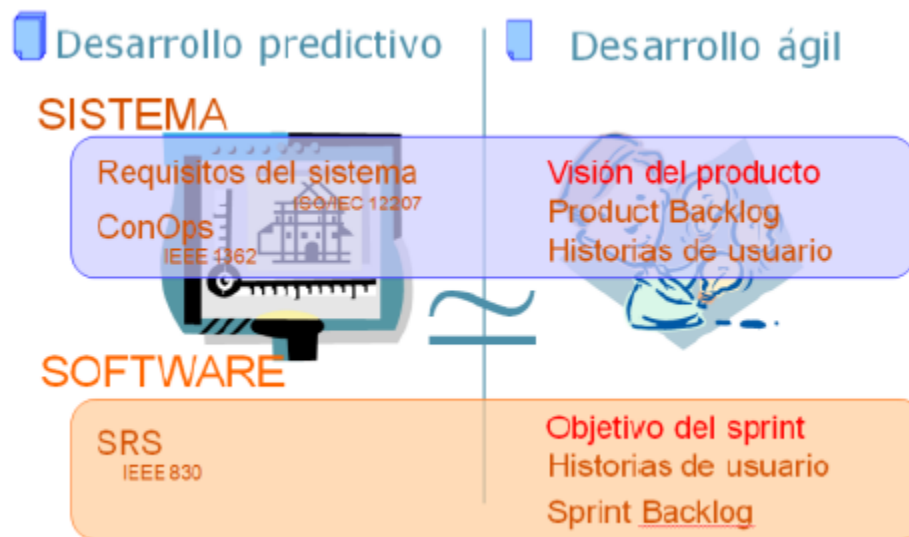


Figura 4 Descripciones de los requisitos en la gestión predictiva y la gestión ágil.

1.3.2. Pila del producto: los requisitos del cliente

La pila del producto es el conjunto de funcionalidades, mejoras, tecnología y corrección de errores que deben integrarse al producto a través de las sucesivas iteraciones de desarrollo (Juan Palacio, 1.4.0 – Enero de 2011).

Representa todo aquello que esperan los clientes, usuarios, y en general a los interesados. Todo lo que suponga un trabajo que debe elaborar el equipo tiene que estar reflejado en esta pila.

Para este estudio se conoce la situación actual en las instituciones de salud para el proceso de consulta, modificación o alta del expediente médico de cada persona. Dicho procedimiento es sumamente lento, se requiere mucha documentación y datos, no es digitalizado, y no todas las personas tienen acceso al documento del expediente, además de que el tiempo de consultas médicas es demorado o no tan certero por la falta de agilidad y conocimiento sobre los pacientes en el diagnóstico. Por lo antes mencionado se han planteado los siguientes requerimientos:

- Permitir a los usuarios la consulta de su expediente médico.
- Permitir a los usuarios exportar el expediente en un archivo PDF.
- Permitir a los usuarios registrarse en el sistema.
- Permitir al usuario administrador dar de alta un expediente médico.
- Permitir al usuario administrador buscar un expediente médico.
- Permitir al usuario administrador modificar el contenido de un expediente médico.
- Reducir tiempo de una consulta médica.
- Permitir la consulta desde una terminal móvil con sistema operativo Android.
- Permitir el registro de usuarios desde una terminal móvil con sistema operativo Android.

A diferencia de un documento de requisitos del sistema, la pila del producto nunca se da por cerrada; está en continuo crecimiento y evolución.

Para empezar el desarrollo se necesita una visión de los objetivos de negocio que se quieren alcanzar con el proyecto, una vez entendida y conocida por todo el equipo, y elementos suficientes en la pila para llevar a cabo el primer sprint.

1.3.3. Formato de la pila del producto

La pila del producto no es un documento de requisitos, sino una herramienta de referencia para el equipo. Si se emplea formato de lista, es recomendable que al menos incluya la siguiente información en cada línea:

- Identificador único de la funcionalidad o trabajo.

- Descripción de la funcionalidad.
- Campo o sistema de prioridad
- Estimación

Es recomendable no adoptar ningún modelo a seguir de trabajo de forma rígida. El formato del product backlog no es cerrado. Los resultados de Scrum Management no dependen de la rigidez en la aplicación del modelo a seguir, sino de sus principios e implementación en un formato adecuado a las características del cliente y del proyecto.

1.3.4. Pila del Sprint

La pila del sprint, (sprint backlog en inglés) es la lista que descompone las funcionalidades de la pila del producto en las tareas necesarias para construir un incremento: una parte completa y operativa del producto.

Es útil porque divide el proyecto en unidades de tamaño adecuado para determinar el avance a diario, e identificar riesgos y problemas sin necesidad de procesos complejos de gestión. Es también una herramienta de soporte para la comunicación directa del equipo.

1.3.5. El incremento

El incremento es la parte de producto generada en un sprint, y tiene como características: que está completamente terminada y operativa, en condiciones de ser entregada al cliente final. No se trata por tanto de módulos o partes a falta de pruebas, o documentación, etc...

Idealmente en el desarrollo ágil:

- Cada funcionalidad de la pila del producto se refiere a funcionalidades entregables, no a trabajos internas del tipo “diseño de la base de datos”.
- Se genera un “incremento” en cada iteración.

Si el proyecto o el sistema requiere documentación, o procesos de validación y verificación documentados, o con niveles de independencia que implican procesos con terceros éstos también tienen que estar realizados para considerar que el producto esta “terminado”.

1.4. Plataformas de desarrollo

1.4.1. Android

Existen muchas plataformas para móviles (Iphone, Symbian, Windows Phone, Blackberry, Palm, Java Mobile Edition, Linux Mobile,...); sin embargo, las características de Android lo hacen diferente ya que es el primero que combina las siguientes cualidades en una misma solución (Jésus, 2012):

- Plataforma realmente abierta. Es una plataforma de desarrollo libre basada en Linux y de código abierto.
- Portabilidad asegurada. Las aplicaciones finales son desarrolladas en Java lo que asegura que podrán ser ejecutadas en gran variedad de dispositivos, tanto presentes como futuros. Lo anterior se consigue gracias al concepto de máquina virtual.
- Arquitectura basada en componentes inspirados en internet. Por ejemplo, el diseño de la interfaz de usuario se hace en XML, lo que permite que una misma aplicación se ejecute en un móvil de pantalla reducida o en un *notebook*.
- Filosofía de dispositivo siempre conectado a Internet
- Gran Cantidad de servicios incorporados: por ejemplo, localización basada tanto en GPS como en torres de telefonía móvil. Incorpora potentes bases de datos con SQL, entre otros.

- Alto nivel de seguridad. Los programas se encuentran aislados unos de otros gracias al concepto de ejecución dentro de una caja que incorpora la máquina virtual. Cada aplicación dispone de una serie de permisos que limitan su rango de actuación (servicios de localización, acceso a internet).
- Optimización para baja potencia y poca memoria. Por ejemplo, Android utiliza la máquina virtual Dalvik. Se trata de una implementación de Google de la máquina virtual de Java optimizada para dispositivos móviles.
- Alta calidad de gráficos y sonido: gráficos vectoriales suavizados, animaciones inspiradas en Flash, gráficos en 3 dimensiones basados en OpenGL. Incorpora códecs estándar más comunes de audio y video incluyendo H.264(AVC), MP3, AAC.

Android combina características muy interesantes. Sin embargo, la pregunta que más se genera entorno a esto es la siguiente: ¿se convertirá Android en el nuevo estándar de sistema operativo (S.O) para móviles? Para contestar la pregunta, habrá que esperar un tiempo y saber cuál es la respuesta de Windows tomando en cuenta el lanzamiento de su nuevo S.O. para móviles. En pocas palabras, Android permitirá una forma sencilla y novedosa de implementar potentes aplicaciones para móviles.

1.4.2 Java

Por otro lado, Existe el lenguaje de programación JAVA, el cual cumple con ciertos puntos muy útiles y que benefician el desarrollo:

- Lenguaje Simple: Es fácil adquirir el conocimiento sobre el uso del lenguaje Java. Todos aquellos usuarios familiarizados con C++ encontrarán que Java es más sencillo, ya que se han eliminado ciertas características, tales como los punteros. Debido a su semejanza con C y C++, y dado que son extensamente conocidos aunque sea de forma elemental, resulta muy fácil aprender Java. Los programadores experimentados en C++ pueden migrar muy rápidamente a Java y ser productivos en poco tiempo.

- **Orientado a Objetos:** Java fue diseñado como un lenguaje orientado a objetos desde el inicio de su creación. Los objetos agrupan, en estructuras encapsuladas, tanto sus datos como los métodos (o funciones) que manipulan dichos datos. La tendencia de Java apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red.
- **Distribuido:** Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, lo que facilita, así, la creación de aplicaciones distribuidas.
- **Interpretado y compilado a la vez:** Java es compilado en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador. Por otra parte dicho código es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina en la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (run-time).
- **Robusto:** Java fue diseñado para crear software altamente fiable. Para ello, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Las características de memoria de Java liberan a los programadores de cometer posibles errores comunes (la aritmética de punteros), puesto que se ha prescindido por completo los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.
- **Indiferente a la Arquitectura.** Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows Nt, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. Para reunir requisitos de ejecución, el compilador de Java genera bytecodes: un formato intermedio indiferente a la arquitectura diseñada para transportar el código eficientemente a múltiples plataformas de hardware y software. El resto de los problemas, los soluciona el intérprete de Java.
- **Alto Rendimiento:** Hoy en día ya se ven muy limitadas las aplicaciones que sólo pueden ejecutar una acción a la vez.

- MultiHilo: Java soporta sincronización de múltiples hilos de ejecución (multithreading) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario; al mismo tiempo, otro presenta una animación en pantalla y otro realiza cálculos.
- Produce Applets: Java puede ser usado para crear dos tipos de programas: aplicaciones independientes y applets. Las aplicaciones independientes se comportan como cualquier otro programa escrito en cualquier lenguaje, como por ejemplo el navegador de Web HotJava, escrito íntegramente en Java. Por su parte, las applets son pequeños programas que aparecen dentro de las páginas Web, tal como se muestran los gráficos o los textos, pero con la capacidad de ejecutar acciones muy complejas: como animar imágenes, establecer conexiones de red, presentar menús y cuadros de diálogo entre otras.

1.4.3. Los Inicios de Android y Java

Google adquiere Android Inc. en el año 2005. Se trataba de una pequeña compañía, recién creada y orientada a la producción de aplicaciones para terminales móviles. En ese mismo año Android Inc. empezó a trabajar en la creación de una máquina virtual Java optimizada para móviles (Dalvik VM).

En el año 2007 se crea el consorcio Handset Alliance ([http](http://www.handsetalliance.com)) con el objetivo de desarrollar estándares abiertos para móviles. Está formado por Google, Intel, Texas Instruments, Motorola, T-Mobile, Samsung, Ericsson, Toshiba, Vodafone, NTT DoCoMo, Sprint Nextel y otros. Una pieza clave de los objetivos de esta alianza es promover el diseño y difusión de la plataforma Android.

En noviembre del mismo año se lanza una primera versión del Android SDK (Jésus, 2012). Al año siguiente aparece el primer móvil con Android (T-Mobile G1). En octubre Google libera el código fuente de Android bajo licencia de código abierto Apache (licencia GPL v2 para el núcleo) principalmente. En ese mismo mes se abre Android Market, para la descarga de

aplicaciones. En abril del año 2009 Google lanza la versión 1.5 del SDK que incorpora nuevas características como el teclado en pantalla. A finales del año se lanza la versión 2.0; y durante el 2010, las versiones 2.1, 2.2 y 2.3.

Durante el año 2010, Android se consolida como uno de los sistemas operativos para móviles más utilizados, con una calidad cercana al Iphone e incluso superando al sistema de Apple en EE.UU.

Por otro lado Java inicia siendo una herramienta de programación para ser usada en un proyecto de set-top-box en una pequeña operación denominada "The Green Project" (Eckel) en Sun Microsystems en el año 1991. El equipo (Green Team), compuesto por trece personas y dirigido por James Gosling, trabajó durante 18 meses en el desarrollo.

El lenguaje se denominó inicialmente Oak (por un roble que había fuera de la oficina de Gosling), luego pasó a denominarse Green tras descubrir que Oak era ya una marca comercial registrada para adaptadores de tarjetas gráficas; y finalmente, se renombró a Java, sin aclarar si es un acrónimo o no, aunque algunas fuentes señalan que podría tratarse de las iniciales de sus creadores: **J**ames Gosling, **A**rthur **V**an Hoff, y **A**ndy Bechtolsheim. Otros abogan por el siguiente acrónimo, **J**ust **A**nother **V**ague **A**cronym ("sólo otro acrónimo ambiguo"). La hipótesis que más respaldo tiene afirma que Java debe su nombre a un tipo de café disponible en una cafetería cercana, de ahí que el icono de java sea una taza de café caliente. Un pequeño signo que da fuerza a esta teoría es que los 4 primeros bytes (el número mágico) de los archivos.class que genera el compilador son en hexadecimal, 0xCAFEBABE. A pesar de todas estas teorías, el nombre fue sacado al parecer de una lista aleatoria de palabras.

1.4.4. Android VS Otras Plataformas

En este apartado se describirán las características principales de las diversas plataformas móviles disponibles en la actualidad. Dado la gran cantidad de datos que se indican, se ha utilizado la Tabla 1 para representar la información. De esta forma resulta más sencillo comparar las plataformas.

Tabla 1 Comparativa de las principales plataformas móviles.



symbian

**Apple
iOS5.1**

Android 4.0

**Windows
7**

**Phone
7**

**BlackBerry
7**

OS

Symbian 9.5

	Apple iOS5.1	Android 4.0	Windows 7	Phone 7	BlackBerry 7	OS	Symbian 9.5
Licencia de software	Propietaria	Software libre y abierto	Propietaria		Propietaria		Software libre
Año de Lanzamiento	2007	2008	2010		2003		1997
Motor del navegador web	Web Kit	Web Kit	Pocket Explorer	Internet	Web Kit		Web Kit
Soporte Flash	No	Si	No		Si		Si
HTML 5	Si	Si	Parcial		Si		No
Tienda de aplicaciones	App Store	Google Play	Windows Marketplace		BlackBerry World	App	Ovi Store
Numero de aplicaciones	400,000	300,000	50,000		30,000		50,000
Coste publicar	\$99 al año	\$25 una vez	\$99 al año		Sin coste		\$1 una vez
Plataforma de desarrollo	Mac	Windows, Mac, Linux	Windows		Windows, Mac		Windows, Mac, Linux
Interfaz personalizable	No	Si	Si		Si		Si
Actualizaciones automáticas del S.O.	Si	Depende del fabricante	Depende del fabricante		Si		Si
Soporte memoria externa	No	Si	No		Si		Si
Fabricante único	Si	No	No		Si		No
Variedad de dispositivos	Modelo único	Muy alta	Baja		Baja		Muy alta
Tipo de pantalla	Capacitativa	Capacitativa /resistiva	Capacitativa		Capacitativa /resistiva		Capacitativa /resistiva
Aplicaciones nativas	Si	Si	No		No		Si

En cuanto a las bases de datos las plataformas de desarrollo fueron las siguientes:

1. MySQL
2. SQLite

1.5 MySQL

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009 desarrolla MySQL como software libre (Gilfillan). Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo.

MySQL está disponible para múltiples plataformas, la seleccionada para los ejemplos de este estudio fue Java. Sin embargo, las diferencias con cualquier otra plataforma son nulas, ya que la herramienta utilizada en este caso es el Phpmyadmin que permite interactuar con un servidor MySQL (local o remoto) en modo gráfico. De este modo es posible realizar todos los ejercicios sobre un servidor instalado localmente o, a través de Internet, sobre un servidor remoto.

MySQL es un SGBD que ha ganado popularidad por una serie de atractivas características tales como las que se mencionan a continuación:

- Está desarrollado en C/C++.
- Se distribuyen ejecutables para cerca de diecinueve plataformas diferentes.
- La API se encuentra disponible en C, C++, Eiffel, Java, Perl, PHP, Python,

Ruby y TCL.

- Está optimizado para equipos de múltiples procesadores.
- Es muy destacable su velocidad de respuesta.
- Se puede utilizar como cliente-servidor o incrustado en aplicaciones.
- Cuenta con un rico conjunto de tipos de datos.
- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el SGBD a cada caso concreto.
- Su administración se basa en usuarios y privilegios.
- Se tiene constancia de casos en los que maneja cincuenta millones de registros, sesenta mil tablas y cinco millones de columnas.
- Sus opciones de conectividad abarcan TCP/IP, *sockets* UNIX y *sockets* NT, además de soportar completamente ODBC.
- Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.
- Es altamente confiable en cuanto a estabilidad se refiere

MySQL tiene como principal objetivo ser una base de datos fiable y eficiente. Ninguna característica es implementada en MySQL si antes no se tiene la certeza que funcionará con la mejor velocidad de respuesta y, por supuesto, sin causar problemas de estabilidad.

1.5.1 Limitaciones

Así como MySQL es muy buena opción para la gestión tiene algunas carencias que conforme a las versiones van mejorando cada vez más para evitarlas, a continuación se mencionan algunas de las cuales se corregirían en la versión 5.0:

- No soporta procedimientos almacenados (se incluirán en la próxima versión 5.0).
- Lento con grandes bases de datos
- Utilidades no documentadas

1.6. SQLite

SQLite es uno de los motores de base de datos de más rápido crecimiento en todo, pero eso es el crecimiento en términos de popularidad, nada tiene que ver con su tamaño. Una de las mayores fortalezas de SQLite es que es extremadamente ligero y a pesar de ello tienes varias características como rendimiento, tamaño, portabilidad, estabilidad y costo.

- Rendimiento. Realiza operaciones de base de datos de manera eficiente y es más rápido que otras bases de datos libres, tales como MySQL y PostgreSQL.
- Tamaño. Tiene una pequeña huella de memoria y solo se requiere una única biblioteca para acceder a bases de datos, por lo que es ideal para aplicaciones de base de datos integrada como lo es en el caso de este estudio para la consulta de los expedientes médicos.
- Portabilidad. Funciona en muchas plataformas y sus bases de datos se pueden trasladar fácilmente sin configuración cliente/servidor o administración continua.
- Estabilidad. Es compatible con ACID(Atomicity, Consistency, Isolation and Durability)
 - Atomicidad: es la propiedad que se asegura de que la operación se ha realizado o no, y por consiguiente ante una falla del sistema no puede quedar a mitad de proceso.
 - Consistencia: es la propiedad que se asegura que solo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper las reglas y directrices de integridad de base de datos.

- Aislamiento: es la propiedad que se asegura que una operación no pueda afectar otras. Esto promete que la realización de dos transacciones sobre la misma información sean independientes y no generen ningún conflicto
- Durabilidad: es la propiedad que se asegura que una vez realizada la operación, esta persistirá y no se podrá deshacer aunque falle el sistema.
- Costo SQLite es de dominio público y por lo tanto es libre de utilizar para cualquier propósito sin el costo y puede ser redistribuido libremente.

Por las características mencionadas anteriormente el desarrollo de la aplicación para móviles se consideró que es rápida y de casi nulo tamaño, dadas las características de los dispositivos móviles en cuanto a hardware y software es ideal utilizar esta herramienta para el desarrollo de la base de datos con SQLite.

CAPÍTULO 2.

DISEÑO Y ANALISIS DEL SISTEMA

En esta sección se proporciona información acerca de los lugares donde se podría desarrollar el proyecto, el estado actual de operación y la propuesta computacional para el SCEML.

Las funciones de las instituciones de salud que se podrían ver apoyadas con el SCEML son sustantivas para el área de expedientes y consultas en general, en el aseguramiento de calidad, rapidez y eficiencia. Abarca desde una consulta general, especializada hasta el registro de un expediente médico.

La creación de este software se sustenta en la necesidad de optimizar la generación, consulta y accesibilidad de los expedientes médicos. Por esta razón la propuesta computacional contempla lo siguiente:

- Establecimiento de requerimientos en base a la metodología Scrum
- Diagrama de flujo de información
- Diagrama de flujo de procesos
- Modelo de Casos de Uso
- Modelo de Base de Datos
- Modelo de Clases
- Modelo de Secuencia

2.1. Establecimiento de requerimientos

Hace algunos años en las instituciones de salud tanto públicas como privadas, la cantidad de expedientes que se manejaban sin el uso de sistemas computacionales era enorme y se cometían errores tanto en el ámbito médico como en el administrativo en el procesamiento de la información.

Este sistema se generó gracias al conocimiento que se tienen en los procesos de:

- Alta: Para un nuevo registro de expediente se necesita en primera instancia es generar una cita, siguiendo de un diagnóstico por parte de algún médico y llenar la información del expediente a mano.
- Modificación: Para la modificación se tiene que solicitar por escrito, esto toma tiempo en lo que se busca y se realiza el cambio.
- Consulta: Para la consulta se trata de manera directa con el Médico (si es que se tiene expediente) identificándolo por el número de folio o afiliación.

El primer paso en el desarrollo del modelo de empresa, se identificó las siguientes entidades:

- Usuario: Sujeto que recibe los servicios de una consulta médica, en las diferentes instituciones de salud identificado por un número único llamado folio o número de afiliación el cual lo diferencia dentro de toda la población dentro de las distintas instituciones de salud o sujeto que presta los servicios de una consulta médica (Administrador) que puede dar de alta un expediente y o modificar alguno existente.
- Expediente: Es una entidad la cual maneja toda la información relacionada al folio o número de afiliación que pertenece al usuario. En base a cuestionarios se fijó la información mínima necesaria para este estudio que contendrá dicho expediente.

Los problemas de trabajo actuales en las instituciones de salud es que la gestión de los expedientes se realiza de forma manual. Para la recopilación de las necesidades se utilizaron las propuestas por Scrum.

La propuesta de Scrum incluye inicialmente una etapa la cual consta de elaborar la pila del producto (requerimientos del cliente) siguiendo un formato no forzoso que contenga datos esenciales para esta. Para dicha pila de producto se utilizaron cuestionarios con los cuales se obtuvieron los requerimientos del sistema. Una vez analizados los cuestionarios se plantearon las siguientes necesidades tanto para la plataforma Java como para la plataforma Android:

Requerimientos plataforma java

1. Registrarse nuevo usuario.
2. Log in de acceso al sistema
3. Consultar el expediente médico por medio del número de afiliación (folio).
4. Exportar el expediente médico en formato PDF para su impresión.
5. Usuario administrador pueda dar de alta un expediente médico.
6. Usuario administrador pueda buscar un expediente médico.
7. Usuario administrador pueda editar un expediente médico.

Requerimientos plataforma Android

1. Registrar nuevo usuario.
2. Log in de acceso al sistema.
3. Consultar el expediente médico del usuario.

En la Tabla 2 se muestra la pila de producto que es una representación fácil de comprender de los procesos que se desarrollan actualmente en las instituciones de salud. La pila de producto contiene:

- Identificador único de la funcionalidad o trabajo.
- Descripción de la funcionalidad.
- Campo o sistema de prioridad
- Estimación

Tabla 2 Ejemplo de una pila de producto

ID	PRIORIDAD	DESCRIPCIÓN	ESTIMACIÓN(DIAS)
1	Muy Alta	Plataforma tecnológica para el desarrollo del sistema	2
2	Muy Alta	Interfaz de usuario	2
3	Muy Alta	Un usuario se registra en el sistema	7
4	Alta	El usuario ingresa al sistema y consulta su expediente	7
5	Muy Alta	El administrador define el flujo y textos de un expediente	7
6	Alta	XXXXXXXX	999

2.1.1. Principales Problemas

Las herramientas empleadas para la obtención de los requerimientos nos arrojan un panorama de los problemas que existen, a continuación en la Tabla 3, se listan en dos

columnas los problemas. En la primera columna aquellos que se pueden resolver a través de la construcción del sistema de software propuesto (SCEML) a las instituciones de salud.

En la segunda columna se listan los problemas que no hallarán solución directa con SCEML. Sin embargo es importante observar, que aunque estos problemas salen del ámbito, del SCEML, pueden verse aliviados de manera indirecta.

Tabla 3 Problemas a los que da solución el SCEML

Problemas que sí pueden ser resueltos por el SCEML	Problemas que salen del ámbito del SCEML
<ul style="list-style-type: none"> • Alta inversión requerida para la reproducción de documentos • Alto número de horas-hombre requeridas para la elaboración de informes • Alto número de horas-hombre requeridas para la captura y procesamiento de formatos e informes escritos • Inconsistencia en la formación, esto es que los datos varían en función de la fuente de consulta • Fragmentación de la información para la elaboración de informes • Dificultad en la consulta de datos para la toma de decisiones oportunas 	<ul style="list-style-type: none"> • Falta o mala comunicación entre las personas • Falta de responsabilidad en la entrega de informes por parte del personal de las instituciones públicas • Conflictos personales entre médicos y pacientes

2.2. Análisis del Sistema

En esta sección se mostraran los diagramas del flujo de información y del flujo de proceso. Estos diagramas ayudan en el análisis de los requerimientos así como para definir los problemas y las soluciones que se tienen que dar.

2.2.1. Diagrama de flujo de información

El diagrama de flujo de información es una forma de ver a manera general y a grandes rasgos un proceso. Cada paso contiene brevemente una descripción de la etapa del proceso. Los símbolos de dicho diagrama son como cualquier otro diagrama de flujo unidos por flechas que indican hacia donde se dirige el proceso.

En la Figura 5 se muestra el diagrama de flujo de información general de los procesos de este estudio. La Figura 6 se muestra el diagrama del proceso de registro de usuario y por último en la Figura 7 se muestra el proceso de registro de expediente en el sistema.

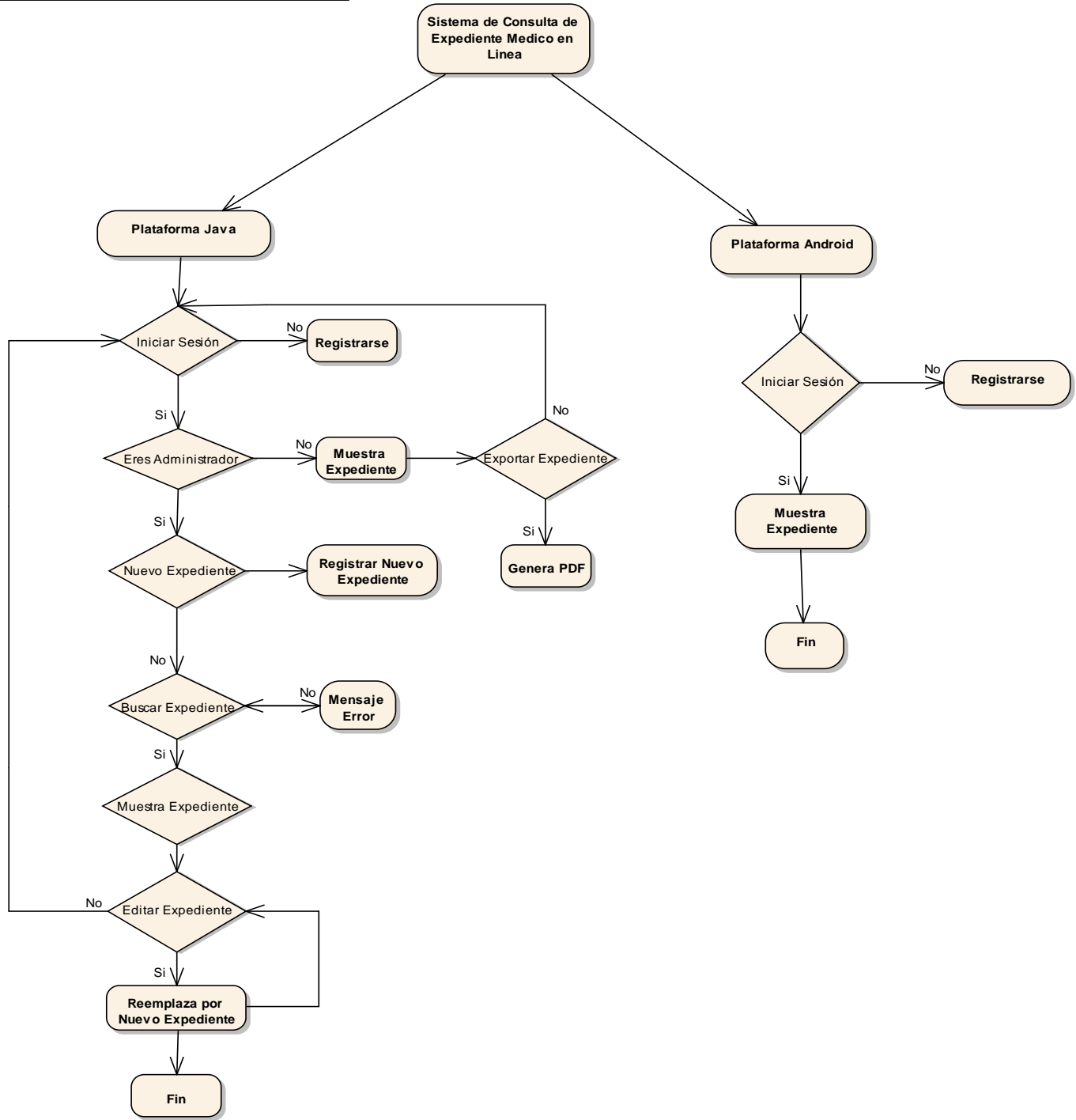


Figura 5 Diagrama de flujo general para el SCEML

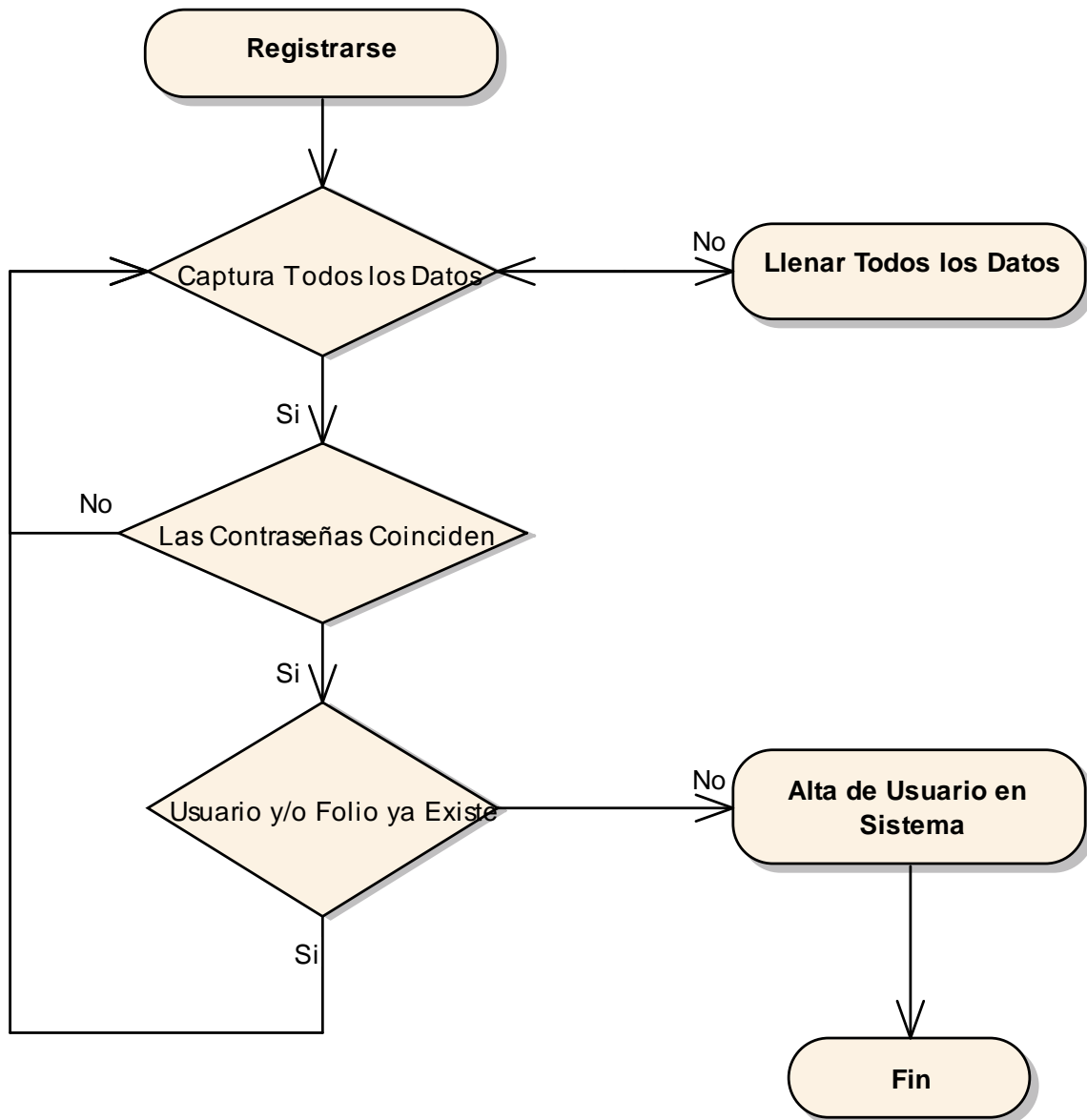


Figura 6 Diagrama de flujo del proceso registrarse

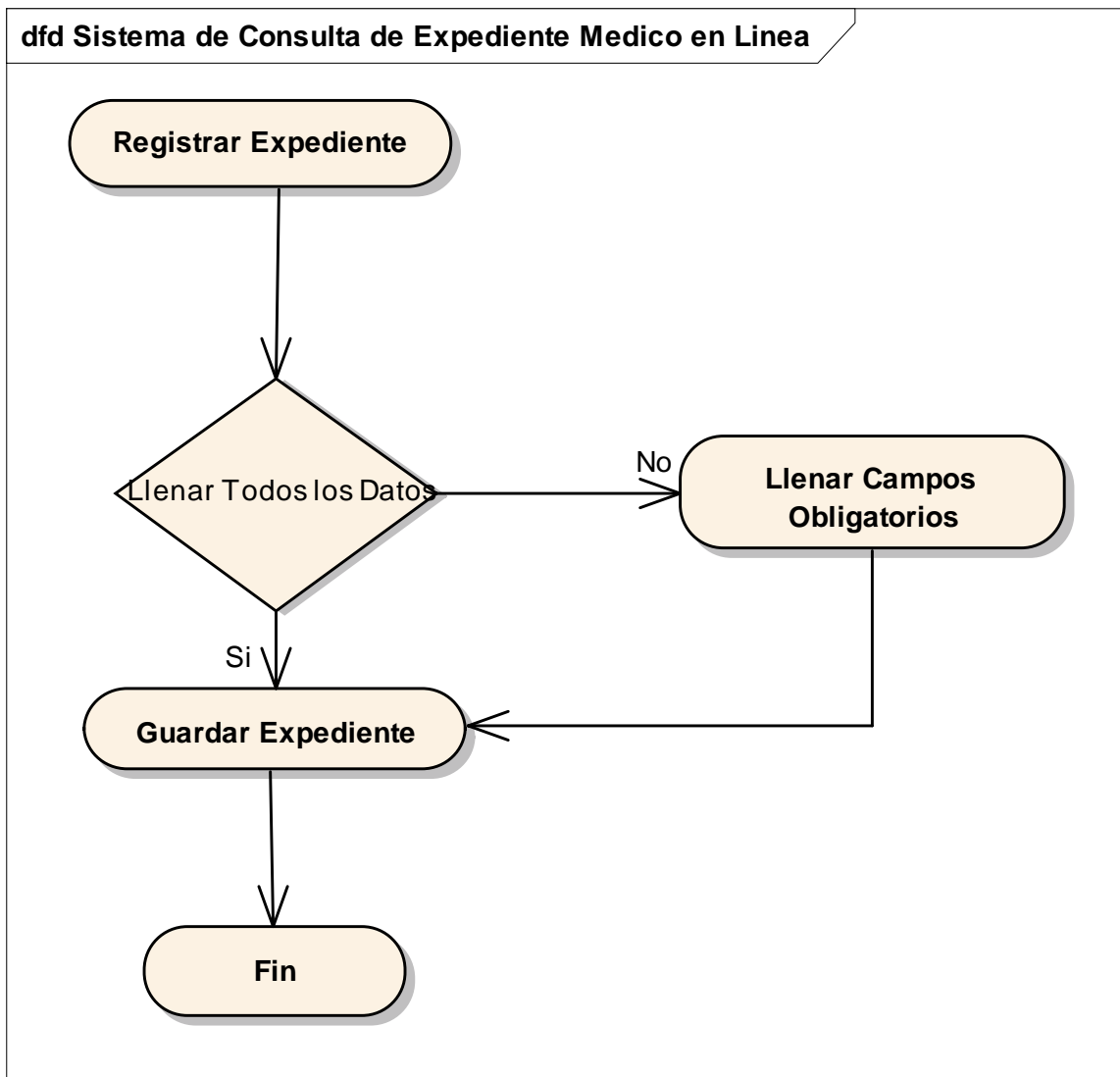


Figura 7 Diagrama de flujo del proceso registrar expediente

2.2.2. Diagrama de flujo de procesos

El diagrama de flujo es una forma de poder ver gráficamente un proceso. Cada paso del proceso es representado por un símbolo diferente que contiene una descripción más a detalle en comparación al diagrama de información de la etapa de proceso. Como se mencionó los símbolos del flujo del proceso están unidos entre sí con flechas que indican hacia dónde se dirige el proceso.

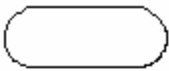


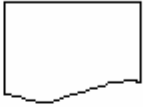
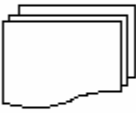


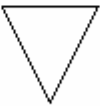


El diagrama de flujo sirve para tener una descripción visual de las actividades implicadas en el proceso mostrando la relación secuencial entre ellas, facilitando la rápida comprensión de cada actividad y su relación con las demás, el flujo de la información y los materiales, las ramas en el proceso, la aparición de bucles repetitivos entre otras.

2.2.2.1. Ventajas del Diagrama de Flujo

- Facilita la obtención de una visión transparente del proceso, mejorando su comprensión. El conjunto de actividades, relaciones e incidencias de un proceso.
- Permiten definir los límites de un proceso. A veces estos límites no son tan evidentes, no estando definidos los distintos proveedores y clientes (internos y externos) involucrados.
- El diagrama de flujo facilita la identificación de los clientes, es más sencillo determinar sus necesidades y ajustar el proceso hacia la satisfacción de sus necesidades y expectativas.
- Estimula el pensamiento analítico en el momento de estudiar un proceso haciendo más factible generar alternativas útiles.
- Proporciona un método de comunicación más eficaz, al introducir un lenguaje común, si bien es cierto que para ello se hace preciso la capacitación de aquellas personas que entrarán en contacto con el diagrama.
- Constituyen el punto de comienzo indispensable para acciones de mejora o reingeniería.

En la Tabla 4 la simbología utilizada para la elaboración de diagramas de flujo.

Tabla 4 Simbología de un diagrama de flujo

SÍMBOLO	REPRESENTA	SÍMBOLO	REPRESENTA
	Terminal: Indica el inicio o la terminación del flujo del proceso.		Actividad: Representa una actividad llevada a cabo en el proceso.
	Decisión: Indica un punto en el flujo en que se produce una bifurcación del tipo "SI" – "NO"		Documento: Se refiere a un documento utilizado en el proceso, se utilice, se genere o salga del proceso.
	Multidocumento: Refiere a un conjunto de documentos. Un ejemplo es un expediente que agrupa a distintos documentos.		Inspección / Firma: Empleado para aquellas acciones que requieren una supervisión (como una firma o "visto bueno").
	Conector de proceso: Conexión o enlace con otro proceso diferente, en la que continúa el diagrama de flujo.		Archivo Manual: Se utiliza para reflejar la acción de archivo de un documento y/o expediente.
	Base de datos/aplicación: Empleado para representar la grabación de datos.		Línea de Flujo. Proporciona indicación sobre el sentido de flujo del proceso.

Para la elaboración del diagrama de flujo de proceso debe de ser realizado por un equipo de trabajo en el que las distintas personas aporten, en conjunto, una perspectiva completa del proceso. También deben ser tomados en cuenta los siguientes puntos

1. Determinar el proceso que será representado en el diagrama
2. Definir el grado de detalle. El diagrama de flujo del proceso puede mostrar a grandes rasgos la información sobre el flujo general de actividades principales, o ser desarrollado de modo que se incluyan todas las actividades y los puntos de decisión. Un diagrama de flujo detallado dará la oportunidad de llevar a cabo un análisis más exhausto del proceso.
3. Identificar la secuencia de pasos del proceso. Situándolos en el orden en que son llevado a cabo.

4. Construir el diagrama de flujo. Para ello se utilizan determinados símbolos. Cada organización puede definir su propio grupo de símbolos. En la Tabla 4 se muestra un conjunto de símbolos habitualmente utilizados. Al respecto cabe decir que en esta misma tabla “Conector de proceso” es frecuentemente utilizado un círculo como símbolo. Para la elaboración de un diagrama de flujo, los símbolos estándar han sido normalizados, entre otros, el American National Standards Institute (ANSI).
5. Revisar el diagrama de flujo del proceso.

En las Figuras 8-10 se muestran los diagramas realizados para el SCEML utilizando toda la información hasta ahora recabada.

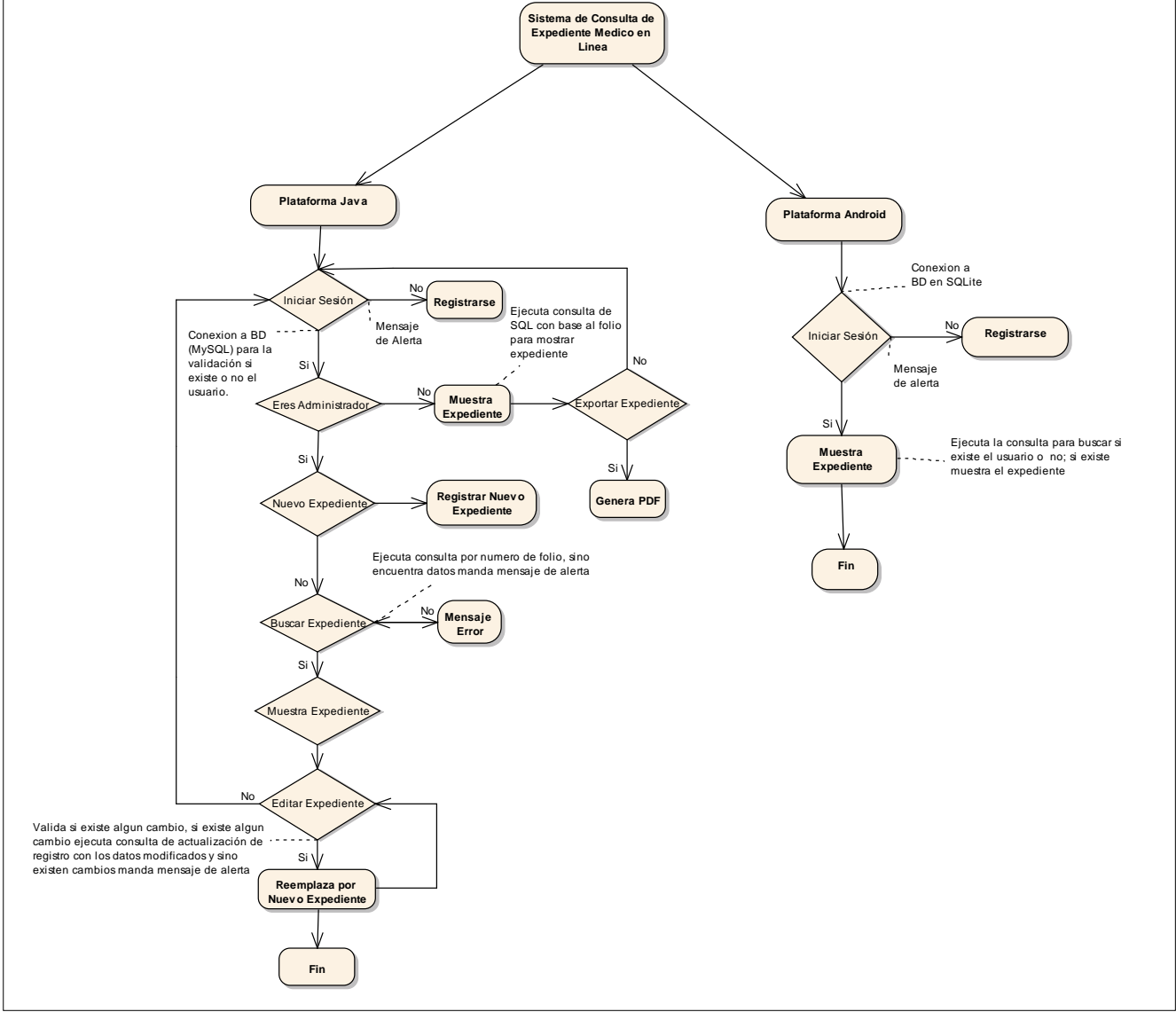


Figura 8 Flujo principal del SCEML

dfd Sistema de Consulta de Expediente Medico en Linea

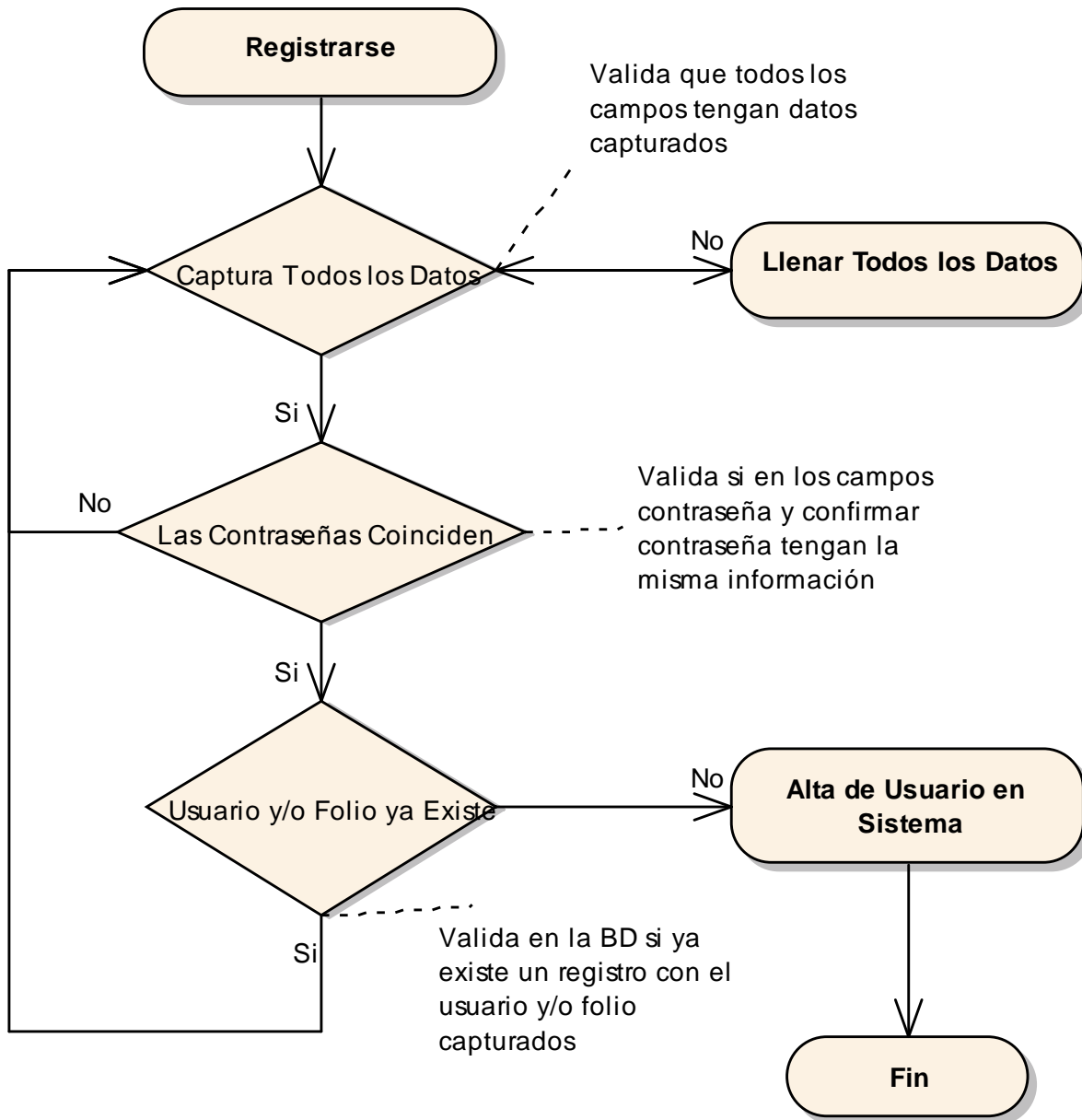


Figura 9 Diagrama de flujo de procesos registrarse

dfd Sistema de Consulta de Expediente Medico en Linea

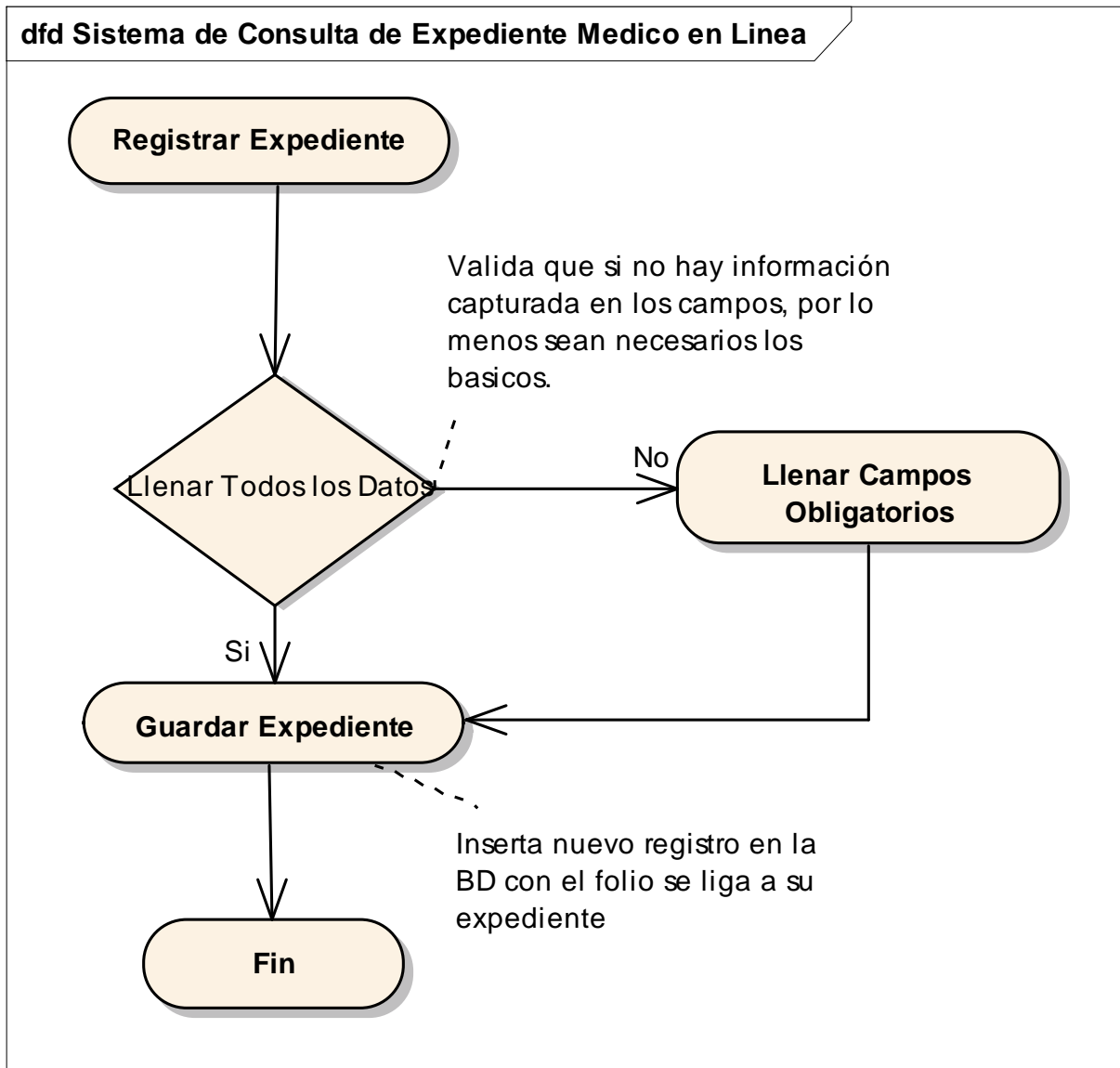


Figura 10 Diagrama de flujo de proceso de registrar expediente

2.3. Diseño del sistema

El diseño del sistema se encarga de desarrollar el seguimiento de las propuestas durante el análisis en términos de la configuración que tenga más posibilidades de satisfacer los objetivos planteados desde el punto de vista funcional así como del no funcional.

Dentro del diseño de sistemas se debe de tomar en cuenta los cambios que puedan presentarse a la introducción del nuevo sistema sobre el ambiente en el que deba de funcionar, acondicionando a los criterios del diseño del mismo. En este contexto es de gran importancia el adaptar todo el sistema-producto a las capacidades de los usuarios que van a utilizar el sistema, de manera que su funcionamiento sea fácil, cómodo, efectivo y eficiente.

El diseño del sistema se basó en los siguientes 4 puntos:

1. Modelo de base de datos
2. Modelo de casos de uso
3. Modelo de clases
4. Modelo de secuencias

2.3.1. Modelo de base de datos

Las bases de datos no son más que una colección de archivos, son una fuente central de datos destinados a compartirse entre muchos usuarios para diversas aplicaciones. El núcleo de una base de datos lo forma el sistema de administración de la base de datos (DBMS, database management system), el cual permite la creación modificación y actualización de la base de datos, la recuperación de datos y la generación de objetivos se conoce como administrador de base de datos.

Entre los objetivos de efectividad de la base de datos los siguientes:

- Asegurar que los datos se puedan compartir entre los usuarios para una diversidad de aplicaciones.
- Mantener los datos que sean exactos y consistentes.
- Asegurar que se podrá acceder con facilidad a todos los datos requeridos por las aplicaciones actuales y futuras.

- Permitir a la base de datos evolucionar conforme aumenten las necesidades de los usuarios.
- Permitir a los usuarios construir su vista personal de los datos sin preocuparse por la forma en que los datos se encuentren almacenados físicamente.

Con base en lo mencionado anteriormente se realizaron los siguientes modelos de base de datos:

- Modelo de base de datos relacional
- Modelo de base de datos entidad-relación

2.3.1.1. Modelo de datos entidad-relación

El modelo de entidad-relación es un modelo de datos basado en una percepción del mundo real, el cual consta de un grupo de objetos básicos denominados entidades y relaciones entre estos objetos, implementándose de forma gráfica a través del diagrama de entidad-relación (Guillermo Storti, 2007).

Una entidad se puede definir como cualquier objeto, real o abstracto, que existe en un contexto definido o puede llegar a existir y del cual se necesite guardar información además de que ésta tiene atributos que son características o propiedades asociadas a la entidad que toman valor en una instancia particular.

Se entiende por relación a la asociación entre 2 o más entidades, en este contexto tenemos que existen diferentes tipos de relaciones y son (Guillermo Storti, 2007):

- Relación Uno a Uno (1:1). Cuando un registro de una tabla solo puede estar relacionado con un único registro de la otra tabla y viceversa.
- Relación Uno a Muchos (1: N). Cuando un registro de una tabla (tabla secundaria) sólo puede estar relacionado con un único registro de la otra tabla (tabla principal) y un registro de la tabla principal puede tener más de un registro relacionado en la tabla secundaria.

- Relación Muchos a Muchos (N: M). Cuando un registro de una tabla puede estar relacionado con más de un registro de la otra tabla y viceversa. En este caso las dos tablas no pueden estar relacionadas directamente, se tiene que añadir una tabla entre las dos (Tabla débil o de vinculación) que incluya los pares de valores relacionados entre sí.

El nombre de tabla débil deviene que con sus atributos propios no se puede encontrar la clave, por estar asociada a otra entidad. La clave de esta tabla se conforma por la unión de los campos claves de las tablas que relaciona.

Teniendo en cuenta lo antes descrito se muestra en la Figura 11 el modelo de entidad-relación para el SCEML.

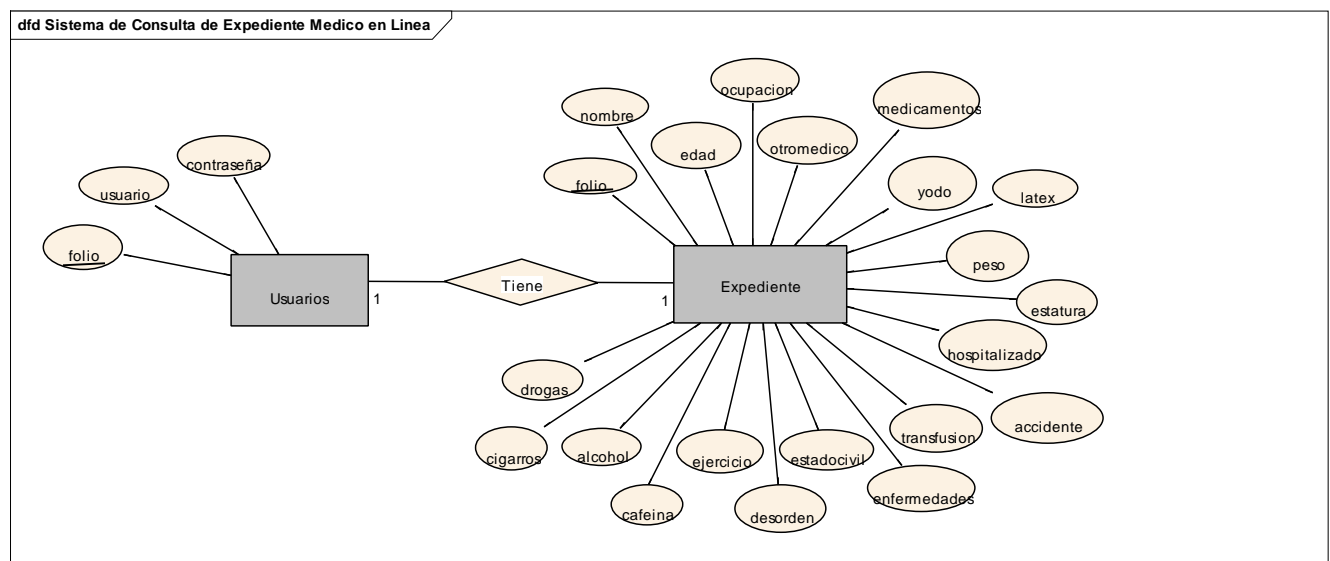


Figura 11 Modelo entidad – relación SCEML

2.3.1.2. Modelo de base de datos relacional

Este es el modelo utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. La idea fundamental es el uso de “relaciones”, las cuales podrían considerarse en forma lógica como conjunto de datos llamados “tuplas”. Se puede conceptualizar de manera más sencilla, esto es, pensando en cada relación como si fuese

una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

Teniendo en cuenta dicha descripción en la Figura 12 se muestra el modelo de datos relacional con respecto al SCEML.

Usuarios

<u>Folio</u>	usuario	contraseña
--------------	---------	------------

Expediente

<u>folio</u>	nombre	edad	ocupación	otromedico	medicamentos	yodo	latex
--------------	--------	------	-----------	------------	--------------	------	-------

peso	estatura	hospitalizado	accidente	transfusión	enfermedades	estadocivil	desorden
------	----------	---------------	-----------	-------------	--------------	-------------	----------

Ejercicio	Cafeína	Alcohol	Cigarros	Drogas	Frecuenciasexo	Lubricante	Dolorsexo
-----------	---------	---------	----------	--------	----------------	------------	-----------

anticonceptivo

Figura 12 Modelo Relacional del SCEML

2.3.2. Modelo de Casos de Uso

El caso de uso es un poderoso concepto que ayuda a un analista a comprender la forma en que un sistema deberá comportarse. Le ayuda a obtener los requerimientos desde el punto de vista del usuario (Schmuller).

Una finalidad del proceso de análisis de un sistema es generar un conjunto de casos de uso. La idea es tener la posibilidad de catalogar y hacer referencia a este conjunto, que sirve como el punto de vista de los usuarios acerca del sistema.

2.3.2.1. Representación de un modelo de caso de uso

La representación gráfica es directa. Una elipse representa a un caso de uso, una figura agregada representa a un actor. El actor que inicia se encuentra a la izquierda del caso de uso, así mismo el nombre aparece justo debajo de él, y el nombre del caso de uso aparece ya sea dentro de la elipse o justo debajo de ella. Una línea asociativa conecta a un actor con el caso de uso, y representa la comunicación entre el actor y el caso de uso.

2.3.2.2. Concepción de las relaciones entre casos de uso

Los casos de usos se pueden relacionar entre sí. Una de ellas, la inclusión, le permite volver a utilizar los pasos de un caso de uso dentro de otro. La otra, extensión, le permite crear un caso de uso mediante la adición de pasos a uno existente.

Existen otros dos tipos de relaciones que son generalización y agrupamiento. La generalización cuenta con un caso de uso que se hereda de otro. El agrupamiento es una manera sencilla de organizar los casos de uso.

2.3.2.2.1. Inclusión

Para la representación de la inclusión se utiliza una línea discontinua con una punta en la flecha que conecta las clases apuntando hacia la clase dependiente. Justo sobre la línea, agregará un estereotipo: la palabra “incluir” bordeada por dos pares de paréntesis angulares. Se tiene en cuenta que un caso de uso incluido nunca aparecerá solo. Simplemente funciona como parte de un caso de uso.

2.3.2.2. Exclusión

La extensión solo se puede realizar en puntos indicados de manera específica dentro de la secuencia del caso de uso base. A estos puntos se les conoce como puntos de extensión. Como la inclusión, se puede concebir la extensión con una línea de dependencia, junto con un estereotipo que muestra “extender” entre paréntesis angulares.

La Figura 13 y 14 se muestra el modelo de casos de uso como usuario y como administrador para el SCEML.

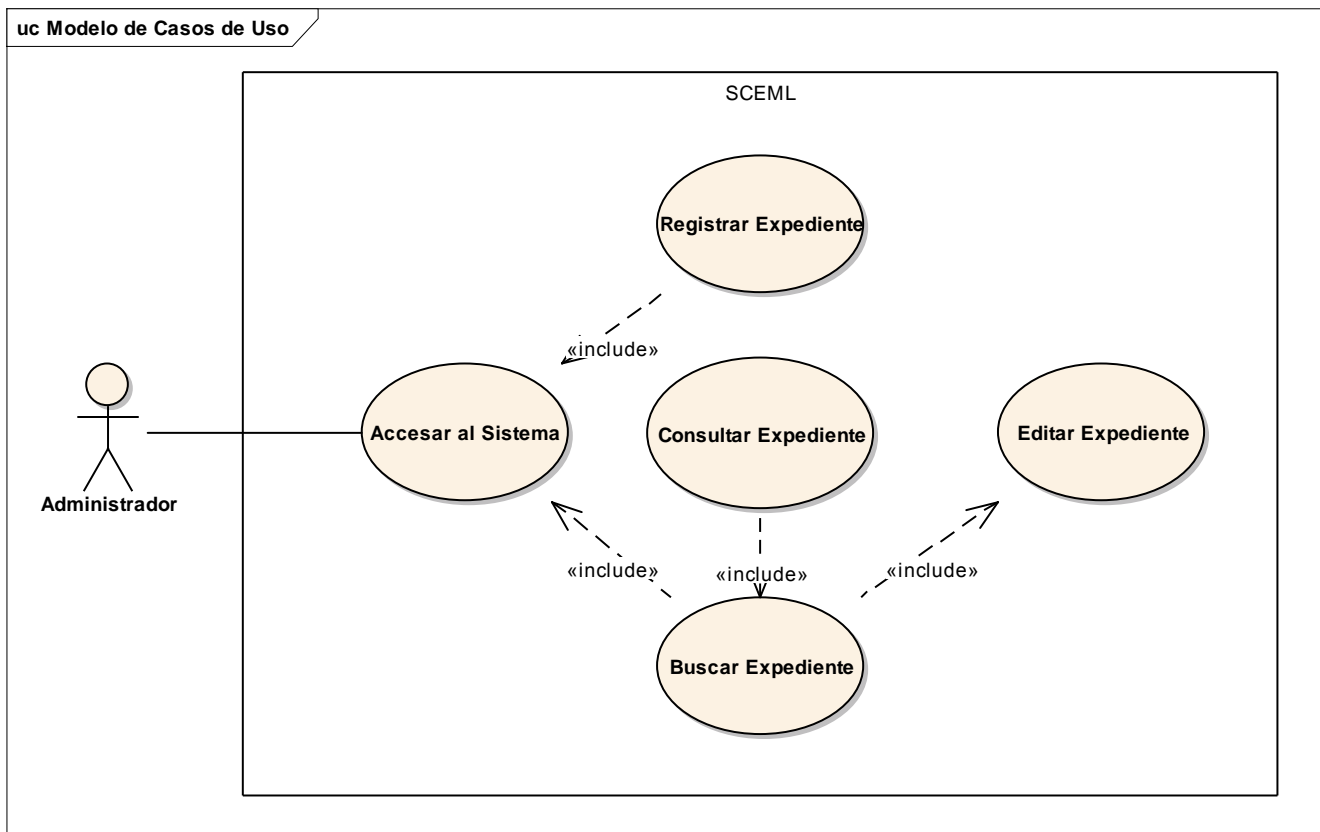


Figura 13 Diagrama de Casos de Uso para el Administrador del SCEML

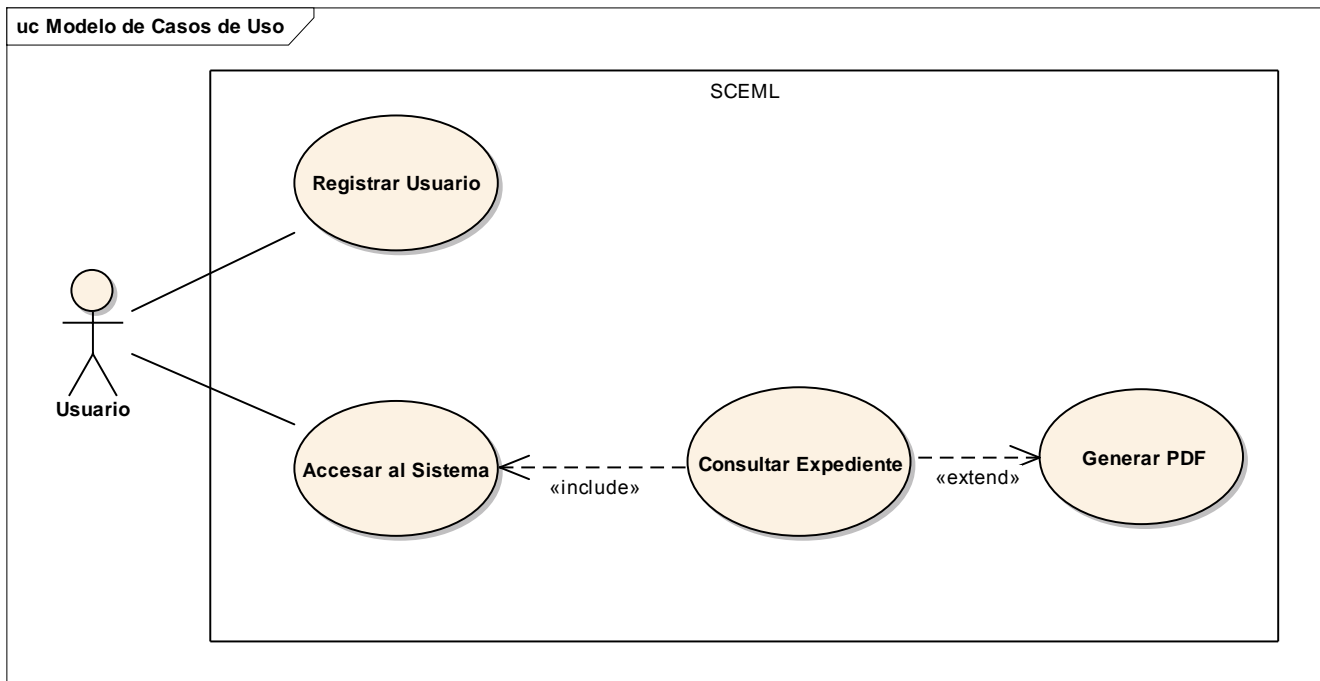


Figura 14 Diagrama de Casos de Uso para el Usuario del SCEML

Cabe mencionar diagrama de caso de uso para la plataforma Android solo cambiaría el de usuario ya que como es únicamente consulta, tomaríamos en cuenta el diagrama para usuario sin el caso de uso Generar PDF.

2.3.3. Modelo de Clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenido (Schmuller).

Un diagrama de clases está compuesto por los siguientes elementos:

- **Clase:** es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio. En UML, una clase es representada por un rectángulo que posee tres divisiones como ese muestra en la Figura 15.

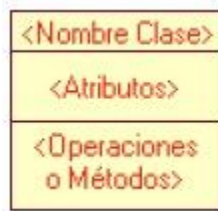


Figura 15 Ejemplo de Clase

- Nombre Clase: Contiene el nombre de la clase.
 - Atributos: Contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, public o protected).
 - Métodos: Los métodos u operaciones de una clase son la forma en como esta interactúa con su entorno, estos pueden tener las características de public, private y protected.
- **Relaciones:** En UML la cardinalidad de las relaciones indica el grado y el nivel de dependencia, se anotan en cada extremo de la relación y estas pueden ser: uno o muchos (1...n), cero o muchos (0...n) y número fijo (m que denota el numero).
 - Herencia: Indica que una subclase hereda los métodos y atributos especificados por una Súper Clase, por ende la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la Súper Clase (public y protected).
 - Agregación:
 - Por Valor: Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido está condicionado por el tiempo de vida del que lo incluye. Este tipo de relación es comúnmente llamada Composición.
 - Por Referencia: Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye. Este tipo de relación es comúnmente llamada Agregación (el objeto base utiliza al incluido para su funcionamiento).
 - Asociación: permite asociar objetos que colaboran entre sí. Cabe destacar que no es una relación fuerte, es decir, el tiempo de vida de un objeto no depende del otro.

- Dependencia o Instanciación: Representa un tipo de relación muy particular, en la que una clase es instancia (su instanciación es dependiente de otro objeto/clase). Se denota por una flecha punteada.

En las Figuras 16 y 17 se muestra el diagrama de clases tanto para Java como Android respectivamente.

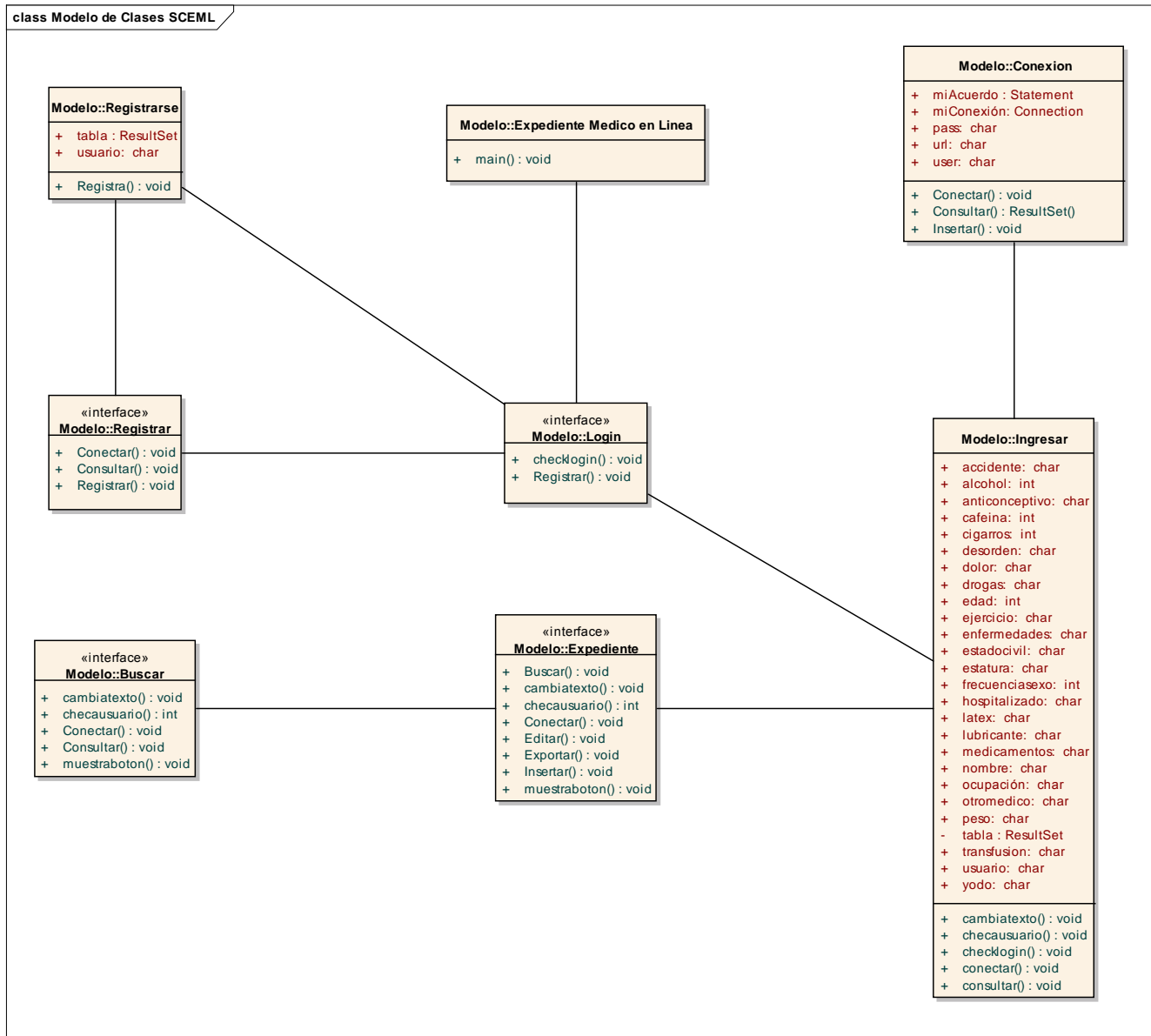


Figura 16 Modelo de Clases del SCEML

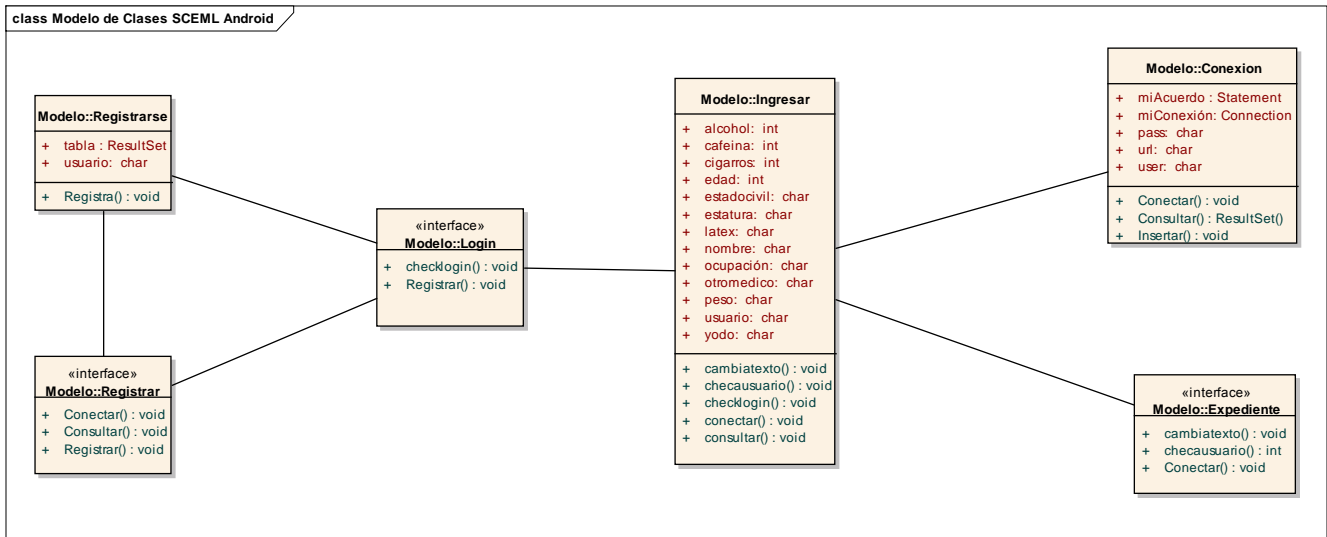


Figura 17 Diagrama de Clases para SCEML Android

2.3.4. Modelo de secuencia

El modelo de secuencia consta de objetos que se representan del modo usual: rectángulos con nombre (subrayado), mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical (Schmuller).

2.3.4.1. Objetos

Los objetos son colocados cerca de la parte superior del diagrama de izquierda a derecha y se acomodan de manera que simplifiquen al diagrama. La extensión que está debajo (y en forma descendente) de cada objeto será una línea discontinua conocida como la línea de vida de un objeto. Además de la línea de vida de un objeto se encuentra un pequeño rectángulo conocido como activación, el cual representa la ejecución de una operación que realiza el objeto. La longitud del rectángulo se interpreta como la duración de la activación.

2.3.4.2. Mensaje

Un mensaje que va de un objeto a otro pasa de la línea de vida de un objeto a la de otro. Un objeto puede enviarse un mensaje a si mismo (es decir, desde su línea de vida hacia su propia línea de vida).

Un mensaje puede ser simple, sincrónico, o asincrónico. Un mensaje simple es la transferencia del control de un objeto a otro. Si un objeto envía un mensaje sincrónico, esperará la respuesta a tal mensaje antes de continuar con su trabajo. Si un objeto envía un mensaje asincrónico, no esperará respuesta antes de continuar. En el diagrama de secuencias, los simbolos del mensaje varían, por ejemplo, la punta de la flecha de un mensaje simple está formada por dos líneas, la punta de la flecha de un mensaje simple está formada por dos líneas, la punta de la flecha de un mensaje sincrónico esta rellena y la de un asincrónico tiene una sola línea, como se aprecia en la Figura 18.

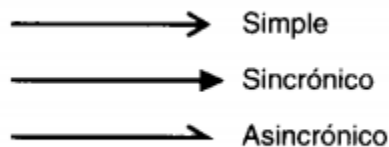


Figura 18 Tipos de mensajes en el diagrama de secuencia.

2.3.4.3. Tiempo

El diagrama representa al tiempo en dirección vertical. El tiempo se inicia en la parte superior y avanza hacia la parte inferior. Un mensaje que este mas cerca de la parte superior ocurrirá antes que uno que esté cerca de la parte inferior.

Con ello, el diagrama de secuencias tiene dos dimensiones. La dimensión horizontal es la disposición de los objetos, y la dimensión vertical muestra el paso del tiempo.

Dada la información anterior con respecto al modelado de secuencia se muestra en la Figura 19 y 20.

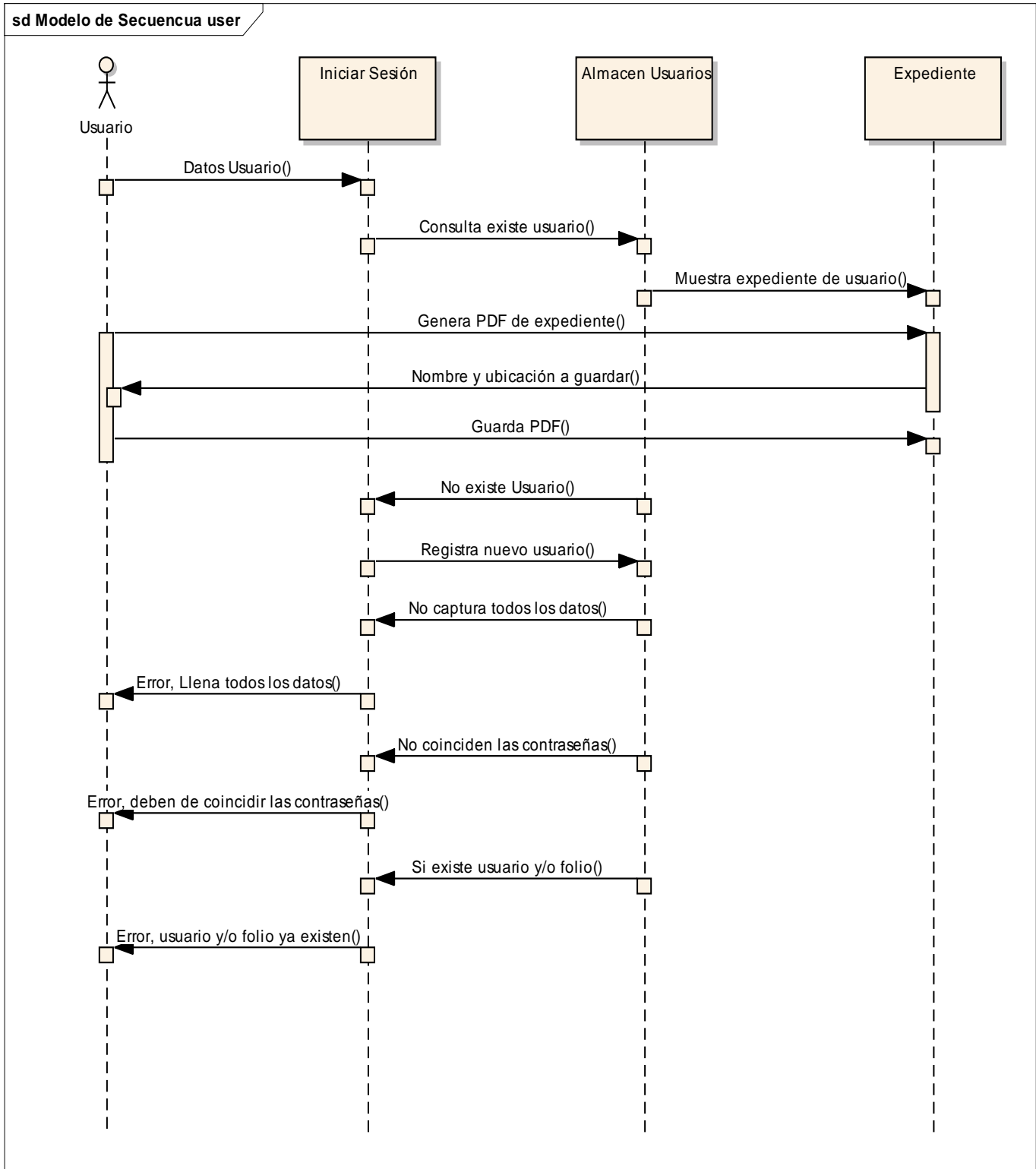


Figura 19 Diagrama de Secuencia (Usuario) SCEML.

sd Modelo de Secuencia Admin

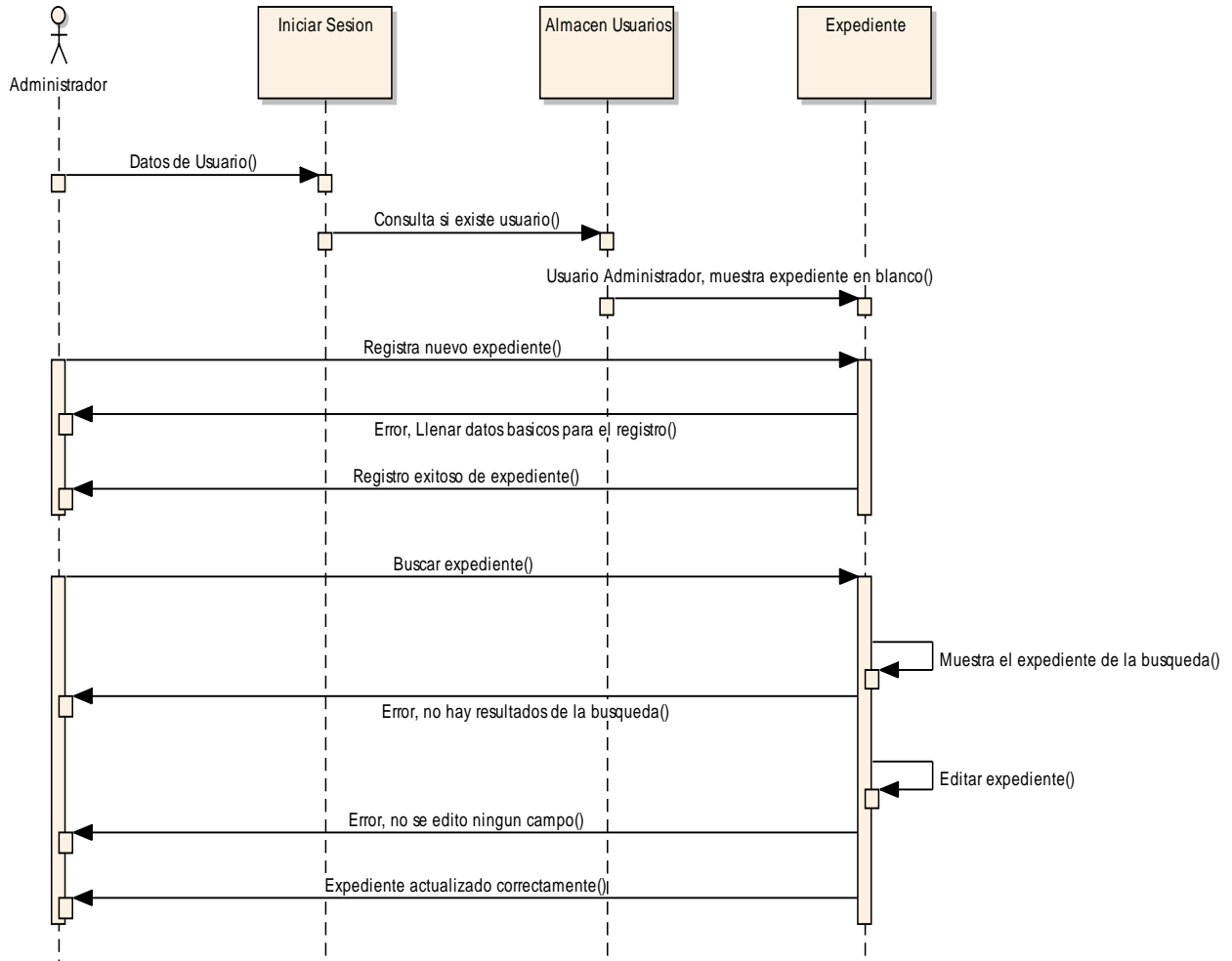


Figura 20 Diagrama de Secuencia (Administrador) SCEML.

CAPÍTULO 3.

IMPLEMENTACIÓN DEL SISTEMA

La implementación del sistema en la integración de la aplicación en una LAN, la instalación en varias computadoras personales de clientes de consultorios, hospitales, farmacias, clínicas y de administradores de las mismas. También incluye levantas la base de datos en una computadora adaptada como servidor.

3.1. Implementación de la base de datos

La base de datos contiene la información de los expedientes médicos y usuarios que se encuentren registrados en el sistema.

El primer paso en la implementación del sistema fue obtener una base de datos clara que diera a conocer el mundo real de la mejor manera; al mismo tiempo fue de gran importancia definir tablas reales para su mejor funcionamiento y área de trabajo.

3.2. Implementación de HW y SW

Se utilizó una computadora personal como servidor local con sistema operativo Linux Ubuntu, Intel Core 2 Duo a 2.20GHz con 4Gb en RAM, las computadoras en cuestión se comunican mediante una red local.

En cada una de las computadoras que estarían corriendo las aplicaciones cliente se podría instalar la versión del 7 del JDK.

En caso para la aplicación en Android es necesario únicamente tener una versión del sistema operativo Android para móviles 4.0 (IcecreamSandwich).

3.3. Implementación de la interfaz del SCEML

Al implementar el sistema en la plataforma Java lo primero que se realizó fue la implementación de las acciones del SCEML, pues el sistema plantea como objetivo principal la automatización del manejo de expedientes médicos, consulta, registro y modificación de la información en ellos.

Como se puede observar en la Figura 21 se utilizaron JFrames para la implementación de las interfaces. Las principales opciones que puede manejar el sistema dependen del usuario que se firme ya que tendrá capacidades diferentes.

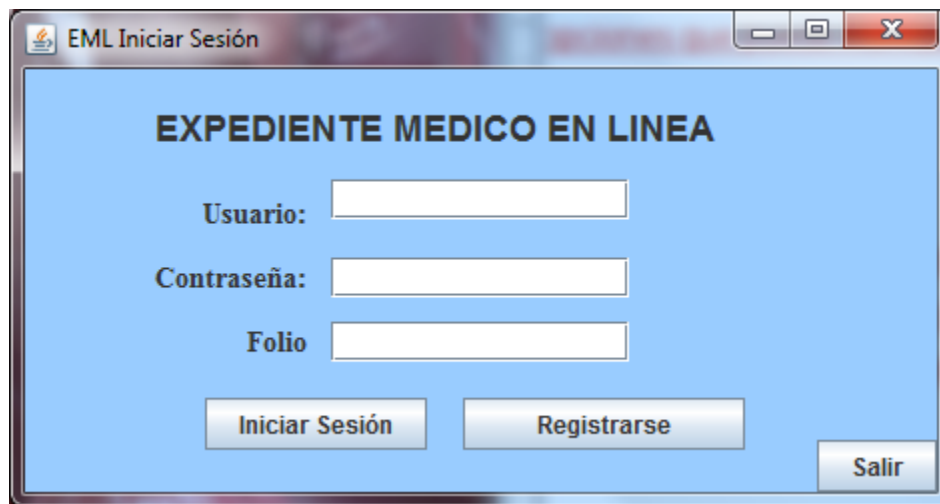


Figura 21 Interfaz de Inicio de Sesión SCEML.

En un principio la base de datos puede estar vacía, por lo que el usuario estándar solo podrá registrar su nombre de usuario y contraseña así como su folio del expediente, en cambio el usuario administrador podrá dar de alta expedientes para que el usuario pueda consultarlos sin problema, se puede observar en la Figura 22 y 23 se necesita ingresar los datos principales del usuario y expediente respectivamente.

Registro Expediente Medico En Linea

Ingrese los siguientes datos:

Nombre de Usuario:

Nueva Contraseña:

Confirmar Contraseña:

Folio:

Aceptar Regresar

Figura 22 Datos para el registro de un usuario

Expediente Medico

Bienvenido a tu Expediente

Administrador No. Folio

Información Demografica

Nombre: Edad: Ocupación:

Si te refirió algun medico? Nombre:

Alergias y Medicamentos

Tienes alguna alergia?

Medicamentos: Yodo/Tintes: Latex:

Historial Clínico

Peso Actual: Estatura Actual:

Has estado hospitalizado por embarazo o cirugia? Cirugia Cual?:

Has sufrido un accidente serio?:

Ha recibido transfusion sanguinea?:

Enfermedades que has padecido?:

Historial Sexual

Frecuencia relaciones sexuales por semana: Usas lubricante:

Dolor durante el sexo?: Anticonceptivo que mas usas:

Historial Social

Estado Civil: Has padecido un desorden alimenticio anorexia o bulimia?:

Haces ejercicio con regularidad?: Drogas en los ultimos 5 años?:

Cafeina(tazas):

Alcohol(vasos): Cigarros al dia:

Buscar Guardar Regresar

Figura 23 Datos para el registro de un expediente medico

La conexión a la base de datos se genero con las siguientes lineas de codigo:

- Se declaran las variables publicas las cuales son los parametros que se requieren para generar la conexión, cabe mencionar que se debe contar con el Driver o el Java DataBase Controller (JDBC) ya que es indispensable.

```
public Connection miConexion=null;
public Statement miAcuerdo=null;
public String user="root";
public String pass="";
public String url="jdbc:mysql://localhost/expedientemedico";
```

- Se crea una nueva instancia del controlador para las bases de datos en java.

```
Class.forName("com.mysql.jdbc.Driver").newInstance();
```

- Se utiliza la variable declarada para obtener la conexión, utilizando el controlador y el metodo *getConnection* enviando como parametros las variables predefinidas de: url que es la direccion de donde se encuentra el servidor con la base de datos, user el usuario de acceso a la base de datos y la contraseña con la que se dio de alta la base de datos.

```
miConexion=DriverManager.getConnection(url,user,pass);
```

- Se establece un acuerdo entre la ejecucion del sistema y la base de datos ya que con este podremos ejecutar las consultas que realicemos.

```
miAcuerdo=miConexion.createStatement();
```

Una vez realizada la conexión podemos hacer consultas directamente a la base de datos y manejar la informacion que nos devuelve. A continuación se muestra la consulta utilizada para la validacion de los usuarios si existen o no en la base de datos:

```
String sql = ("SELECT usuario,password FROM usuarios WHERE "
+ "usuario='"+user + "' AND password='"+pass+"' AND folio='"+folio+"'");
```

Si el resultado de la consulta tiene un resultado continuara con la siguiente consulta que es para obtener toda la informacion referente al expediente con el folio dado. El codigo es el siguiente:

```
String sql2 = ("SELECT * FROM expediente WHERE folio=\'"+folio+"\';");
```

Una vez mostrado el expediente se tiene la opción de exportar el expediente en formato PDF para tener una impresión física. Como precondition para lograr este cometido fue necesario utilizar la librería *itextpdf* ya que esta permite crear y manipular archivos PDF, RTF y HTML en Java el código para generar es el siguiente:

- Se utilizó la herramienta JFileChooser con la cual se puede abrir un cuadro de diálogo para seleccionar la ubicación y escribir el nombre que tendrá el PDF.

```
JFileChooser fc = new JFileChooser();  
int opcion = fc.showSaveDialog(this);
```

- Una vez que se muestra el cuadro de diálogo y capturamos el nombre con el que se guarda el PDF, se obtiene la ruta que se ha seleccionado con el método `getSelectedFile()`, el código es el siguiente

```
if(opcion == JFileChooser.APPROVE_OPTION){  
    File f = fc.getSelectedFile();  
    String file=f.toString();
```

- Utilizamos un flujo de salida para escribir los datos en un archivo en este caso el PDF y le mandamos

```
FileOutputStream archivo= new FileOutputStream(file+".pdf");
```

- Una vez teniendo el flujo de salida, se genera un nuevo documento con el formato del tamaño de hoja, y márgenes de escritura. También se genera la instancia para la escritura de un pdf mandando las variables del documento y el flujo de salida (archivo) y por último se deja el documento en modo abierto para escritura.

```

Document doc = new Document(PageSize.LETTER,80,80,75,75);
PdfWriter.getInstance(doc, archivo);
doc.open();

```

Lo anterior es basicamente la funcionalidad a la que tiene acceso un usuario, sin embargo existen aun mas funcionalidades que son unicamente para un administrador en el SCEML. Las proximas descripciones seran del codigo pero del lado de un administrador del sistema.

En primera instancia se tiene que, cuando se firma el administrador se le presenta un formulario en blanco con la capacidad de registrar un nuevo expediente medico. Por lo tanto, se tiene que ejecutar una consulta obteniendo la información e insertandola en la base de datos, dicha consulta es la siguiente:

```

String elsql = ("INSERT INTO expediente set folio= "+folio+" , nombre=\""+nombre+"
+ ", ocupacion=\""+ocupacion+"\", otromedico=\""+otromedico+"\", medicamento
+ ", peso=\""+peso+"\", estatura=\""+estatura+"\", hospitalizado=\""+hospi
+ ", enfermedades=\""+enfermedades+"\", estadocivil=\""+estadocivil+"\", de
+ ", alcohol=\""+alcohol+"\", cigarros=\""+cigarros+"\", drogas=\""+drogas-
+ ", anticonceptivo=\""+anticonceptivo+"\" ");

```

En caso de que el administrador por error o quisiera guardar un expediente sin datos el sistema hace una validación para que, por lo menos los datos basicos sean llenados. Para esto se genera la siguientes lineas de codigo validando si existe algo escrito en los campos de: numero de folio, nombre y ocupación

```

if(foliocad.isEmpty() || nombre.isEmpty() || ocupacion.isEmpty()) {
    JOptionPane.showMessageDialog(null, "Los campos folio, "
+ "nombre y ocupacion son obligatorios");
}

```

El siguiente punto se refiere cuando el administrador desea editar un expediente pueda buscarlo mediante el numero de folio. Entonces se genera la siguiente consulta para la busqueda:

```

String elsql = ("SELECT * FROM expediente WHERE folio=\"'+folio+'\"");

```

En caso de que no se haya editado ningún campo del expediente en la Figura 24 se muestra el mensaje del sistema.

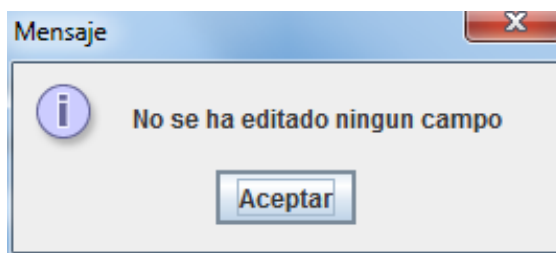


Figura 24 Mensaje sin edición de expediente

Una vez modificado en el expediente la información necesaria podemos ejecutar la acción de guardar, pero para no generar un nuevo registro en la base de datos se ejecuta la consulta UPDATE que simplemente reemplaza la información en ese mismo registro, y es la siguiente:

```
String elsql = ("UPDATE expediente set nombre=\""+nombre+"\" , edad=\""+edad+" \"\"  
+ ", ocupacion=\""+ocupacion+"\", otromedico=\""+otromedico+"\", medicament  
+ ", peso=\""+peso+"\", estatura=\""+estatura+"\", hospitalizado=\""+hospi  
+ ", enfermedades=\""+enfermedades+"\", estadocivil=\""+estadocivil+"\", de  
+ ", alcohol=\""+alcohol+"\", cigarros=\""+cigarros+"\", drogas=\""+drogas-  
+ ", anticonceptivo=\""+anticonceptivo+"\" WHERE folio="+folio+" ");
```

En cuanto al sistema o aplicación en la plataforma Android se mencionara los puntos esenciales de la aplicación como su interfaz y código. Se debe tomar en cuenta que para el desarrollo en Android solo se toma en cuenta que fuese únicamente consulta de expediente por lo que se genero una interfaz de inicio de sesión y es la siguiente que se muestra en la Figura 24.

Además se generaron otras interfaces como la de registro de usuario y en la que se muestra el expediente en las Figuras 26 y 27.

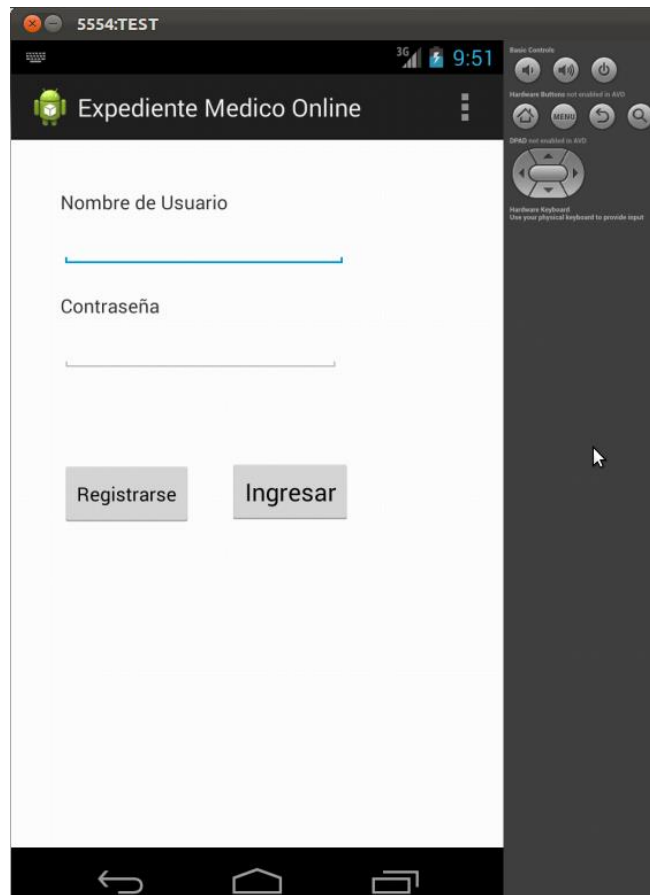


Figura 25 Interfaz de inicio de sesión en Android

En esta interfaz se puede notar que unicamente basta con el usuario y contraseña para poder consultar el expediente. Ademas, se tiene una interfaz de registro muy similar a la que se vio en la plataforma Java, dicha interfaz se observa en la Figura 25.

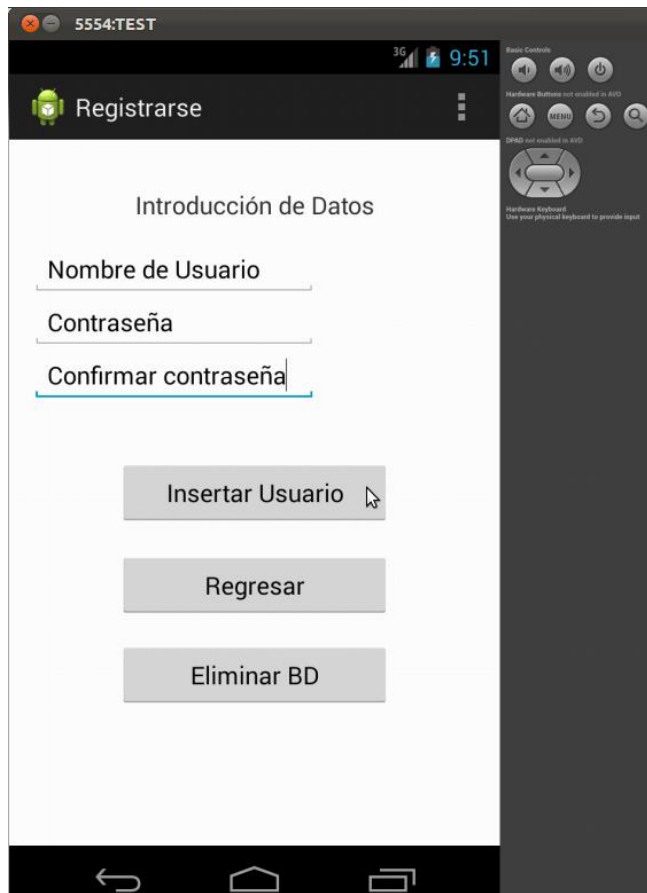


Figura 26 Interfaz de registro de usuario para consultar expedientes.

La interfaz de registro se visualiza que unicamente pide los datos minimos para poder registrarse en la aplicación y con esto tener acceso a la informacion del expediente.

Una vez que se esta registrado y pueda ingresar al expediente se mostrara la interfaz donde contiene la información de su expediente con respecto a ese usuario, y dicha interfaz es la siguiente con información basica del expediente que se muestra en la Figura 26.

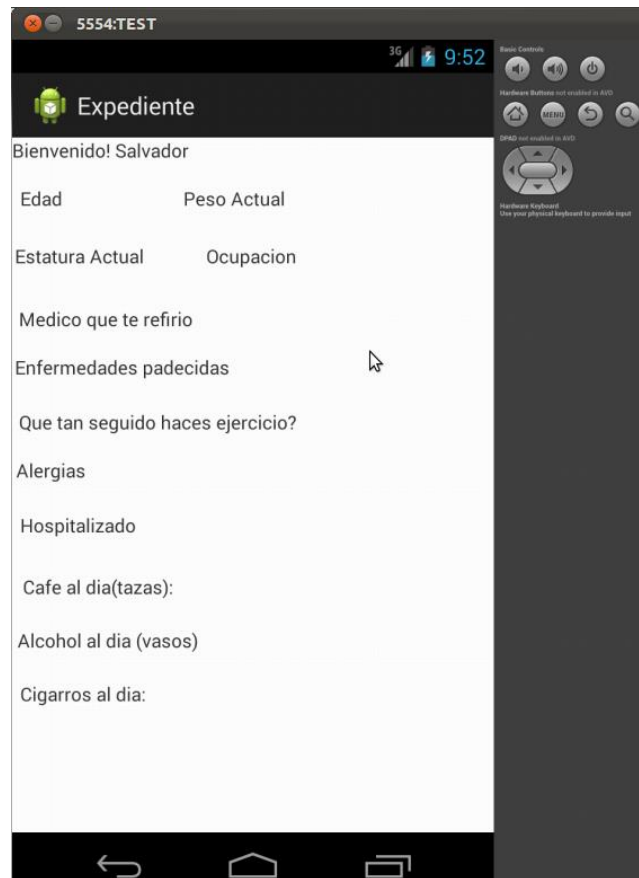


Figura 27 Interfaz del expediente en Android

Despues teniendo en cuenta las interfaces los principales puntos del codigo que son fundamentales para crear este tipo de aplicaciones son los siguientes:

- Se tiene que importar las clases de SQLite ya que estas permiten la gestion y manejo de la base de datos local para moviles, y con eso se declaran las variables que se ocuparon para esta base de datos.

```
private SQLiteDatabase baseDatos;
private static final String TAG="bdAsociados";
private static final String nombreBD="asociados";
private static final String tablaUsuario="usuario";

//guardamos la creación de la tabla en string
private static final String crearTablaUsuario = "create table if not exists "
    + "usuario (codigo integer primary key autoincrement, "
    + "nombre text not null, pass text not null unique);";
```

- En cuanto a las interfaces (layout) con el siguiente metodo se obtiene lo necesario para mostrar el layout.

```
setContentView(R.layout.activity_emo);
```

- El siguiente codigo sirve para generar la consulta con la cual se obtiene respuesta de parte de la base de datos si existe o no un registro con los datos ingresados.

```
String rconsulta=null;
try
{
    baseDatos = openOrCreateDatabase(nombreBD, MODE_WORLD_READABLE, null);
}
catch (Exception e)
{
    Log.i(TAG, "Error al abrir base de datos" + e);
}

String [] valores = {"nombre","pass"};
Cursor c = baseDatos.query("usuario", valores, "nombre='"+editN.getText().toString()+"'", null, null, null,
if(c.getCount() != 0){
c.moveToFirst();
rconsulta = c.getString(0);
}
else{
    rconsulta="VACIO";
}

return rconsulta;
```

- Una vez que se tiene la validación de los datos el siguiente paso es verificar el resultado de la consulta y si es un resultado valido se envia el valor a otra clase y con esto se lanza dicha clase junto con la actividad donde se presenta el expediente junto con sus datos.

```

if(resultado.compareTo("VACIO")==0){
    Toast.makeText(getApplicationContext(), "Usuario " + editN.getText().toStrin
}
else{
    Toast.makeText(getApplicationContext(), "Usuario " + resultado + " ya esta r

Bundle bundle = new Bundle();
bundle.putString(KEY_NAME, nombre);

Intent intent = new Intent();
intent.setClass(Emo.this, Expediente.class);
intent.putExtras(bundle);
startActivity(intent);
}

```

- Ya en la clase junto con la actividad se obtiene el valor que se envió previamente desde la validación y se asigna el texto a una etiqueta.

```

Bundle bundle = getIntent().getExtras();
String editN = bundle.getString(Emo.KEY_NAME);
if(editN.length() > 0){
    nomb.setText("Bienvenido! " + editN);
}

```

- Ahora en caso de un registro nuevo de usuario se realizan las validaciones como lo son la confirmación de contraseña y que se hayan ingresado todos los datos.

```

String ecp = editConfirmpass.getText().toString();
String ep = editPass.getText().toString();
if(ep.equals(ecp)){

```

- Ya realizadas las validaciones se continúa con el paso a insertar el nuevo registro del usuario y para eso obtenemos lo que se captura en los campos de la interfaz y los mandamos como parámetros del método insert para SQLite.

```
ContentValues values = new ContentValues();
values.put("nombre", nombre );
values.put("pass", pass);
Toast.makeText(getApplicationContext(), "Nombre: " + nombre + ", " +
    "Contraseña: " + pass, Toast.LENGTH_LONG).show();
return (baseDatos.insert(tablaUsuario,null,values) > 0);
```

Escencialmente los puntos descritos tanto para la plataforma Java como la plataforma Android son los que permiten el funcionamiento general del SCEML.

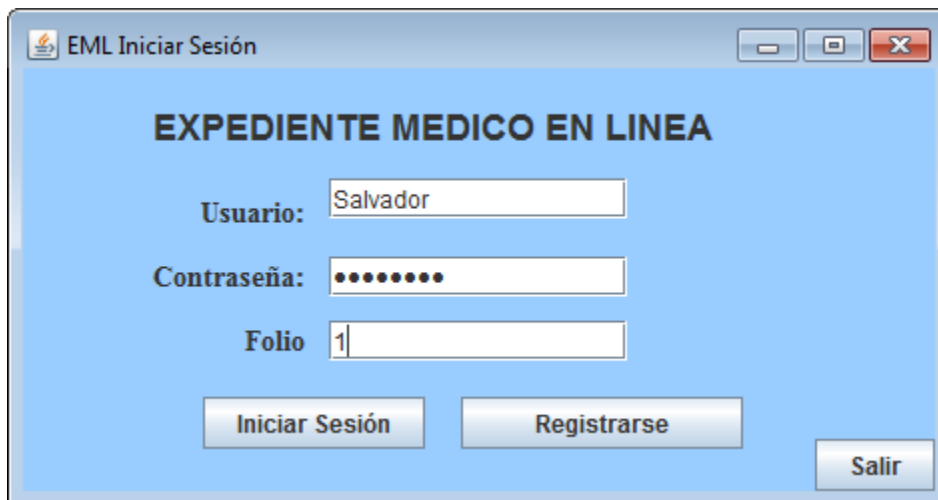
CAPÍTULO 4.

PRUEBAS Y EVALUACIÓN DEL SISTEMA

En esta sección del estudio se mostrarán imágenes del SCEML funcionando tanto en Java como en Android, a continuación veremos las imágenes de prueba del flujo principal en Java

4.1. Imágenes de prueba

- Modo Usuario
 - Interfaz inicial donde se ingresan los datos del usuario



The screenshot shows a Java Swing window titled "EML Iniciar Sesión". The window has a light blue background and contains the following elements:

- Title Bar:** "EML Iniciar Sesión" with standard minimize, maximize, and close buttons.
- Header:** "EXPEDIENTE MEDICO EN LINEA" in bold black text.
- Form Fields:**
 - Usuario:** A text input field containing "Salvador".
 - Contraseña:** A password input field with 10 dots.
 - Folio:** A text input field containing "1".
- Buttons:** Three buttons are located at the bottom: "Iniciar Sesión" (left), "Registrarse" (middle), and "Salir" (right).

- Una vez iniciada la sesion se muestra el expediente ligado al numero de folio y si aun no se ha registrado un expediente al usuario firmado se mostrará en blanco por lo que habrias que ponerse en contacto con el administrador para dar de alta el nuevo expediente

Expediente Medico de Usuario

Bienvenido a tu Expediente

Salvador

Información Demografica

Nombre: Edad: Ocupación:

Si te refirió algun medico? Nombre:

Alergias y Medicamentos

Tienes alguna alergia?

Medicamentos: Yodo/Tintes: Latex:

Historial Clinico

Peso Actual: Estatura Actual:

Has estado hospitalizado por emabarazo o cirugia? Cirugia Cual?:

Has sufrido un accidente serio?:

Ha recibido transfusion sanguinea?:

Enfermedades que has padecido?:

Historial Sexual

Frecuencia relaciones sexuales por semana: Usas lubricante:

Dolor durante el sexo?: Anticonceptivo que mas usas:

Historial Social

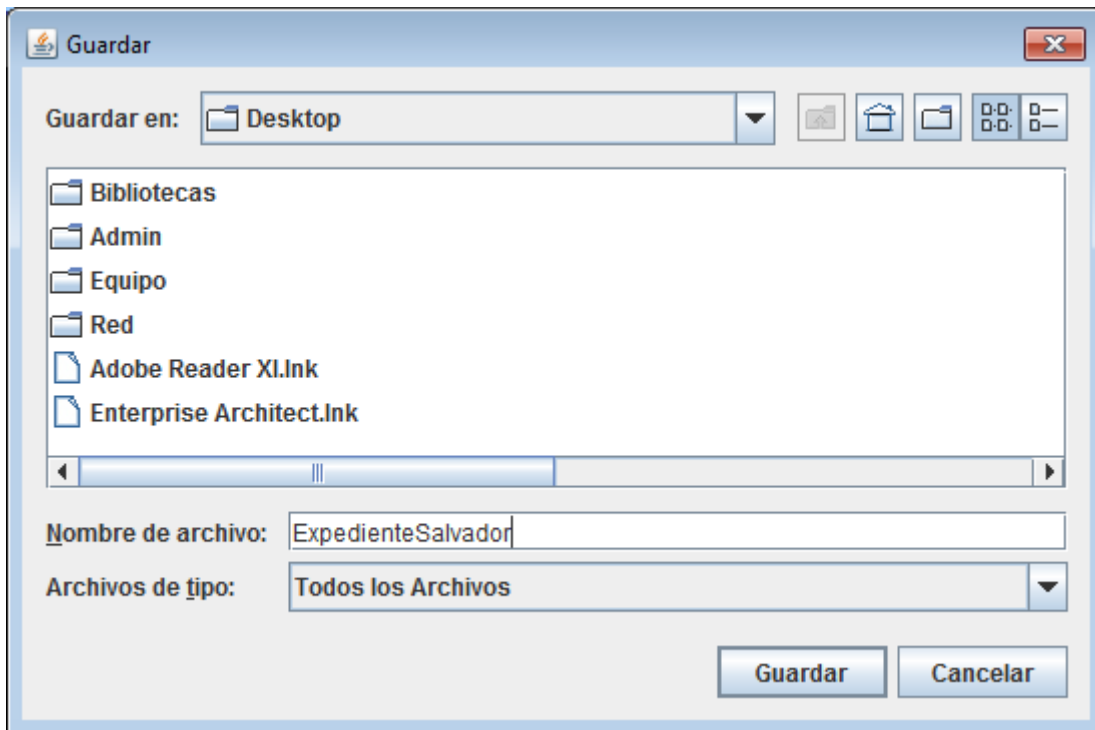
Estado Civil: Has padecido un desorden alimenticio anorexia o bulimia?:

Haces ejercicio con regularidad?: Drogas en los ultimos 5 años?:

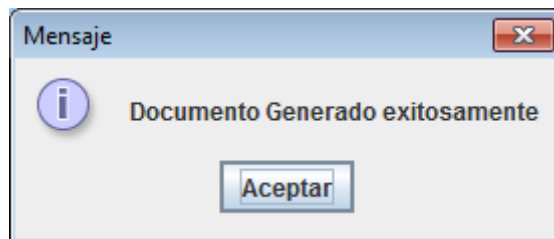
Cafeina(tazas):

Alcohol(vasos): Cigarros al dia:

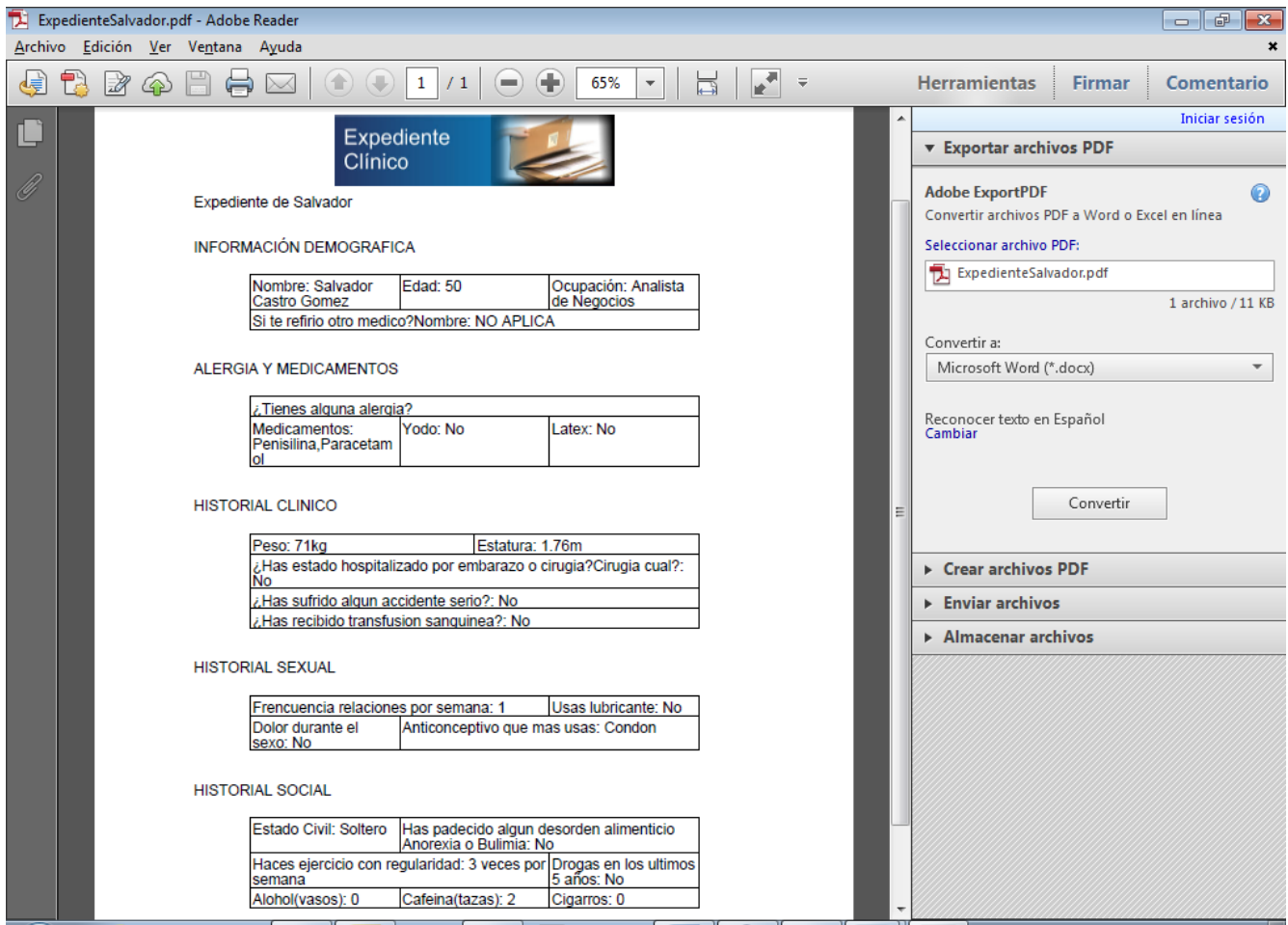
- Ya que se esta en la interfaz donde se muestra la información se mencionaba en el capitulo anterior acerca de la generación de un PDF, por lo tanto la siguiente imagen nos muestra el JFileChooser que no es nadamas que la selección de donde se guardara el PDF y el titulo del archivo



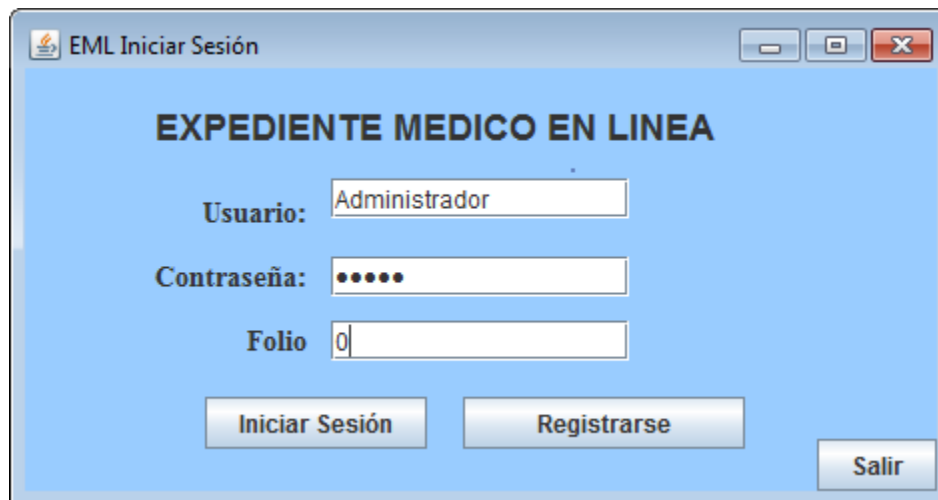
- Ya seleccionado lo anterior se muestra el mensaje de que se ha generado el PDF.



- Una vez obtenido el mensaje de que se ha generado el PDF exitosamente tenemos el archivo para abrirlo y ver su contenido y hasta con la opción de imprimirlo.



- Modo Administrador
 - Misma interfaz de inicio pero ahora la firma es como administrador



- En lugar de mostrar información de un expediente se muestra un formulario vacío para que pueda ser llenado y dar de alta un nuevo expediente

Expediente Medico de Usuario

Bienvenido a tu Expediente
Administrador

No. Folio

Información Demografica

Nombre: Edad: Ocupación:

Si te refirió algún médico? Nombre:

Alergias y Medicamentos

Tienes alguna alergia?

Medicamentos: Yodo/Tintes: Latex:

Historial Clínico

Peso Actual: Estatura Actual:

Has estado hospitalizado por embarazo o cirugía? Cirugía Cual?:

Has sufrido un accidente serio?:

Ha recibido transfusión sanguínea?:

Enfermedades que has padecido?:

Historial Sexual

Frecuencia relaciones sexuales por semana: Usas lubricante:

Dolor durante el sexo?: Anticonceptivo que más usas:

Historial Social

Estado Civil: Has padecido un desorden alimenticio anorexia o bulimia?:

Haces ejercicio con regularidad?: Drogas en los últimos 5 años?:

Cafeína(tazas):

Alcohol(vasos): Cigarros al día:

Buscar Guardar Regresar

- Se llenan los campos del formulario con la información especificada para dar de alta el nuevo registro

Expediente Medico de Usuario

Bienvenido a tu Expediente

Administrador

No. Folio

Información Demografica

Nombre: Edad: Ocupación:

Si te refirió algun medico? Nombre:

Alergias y Medicamentos

Tienes alguna alergia?

Medicamentos: Yodo/Tintes: Latex:

Historial Clinico

Peso Actual: Estatura Actual:

Has estado hospitalizado por emabarazo o cirugia? Cirugia Cual?:

Has sufrido un accidente serio?:

Ha recibido transfusion sanguinea?:

Enfermedades que has padecido?:

Historial Sexual

Frecuencia relaciones sexuales por semana: Usas lubricante:

Dolor durante el sexo?: Anticonceptivo que mas usas:

Historial Social

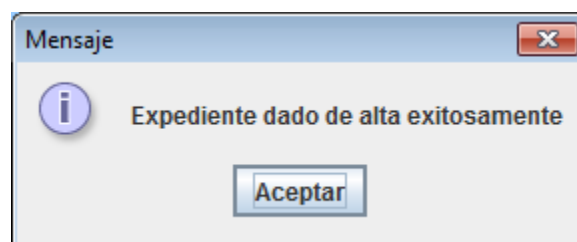
Estado Civil: Has padecido un desorden alimenticio anorexia o bulimia?:

Haces ejercicio con regularidad?: Drogas en los ultimos 5 años?:

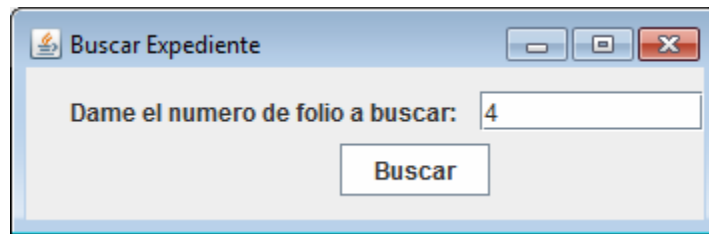
Cafeina(tazas):

Alcohol(vasos): Cigarros al dia:

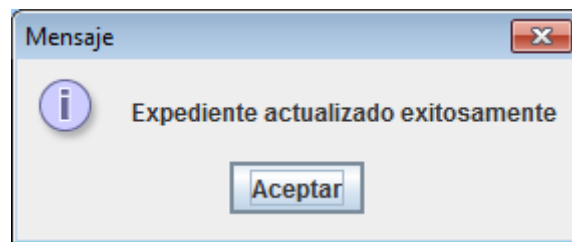
- Cuando ya este la información capturada se da de alta el expediente y manda un mensaje



- Ahora otra opción para el administrador es buscar expedientes por lo que la siguiente imagen muestra el cuadro de dialogo donde se realiza la busqueda por el numero de folio

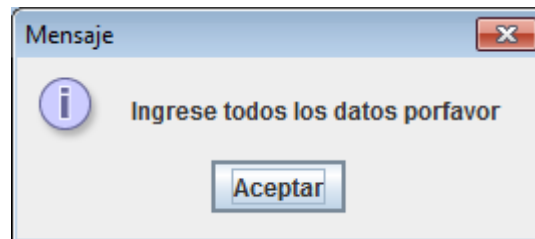


- Con esta búsqueda nos devuelve el expediente con el número de folio dado y el administrador puede validar la información únicamente o también puede editarla. Si se edita la información nos devuelve el siguiente mensaje.

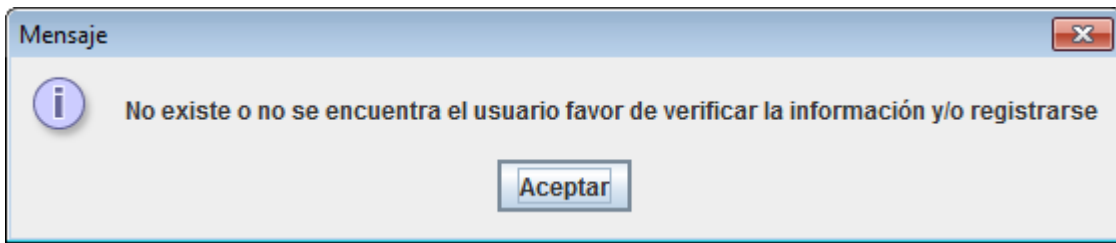


Las imágenes anteriores nos muestran un funcionamiento del SCEML sin alguna validación aun, pero existen muchas validaciones en el registro de usuarios si existe o no al momento de dar de alta un expediente, al buscar, al editar y porque no hasta para iniciar sesión. A continuación se mostrarán los mensajes de algunas de las validaciones que se han mencionado.

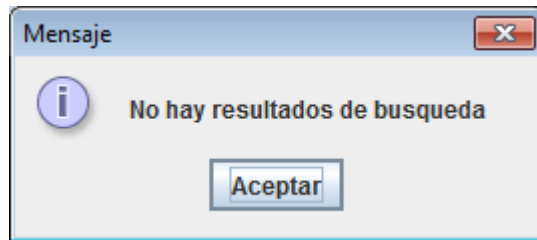
- Iniciar Sesión no se capturan todos los datos



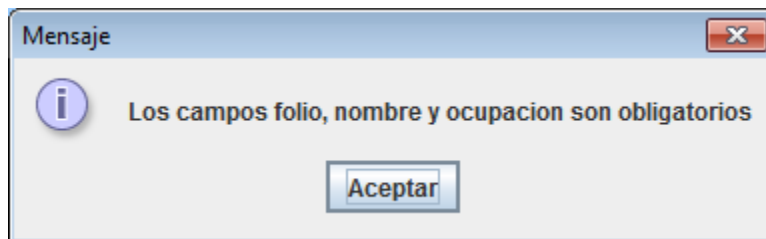
- El usuario no existe o la información es errónea



- No se encuentra el expediente con el folio a buscar



- Cuando se intenta guardar un expediente en blanco



Ahora se mostraran las imágenes del flujo principal para Android, cabe mencionar que es unicamente consulta.

4.2. Evaluación de tiempo de respuesta

Para poder realizar la evaluación en cuanto a los tiempos de respuesta en la ejecución del SCEML en Java y Android se utilizo un metodo que pertenece a Java y como Android se basa en Java, se pudo tomar la misma para dicho objetivo.

El metodo utilizado fue “*java.lang.System.currentTimeMillis()*”, el cual devuelve la hora actual en la unidad milisegundos. El tiempo del valor de retorno es una milésima de segundo, la granularidad del valor depende del sistema operativo subyacente y puede ser mayor. Por ejemplo, existen sistemas operativos que miden el tiempo en unidades de decenas de milisegundos.

Para obtener los tiempos en Java se implemento el codigo de la siguiente manera:

- Al principio de cada proceso dentro del codigo se escribio la linea siguiente , esto fue para obtener el tiempo al inicio de la accion.

```
double tiempoInicio =System.currentTimeMillis();
```

- Al final de cada proceso dentro del codigo se escribieron las siguientes lineas para obtener el tiempo actual al finalizar la acción.

```
double tiempoFin = System.currentTimeMillis();
```

- Al fin para calcular el tiempo total se realiza la siguiente operación , al mismo tiempo se transforman a segundos ya que java lo maneja en milisegundos.

```
double tiempo = tiempoFin - tiempoInicio;  
System.out.println("Tiempo en mostrar interfaz de Buscar:"+tiempo*.001);
```

Una vez obtenidos los tiempos se realiza la Tabla 5 comparativa en tiempos de respuesta.

Tabla 5 Comparación de tiempos de respuesta entre JAVA y ANDROID

JAVA		ANDROID	
Acción	Tiempo (segundos)	Acción	Tiempo (segundos)
Acceso al Sistema		Acceso al Sistema	
Tiempo de Validación de Usuario y Mostrar Expediente:	1.915	Tiempo de Validación de Usuario y Mostrar Expediente:	1.850
Consultar Expediente		Consultar Expediente	
Tiempo de Validación de Usuario y Mostrar Expediente:	1.915	Tiempo de Validación de Usuario y Mostrar Expediente:	1.706
Registrar Usuario		Registrar Usuario	
Tiempo Mostrar Interfaz Registrarse:	0.123	Tiempo Mostrar Interfaz Registrarse:	0.230
Tiempo de validar nuevo registro de usuario en la base de datos:	1.57	Tiempo de validar nuevo registro de usuario en la base de datos:	1.82
Tiempo de insertar nuevo registro de usuario en la base de datos:	1.234	Tiempo de insertar nuevo registro de usuario en la base de datos:	1.825
Registrar Expediente			
Tiempo de validar un nuevo expediente en la base de datos:	1.263		
Tiempo de insertar un nuevo expediente en la base de datos:	1.39		
Buscar Expediente			
Tiempo en mostrar interfaz de Buscar:	0.22		
Tiempo en mostrar el nuevo expediente encontrado:	0.293		
Editar Expediente			
Tiempo en validar cambios en expediente:	2.35		
Tiempo en actualizar un expediente:	1.181		
Generar Expediente (PDF)			
Tiempo en generar el PDF:	2.88		
TOTAL TIEMPOS DE RESPUESTA	16.334		7.431

Nota: Los datos obtenidos fueron en base a la primera ejecución del programa, sin embargo una vez en ejecución el software el tiempo de respuesta es mucho menor.

4.3. Ejemplos de otro sistema de expediente en el mercado

En este apartado del estudio se da a conocer algunas aplicaciones o sistemas que se encuentran en el mercado hoy en día. A continuación se enlistaran y tendrán una breve descripción de lo que realizan.

- a) **Medicatec.** Es un sistema el cual está diseñado para tener comunicación entre médicos y pacientes constantemente así como tener información básica, entre otras funciones tiene:
 - a. **Expedientes.** Es capaz de consultar, modificar e imprimir información acerca de los pacientes: Información general del paciente, historial de consultas, historial de recetas, antecedentes clínicos.
 - b. **Hecho a la medida.** Portal Medicatec personalizado basándose en las necesidades particulares, con el fin de brindar el mejor servicio de asistencia administrativa a todas las especialidades de la salud.
 - c. **Archivos.** Almacene los archivos digitales de cada de sus pacientes. Fotografías, documentos de Microsoft Office, PDF, videos etc.

- b) **Clinixon.** Es un sistema de gestión para clínicas cuenta con agenda, historia clínica, recordatorios entre otras características que a continuación se mencionaran
 - a. **Monitoreo.** Sigue a los pacientes, para que no abandonen los tratamientos. Puedes configurar funciones de seguimiento y mensajes motivacionales automatizados.
 - b. **Agenda.** Realiza citas con solamente un clic con la agenda interactiva fácil de usar. Puede visualizar la agenda de un solo doctor o de varios a la vez, de una o más sucursales, todo ello en el mismo calendario.
 - c. **Historia Clínica.** Acceder a los registros clínicos, indicaciones, diagnóstico, y demás datos que constituyen la historia clínica del paciente, las 24 hs y desde tu computadora o tablet.

- c) **MedicalApp.** Aplicación diseñada para la gestión de un consultorio por internet, puede ver historial clínico de sus pacientes, programar citas por día, semana o mes. Se

pueden dar de altas pacientes, ordenar historial clínico, incluir fotos o PDF's de estudios médicos, padecimientos y medicamentos.

4.4. Tabla comparativa de ventajas y desventajas de ambas plataformas Java y Android

En esta sección del estudio se hace un breve análisis comparativo entre las dos plataformas utilizadas para el desarrollo del SCEML, así como se mencionan algunas ventajas y desventajas de las plataformas.

A continuación se muestra la Tabla 6 que contiene la información sobre el análisis comparativo

Tabla 6 Comparativa de ventajas y desventajas entre la plataforma Java y Android

JAVA		ANDROID	
Ventajas	Desventajas	Ventajas	Desventajas
Lenguaje Simple: Es fácil adquirir el conocimiento sobre el uso del lenguaje Java. Todos aquellos usuarios familiarizados con C++ encontrarán que Java es más sencillo, ya que se han eliminado ciertas características, tales como los punteros.	Menos Eficiente, comparado a C/C++.	Plataforma realmente abierta. Es una plataforma de desarrollo libre basada en Linux y de código abierto.	Google tiene una política restrictiva hacia las versiones más recientes, no haciéndolas públicas hasta que ellos lo vean conveniente.
Orientado a Objetos: Java fue diseñado como un lenguaje orientado a objetos desde el inicio de su creación. Los objetos agrupan, en estructuras encapsuladas, tanto sus datos como los métodos (o funciones) que manipulan dichos datos.	Una mala implementación de un programa en java, puede resultar en algo muy lento.	Portabilidad asegurada. Las aplicaciones finales son desarrolladas en Java lo que asegura que podrán ser ejecutadas en gran variedad de dispositivos, tanto presentes como futuros. Lo anterior se consigue gracias al concepto de máquina virtual.	Al ser un sistema multitarea, esto puede presentar un problema ya que se puede llevar a cabo un consumo elevado de la batería.

<p>Distribuido: Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, lo que facilita, así, la creación de aplicaciones distribuidas.</p>		<p>Arquitectura basada en componentes inspirados en internet. Por ejemplo, el diseño de la interfaz de usuario se hace en XML, lo que permite que una misma aplicación se ejecute en un móvil de pantalla reducida o en un notebook.</p>	<p>Suele ser poco intuitivo para los usuarios nuevos</p>
<p>Interpretado y compilado a la vez: Java es compilado en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador. Por otra parte dicho código es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina en la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (run-time).</p>		<p>Gran Cantidad de servicios incorporados: por ejemplo, localización basada tanto en GPS como en torres de telefonía móvil. Incorpora potentes bases de datos con SQL, entre otros.</p>	
<p>Robusto: Java fue diseñado para crear software altamente fiable. Para ello, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Las características de memoria de Java liberan a los programadores de cometer posibles errores comunes (la aritmética de punteros), puesto que se ha prescindido por completo los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.</p>		<p>Alto nivel de seguridad. Los programas se encuentran aislados unos de otros gracias al concepto de ejecución dentro de una caja que incorpora la máquina virtual. Cada aplicación dispone de una serie de permisos que limitan su rango de actuación (servicios de localización, acceso a internet).</p>	

<p>Indiferente a la Arquitectura. Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows Nt, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos.</p>		<p>Alta calidad de gráficos y sonido: gráficos vectoriales suavizados, animaciones inspiradas en Flash, gráficos en 3 dimensiones basados en OpenGL. Incorpora códecs estándar más comunes de audio y video incluyendo H.264(AVC), MP3, AAC.</p>	
---	--	--	--

5. CONCLUSIONES

En el proyecto de tesis “Consulta de Expediente Médico en Línea a través de las plataformas Java y Android” se desarrolló un programa computacional en lenguaje Java y Android en su versión 4.0. Al finalizar el trabajo se establecen las conclusiones que se explicarán a continuación.

En primer lugar se concluye que el fin del sistema desarrollado en este proyecto no es comercial ni para ser empleado por el sector salud ya que antes se requeriría validarlo utilizando datos experimentales, teniendo ambientes de prueba ya que algunas no cuenta con la infraestructura.

De igual modo, el presente trabajo de tesis tuvo como fin establecer un funcionamiento tanto para el público en general como público especializado, para realizar una consulta del expediente médico y en dado caso un alta o modificación. Se concluye que tiene un buen funcionamiento, además de que cuenta con la información mínima requerida.

Asimismo, los lenguajes de programación tanto Java como Android, permitieron el desarrollo de un software con una interface amigable con el usuario lo cual satisface el requerimiento de ser apto para fines educativos; además de que el resultado final fue un programa computacional con características profesionales y que permite su fácil entendimiento, entre las cuales se pueden mencionar cuadros de dialogo, notificaciones, consejos, etc.

5.1. Oportunidades de Mejora

El software desarrollado en este proyecto tiene oportunidad de mejoras respecto a incluir otros parámetros para la consulta, alta o modificación de un expediente, generación de documentos, y porque no hasta implementar nuevos de información hasta tener un expediente muy completo.

También se puede mejorar las interfaces para que sea más dinámicas, con atractivos visuales, la presentación de los datos, el despliegue de información, la generación de documentos, etc. Finalmente, se exhorta a continuar con proyecto enfocados al desarrollo de programas computacionales para ser empleados en el proceso enseñanza-aprendizaje.

Bibliografía

(s.f.). Obtenido de <http://www.openhansetalliance.com>

Eckel, B. (s.f.). *Piensa en Java*. Prentice Hall.

Gilfillan, I. (s.f.). *La Biblia MySQL*. Anaya Multimedia.

Guillermo Storti, G. R. (2007). *Tecnología de la Información y la Comunicación*. Recuperado el 12 de 11 de 2013, de Base de datos Modelo Entidad Relación: www.belgrano.esc.edu.ar

INEGI. (28 de 01 de 2014). *INEGI*. Obtenido de www.inegi.org.mx

Jésus, T. G. (2012). *El Gran Libro de Android*. Mexico.

Juan Palacio, C. R. (1.4.0 – Enero de 2011). *Scrum Manager Gestión de Proyectos*. Safe Creative.

Schmuller, J. (s.f.). *Aprendiendo UML en 24 Horas*. Prentice Hall.

ANEXOS

Manual de Usuario



Manual de usuario

Sistema de Consulta de Expediente Médico en Línea

Autor: Salvador Castro Gómez

Versión: 01.00

CONTENIDO

1 OBJETO DEL DOCUMENTO	89
2 OBJETIVOS	89
3 MANUAL DE USUARIO JAVA	89
3.1 Pantalla Inicial.....	89
3.2 Iniciar Sesión.....	89
3.3 Exportar PDF	90
3.4 Registrarse	91
4 MANUAL DE USUARIO ADMINISTRADOR	92
4.1 Alta de un expediente	94
4.2 Buscar expediente.....	95
4.3 Editar expediente.....	96
4.4 Datos asociados	97
5 MANUAL DE USUARIO ANDROID	97
5.1 Pantalla Inicial.....	97
5.2 Iniciar sesión	98
5.3 Registrarse	99
5.4 Muestra Expediente	101

Tablas de Ilustraciones

Ilustración 1 Inicio de la aplicación	89
Ilustración 2 Muestra expediente.....	90
Ilustración 3 Seleccionar nombre y ubicación del PDF	90
Ilustración 4 PDF generado con la información	91
Ilustración 5 Mensaje de error de autenticación.....	92
Ilustración 6 Interfaz de registro de usuarios	92
Ilustración 7 Firma como administrador	93
Ilustración 8 Interfaz para usuario administrador	93
Ilustración 9 Mensaje de campos requeridos para el expediente	94
Ilustración 10 Vista previa a guardar expediente	95
Ilustración 11 Mensaje de alta de expediente	95
Ilustración 12 Ventana de búsqueda de expediente	96
Ilustración 13 Expediente listo para editar	96
Ilustración 14 Mensaje de actualización de expediente.....	97
Ilustración 15 Mensaje de datos asociados	97
Ilustración 16 Pantalla inicial Android	98
Ilustración 17 Inicio de sesión	98
Ilustración 18 Mensaje de usuario no registrado o información errónea	99
Ilustración 19 Datos para registro de usuario.....	100
Ilustración 20 Validación de contraseñas	100
Ilustración 21 Datos de expediente.....	101

1. OBJETO DEL DOCUMENTO

El presente documento pretende mostrar al usuario el funcionamiento de la aplicación Sistema de Consulta de Expediente Médico en Línea.

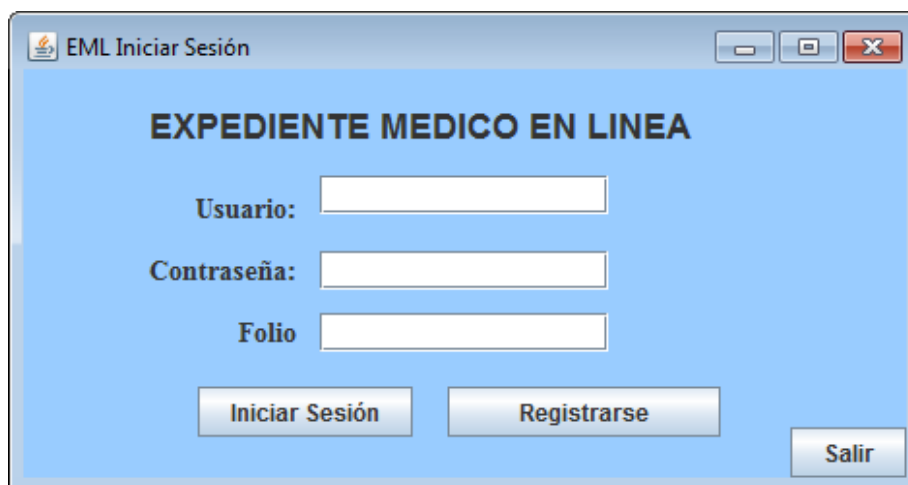
2. OBJETIVOS

Se pretende mostrar de una manera clara y concisa el funcionamiento de la aplicación Sistema de Consulta de Expediente Médico en Línea.

3. MANUAL DE USUARIO JAVA

3.1. Pantalla Inicial

La pantalla de inicio de la aplicación da la bienvenida al usuario al sistema y ofrece las distintas alternativas para la consulta de expediente médico. Las distintas funcionalidades se definen en los apartados posteriores.



The image shows a screenshot of a Java application window titled "EML Iniciar Sesión". The window has a light blue background and a title bar with standard Windows window controls (minimize, maximize, close). The main content area features the heading "EXPEDIENTE MEDICO EN LINEA" in bold, dark blue text. Below the heading, there are three input fields: "Usuario:", "Contraseña:", and "Folio". Each field is a simple white rectangle with a thin border. At the bottom of the window, there are three buttons: "Iniciar Sesión", "Registrarse", and "Salir". The "Iniciar Sesión" and "Registrarse" buttons are larger and positioned side-by-side, while the "Salir" button is smaller and positioned to the right. All buttons have a light blue gradient and a thin border.

Ilustración 1 Inicio de la aplicación

3.2. Iniciar Sesión

El usuario debe capturar los datos solicitados, si decide solicitar el inicio de sesión, el sistema realizara las validaciones correspondientes con la información capturada. Si la información es correcta el sistema mostrara una nueva ventana mostrando la información relacionada a los datos del usuario.

Bienvenido a tu Expediente
Salvador

Información Demografica
 Nombre: Edad: Ocupación:
 Si te refirió algun medico? Nombre:

Alergias y Medicamentos
 Tienes alguna alergia?
 Medicamentos: Yodo/Tintes: Latex:

Historial Clínico
 Peso Actual: Estatura Actual:
 Has estado hospitalizado por emabarazo o cirugia? Cirugia Cual?:
 Has sufrido un accidente serio?:
 Ha recibido transfusion sanguinea?:
 Enfermedades que has padecido?:

Historial Sexual
 Frecuencia relaciones sexuales por semana: Usas lubricante:
 Dolor durante el sexo?: Anticonceptivo que mas usas:

Historial Social
 Estado Civil: Has padecido un desorden alimenticio anorexia o bulimia?:
 Haces ejercicio con regularidad?: Drogas en los ultimos 5 años?:
 Cafeina(tazas):
 Alcohol(vasos): Cigarros al día:

Ilustración 2 Muestra expediente

3.3. Exportar PDF

El usuario puede tan solo verificar los datos. Pero también puede generar un archivo PDF para su impresión física de los datos del usuario. Si el usuario decide seleccionar la opción exportar el sistema mostrara una ventana para seleccionar el nombre y la ubicación donde se generará el PDF.

Guardar

Guardar en:

Camtasia Studio
 DrExplain projects
 Navicat
 NetBeansProjects
 Bitácora de Instalación y evaluación del System Center Endpoint Protection 2012.docx
 confKAS.cfg

Nombre de archivo:

Archivos de tipo:

Ilustración 3 Seleccionar nombre y ubicación del PDF

Una vez seleccionado lo anterior tenemos el archivo PDF generado como el que se muestra a continuación

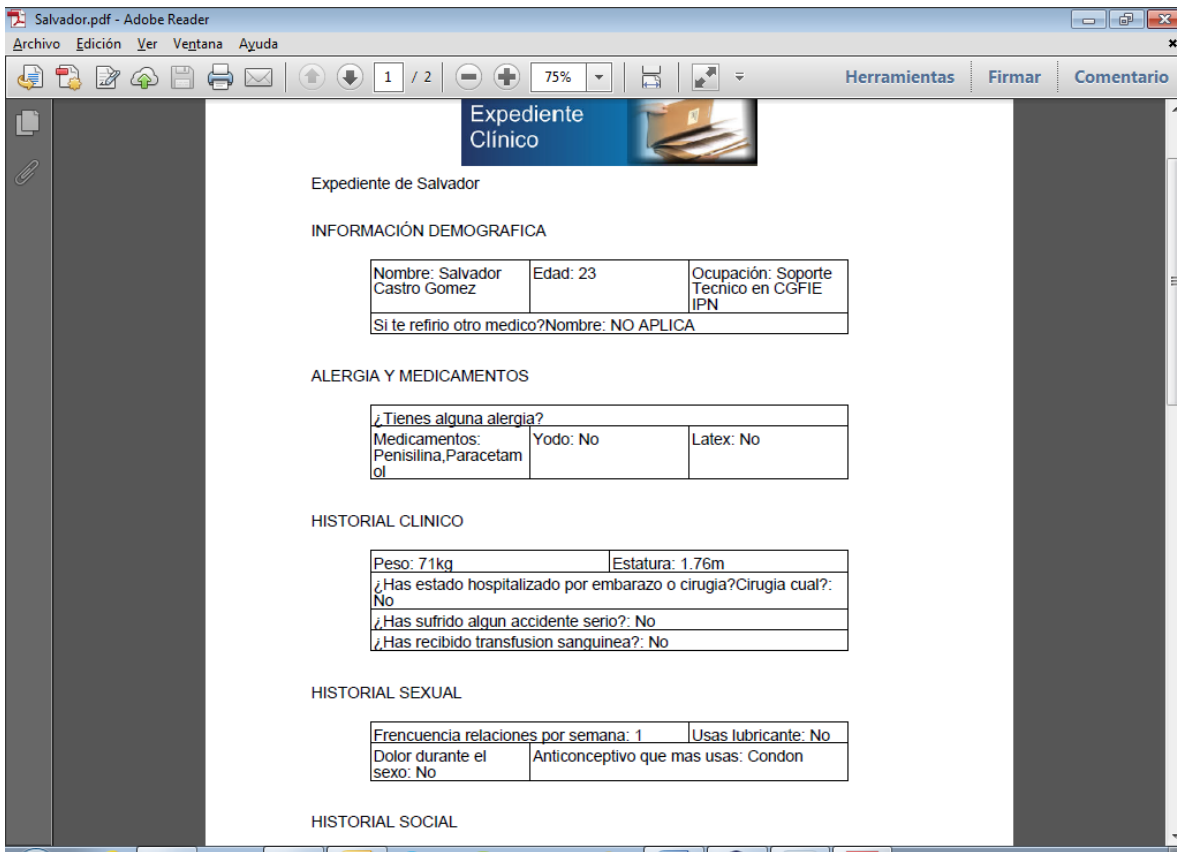


Ilustración 4 PDF generado con la información

3.4. Registrarse

En caso de que el usuario no haya podido iniciar sesión solo se debe a dos posibles causas que son:

- Información capturada errónea
- No exista el usuario

En ambos casos el sistema muestra el siguiente mensaje

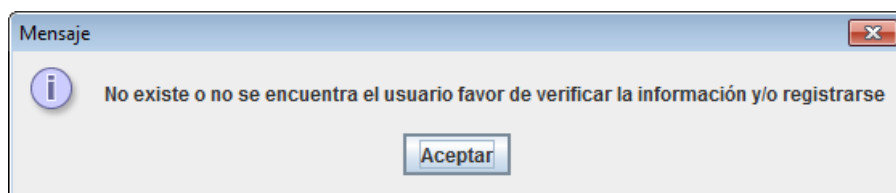


Ilustración 5 Mensaje de error de autenticación

Sin embargo si es el caso en que no exista el usuario el usuario debe acceder a la interfaz de registrarse para que tenga acceso al sistema. La interfaz es la siguiente

Ilustración 6 Interfaz de registro de usuarios

Una vez dada la información podrá ingresar al sistema sin problema alguno y verificar si existe un expediente asociado a su número de folio o no en caso de no tener alguna información ponerse en contacto con el administrador.

4. MANUAL DE USUARIO ADMINISTRADOR

El usuario administrador tiene los permisos en el sistema de poder realizar un alta o cambio de cualquier expediente, únicamente necesita firmarse con el usuario administrador y la contraseña personalizada.

Ilustración 7 Firma como administrador

Una vez validada la información se muestra la interfaz de administrador con las diferentes opciones

Ilustración 8 Interfaz para usuario administrador

4.1. Alta de un expediente

El usuario administrador para dar de alta un expediente, únicamente es necesario llenar la información que se muestra en la interfaz. Existen campos básicos que son requeridos para dar de alta el expediente los cuales se mencionan en un mensaje de error al intentar guardar.

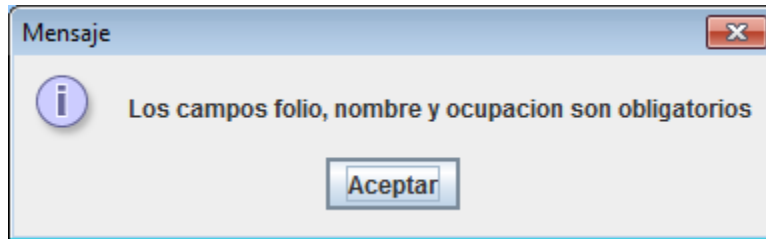


Ilustración 9 Mensaje de campos requeridos para el expediente

Con estos campos se puede dar de alta un expediente sin embargo todos los campos tienen sus validaciones en cuanto a la aceptación únicamente de letras, números o algún formato en especial. Una vez aplicadas las validaciones tendremos una vista de la siguiente manera.

Expediente Medico de Usuario

Bienvenido a tu Expediente
Administrador

No. Folio: 6

Información Demografica
 Nombre: Edad: Ocupación:
 Si te refirió algun medico? Nombre:

Alergias y Medicamentos
 Tienes alguna alergia?
 Medicamentos: Yodo/Tintes: Latex:

Historial Clinico
 Peso Actual: Estatura Actual:
 Has estado hospitalizado por embarazo o cirugia? Cirugia Cual?:
 Has sufrido un accidente serio?:
 Ha recibido transfusion sanguinea?:
 Enfermedades que has padecido?:

Historial Sexual
 Frecuencia relaciones sexuales por semana: Usas lubricante:
 Dolor durante el sexo?: Anticonceptivo que mas usas:

Historial Social
 Estado Civil: Has padecido un desorden alimenticio anorexia o bulimia?:
 Haces ejercicio con regularidad?: Drogas en los ultimos 5 años?:
 Cafeina(tazas):
 Alcohol(vasos): Cigarros al dia:

Ilustración 10 Vista previa a guardar expediente

Si las validaciones son correctas y no hay ningún error o advertencia el sistema mostrara el mensaje de que se dio de alta el expediente exitosamente.

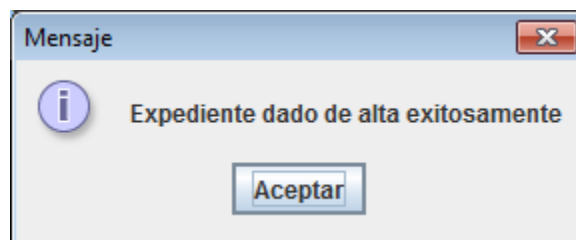


Ilustración 11 Mensaje de alta de expediente

4.2. Buscar expediente

Para la búsqueda de un expediente es necesario firmarse como administrador, una vez en la interfaz de administrador se selecciona la opción buscar y tendremos la siguiente ventana donde la búsqueda se realiza por el número de folio.

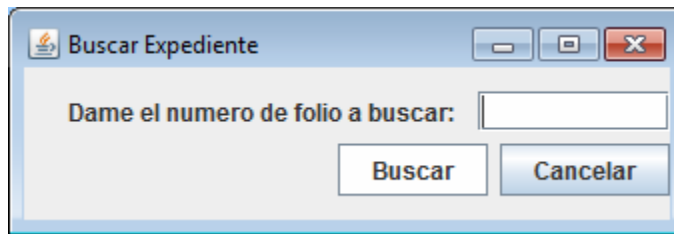


Ilustración 12 Ventana de búsqueda de expediente

4.3. Editar expediente

Una vez que el usuario administrador haya buscado un expediente por número de folio el sistema mostrara la información de dicho expediente de forma editable ya que si es necesario editarlo pueda realizarlo.

Bienvenido a tu Expediente No. Folio

Información Demografica
Nombre: Edad: Ocupación:
Si te refirió algun medico? Nombre:

Alergias y Medicamentos
Tienes alguna alergia?
Medicamentos: Yodo/Tintes: Latex:

Historial Clínico
Peso Actual: Estatura Actual:
Has estado hospitalizado por emabarazo o cirugia? Cirugia Cual?:
Has sufrido un accidente serio?:
Ha recibido transfusion sanguinea?:
Enfermedades que has padecido?:

Historial Sexual
Frecuencia relaciones sexuales por semana: Usas lubricante:
Dolor durante el sexo?: Anticonceptivo que mas usas:

Historial Social
Estado Civil: Has padecido un desorden alimenticio anorexia o bulimia?:
Haces ejercicio con regularidad?: Drogras en los ultimos 5 años?:
Cafeina(tazas):
Alcohol(vasos): Cigarros al dia:

Ilustración 13 Expediente listo para editar

Si el usuario modifica algún campo de la información y selecciona la opción “Editar”, se mostrará el siguiente mensaje.

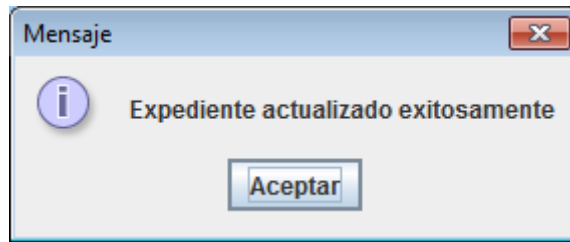


Ilustración 14 Mensaje de actualización de expediente

4.4. Datos asociados

Si el usuario está registrado, pero sin embargo no existen información asociada a dicho usuario se mostrará el siguiente mensaje.

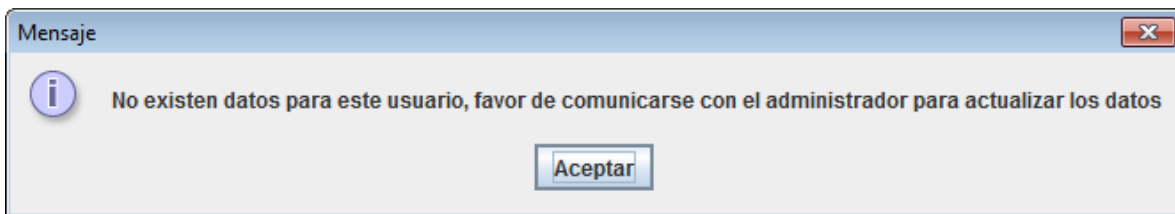


Ilustración 15 Mensaje de datos asociados

5. MANUAL DE USUARIO ANDROID

5.1. Pantalla Inicial

La pantalla de inicio de la aplicación da la bienvenida al usuario al sistema y ofrece dos distintas alternativas, ya sea el inicio de sesión o el registrarse en el sistema para la consulta de expediente médico. Las distintas funcionalidades se definen en los apartados posteriores.

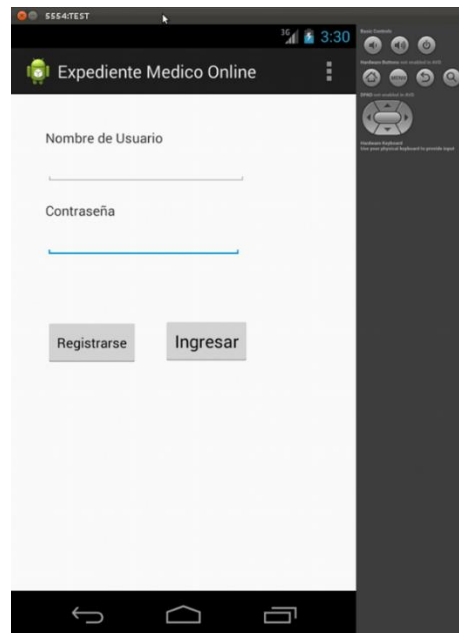


Ilustración 16 Pantalla inicial Android

5.2. Iniciar sesión

El usuario debe capturar los datos solicitados, si decide solicitar el inicio de sesión, el sistema realizara las validaciones correspondientes con la información capturada. Si la información es correcta el sistema mostrara una nueva ventana mostrando la información relacionada a los datos del usuario como se ve en la Ilustración 17.

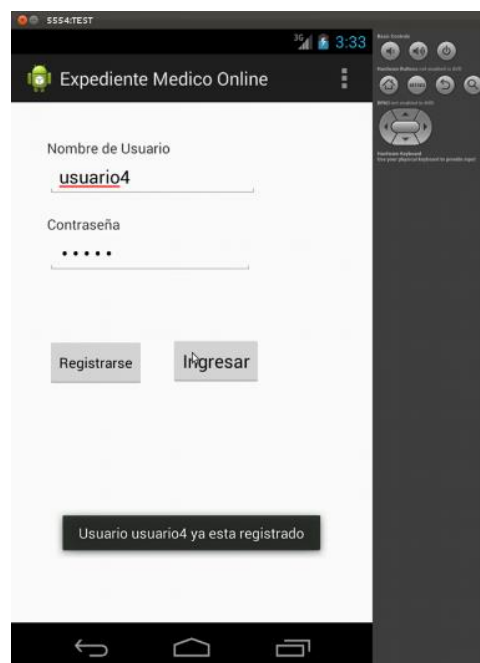


Ilustración 17 Inicio de sesión

5.3. Registrarse

En caso de que el usuario no haya podido iniciar sesión solo se debe a dos posibles causas que son:

- Información capturada errónea
- No exista el usuario

En ambos casos la aplicación muestra el siguiente mensaje en la Ilustración 18

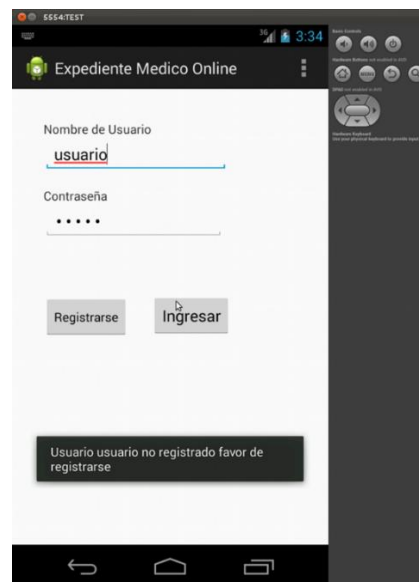


Ilustración 18 Mensaje de usuario no registrado o información errónea

En el caso en que no exista el usuario el usuario debe acceder a la interfaz de registrarse para que tenga acceso al sistema. La interfaz se muestra en la Ilustración 19.

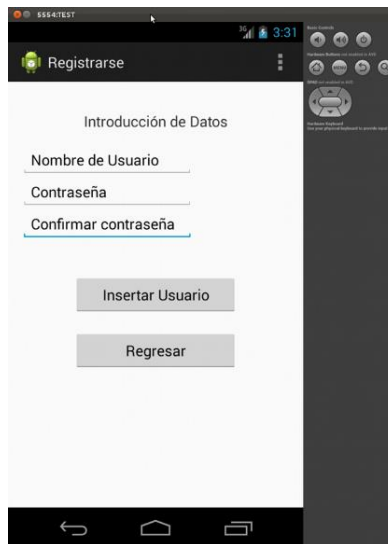


Ilustración 19 Datos para registro de usuario

Quando el usuario se está registrando existe la validación de que deben coincidir las contraseñas, en caso contrario muestra un mensaje de error que se muestra en la Ilustración 20.

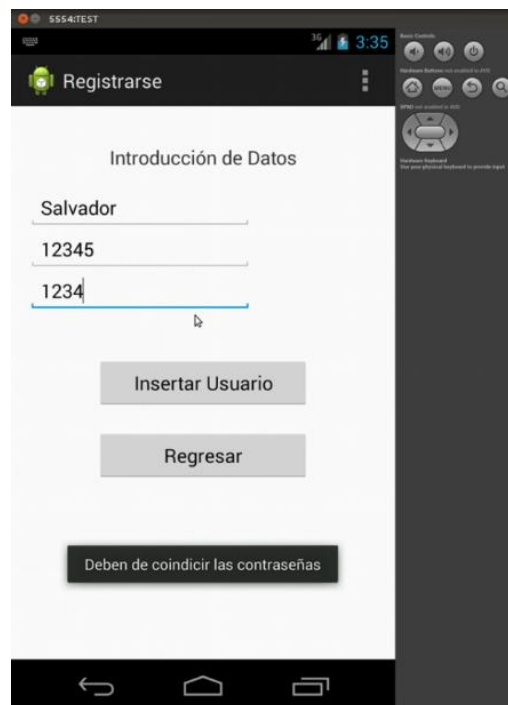


Ilustración 20 Validación de contraseñas

5.4. Muestra Expediente

Finalmente cuando el usuario tiene la información necesaria y está registrado en la aplicación se podrá mostrar su información correspondiente a dicho usuario. En la Ilustración 21 se muestran los datos del expediente.

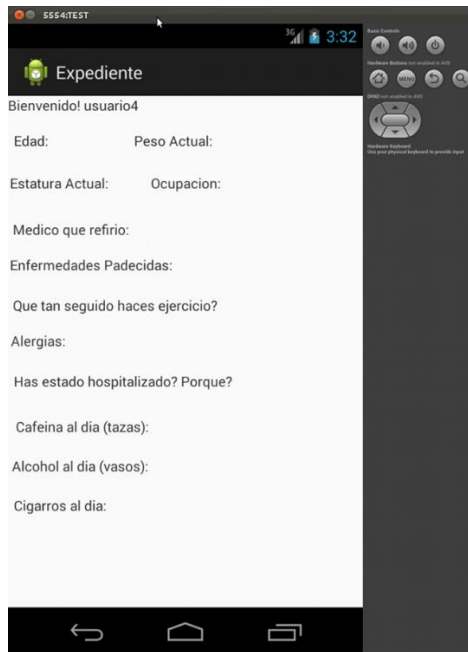


Ilustración 21 Datos de expediente