

Universidad Autónoma Metropolitana Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Reporte final del proyecto de integración

Modalidad de Proyecto Tecnológico

Aplicación móvil para visualizar el expediente médico de un paciente utilizando servicios web

Erick Esteven Montelongo González
Matrícula: 2123030877

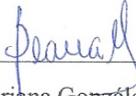
Asesora
Dra. Beatriz Adriana González Beltrán

Trimestre 17 – Invierno

Fecha de entrega

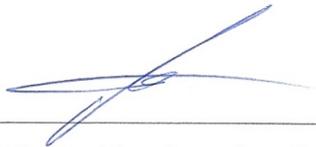
3 de abril del 2017

Yo, Beatriz Adriana González Beltrán, declaro que aprobé el contenido del Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco



Beatriz Adriana González Beltrán

Yo, Erick Esteven Montelongo González, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Erick Esteven Montelongo González

Resumen

En la actualidad existen muy pocas aplicaciones que consideran la información necesaria para un expediente médico. Si consideramos que en México los pacientes no acostumbran ser atendidos por un solo médico a lo largo de su vida; sino que son atendidos por distintos médicos en distintas instituciones de salud, cada una de ellas puede contar con un expediente médico propio de cada paciente, y la información en cada uno de estos expedientes puede cambiar.

El objetivo de este proyecto de integración es el desarrollar una aplicación móvil para visualizar el expediente médico de un paciente, a saber: historial médico, medicamentos consumidos, síntomas presentados y estudios realizados, utilizando servicios web.

Para desarrollar el proyecto, se consideraron que existen aplicaciones que se han desarrollado o se encuentran en fase de desarrollo como proyecto de integración, que permiten al paciente administrar su información médica, que corresponden a las áreas del expediente médico. En base a estas aplicaciones se obtiene la información para formar el expediente médico y presentarlo a un médico. Las aplicaciones mencionadas son las siguientes:

- **Prototipo de una aplicación móvil para la gestión de síntomas de un paciente.** Aplicación móvil que se encarga de la administración de los síntomas del paciente.
- **FotoMed: Prototipo de aplicación Android para la gestión de los estudios de un paciente.** Aplicación móvil que se encarga de la administración de los estudios de laboratorio del paciente.
- **Prototipo de aplicación Android para la gestión de medicamentos.** Aplicación móvil que se encarga de la administración de los medicamentos del paciente
- **Aplicación Móvil para la administración del historial médico.** Aplicación que se encarga de la administración del historial clínico del paciente.

La metodología utilizada para el desarrollo del proyecto fue el Proceso Unificado (UP) que considera para el desarrollo de software etapas iterativas e incrementales, centradas en la arquitectura y dirigidas por casos de uso.

En el proyecto se diseñó una base de datos que considera el expediente médico completo, es decir, puede guardar la información de las aplicaciones externas para estandarizar la información a ser almacenada. También se diseñaron e implementaron servicios web que cumplen con las operaciones de inserción y obtención de información de la base de datos. Además se diseñaron e implementaron los módulos de registro de usuario, autenticación de usuario y visualización de cada uno de los elementos del expediente médico que componen una aplicación móvil para dispositivos Android.

La aplicación cumple su propósito de visualización del expediente de un paciente, pero como líneas futuras de trabajo se propone mejorar la interfaz de la aplicación y extender las funcionalidades de la misma; por ejemplo, implementar un sistema de mensajes entre el médico y paciente.

Índice de contenidos

1.	Introducción	1
2.	Antecedentes	2
2.1	Referencias externas	2
2.2	Referencias internas	3
3.	Justificación	4
4.	Objetivos	5
4.1	Objetivo general	5
4.2	Objetivos específicos	5
5.	Marco teórico	6
5.1	Plataforma Android	6
5.2	Características básicas de una aplicación Android	6
5.3	Uso de base de datos MySQL	8
5.4	Servicios web REST	9
6.	Desarrollo del proyecto	10
6.1	Metodología empleada en el desarrollo del proyecto	10
6.2	Diseño del sistema	10
6.2.1	Diagrama de casos de uso	10
6.2.2	Casos de uso de texto	12
6.2.3	Diagrama de clases	17
6.2.4	Arquitectura del sistema	25
6.2.5	Estructura de la base de datos	26
6.3	Uso del sistema	36
6.3.1	Registro de usuario	36
6.3.2	Ingreso a la aplicación	39
6.3.3	Visualización y búsqueda de la lista de pacientes	41
6.3.4	Visualización de historia clínica	44
6.4	Hardware y software necesarios	47
6.4.1	Hardware	47
6.4.2	Software	47
7.	Resultados	49
8.	Análisis y discusión de resultados	50

8.1	Discusión de resultados.....	50
9.	Conclusiones.....	51
9.1	Experiencia al realizar el proyecto.....	52
10.	Perspectivas del proyecto.....	52
12.	Referencias bibliográficas.....	53
13.	Apéndice A. Entregable: Listado del API del código fuente de los servicios web desarrollados	55
14.	Apéndice B. Entregable: Listado del API del código fuente de la aplicación.....	55

Índice de figuras

Figura 1. Esquema del ciclo de vida de una actividad en Android.....	7
Figura 2. Comunicación llevada a cabo por un servicio web REST.....	9
Figura 3. Diagrama de casos de uso del sistema.....	10
Figura 4. Diagrama de clases del paquete activities (1/2).....	17
Figura 5. Diagrama de clases del paquete activities (2/2).....	21
Figura 6. Clases contenidas en el paquete webServices (1/2).....	24
Figura 7. Clases contenidas en el paquete webServices (2/2).....	25
Figura 8. Arquitectura general del proyecto.....	26
Figura 9. Diagrama entidad - relación de la base de datos del sistema.....	27
Figura 10. Pantalla de registro de la aplicación.....	36
Figura 11. Pantalla de operación de registro.....	37
Figura 12. Texto de ayuda para el registro.....	38
Figura 13. Pantalla de operación de autenticación.....	39
Figura 14. Texto de error al ingresar a la aplicación.....	40
Figura 15. Pantalla de bienvenida a la aplicación.....	40
Figura 16. Pantalla de operación para obtener lista de pacientes.....	41
Figura 17. Pantalla de visualización de lista de pacientes.....	42
Figura 18. Búsqueda de un paciente.....	43
Figura 19. Pantalla de resumen de expediente.....	44
Figura 20. Ícono de menú lateral del expediente médico.....	44
Figura 21. Menú lateral de navegación del expediente médico.....	45
Figura 22. Pantalla principal del historial clínico del paciente.....	46

1. Introducción

El expediente médico se define como el conjunto único de información y datos personales de un paciente utilizado para la atención médica. Usualmente este expediente se compone de una historia clínica, resultados de laboratorio, estudios de gabinete, hojas de trabajo social y otros documentos como indicaciones médicas.

Existen aplicaciones desarrolladas como proyecto de integración para que el paciente gestione su propia información médica, tales como: un prototipo para la gestión de síntomas [1] o FotoMed [2] para la gestión de estudios de laboratorio. Además, existen propuestas para el desarrollo de aplicaciones para la administración de medicamentos [3] y para la administración del historial médico de un paciente [4].

Estas aplicaciones y propuestas, tienen como objetivo la administración personal de la información, sin contemplar una vista para el médico y sin relacionarse entre ellas. Si consideramos la unión de la información de estas aplicaciones se puede obtener un expediente médico para el paciente.

De esta manera, la presente propuesta pretende la centralización de esta información para formar el expediente médico de un paciente a través de aplicación móvil que permita a un médico visualizar dicha información. Además, es necesario incluir un mecanismo para que un paciente pueda autorizar a los médicos el acceso a su información médica.

Cabe señalar que la información obtenida se obtendrá a través de los servicios web que proveen cada una de las aplicaciones mencionadas anteriormente, a saber:

- *Prototipo de una aplicación móvil para la gestión de síntomas de un paciente [1].*
- *FotoMed: Prototipo de aplicación Android para la gestión de los estudios de un paciente [2].*
- *Prototipo de aplicación Android para la gestión de medicamentos [3].*
- *Aplicación Móvil para la Administración del Historial Médico [4].*

2. Antecedentes

Esta sección se refiere a trabajos internos o externos que tienen relación con el proyecto de integración realizado.

2.1 Referencias externas

mobile Medical Records (mMR) [5]. Se trata de una aplicación móvil que considera el expediente clínico de un paciente desde el punto de vista del médico. Tanto la propuesta presentada como mMR se basan en mostrar información clínica relevante al médico, donde este puede tener una vista clara del expediente del paciente. mMR no considera la oportunidad de que el paciente de permisos a los médicos para visualizar la información.

My Medical [6]. Es una aplicación móvil que guarda el expediente médico de un paciente, pudiendo anexar archivos para su visualización. Esta aplicación ofrece de manera similar a esta propuesta la visualización de la información del paciente. La diferencia más importante que se presenta entre My Medical y la aplicación que se propone, es que la primera solamente considera el uso personal para el paciente sin considerar la visualización para el médico. El proyecto a realizar permite que un médico visualice la información del paciente si es que éste último lo autoriza.

Electronic Medical Record Systems for Developing Countries: Review [7]. Este trabajo explora la utilidad del desarrollo de un expediente médico electrónico, específicamente en países en desarrollo. Sobresale la utilidad de que el paciente pueda tener de manera centralizada sus expedientes clínicos contra llevarlos en papel a consultas médicas. El proyecto a realizar toma ideas de este trabajo para el beneficio de los pacientes, como es mantener de manera centralizada la gran cantidad de información que este expediente conlleva.

Mobile Healthcare Information Management utilizing Cloud Computing and Android OS [8]. El trabajo explora los beneficios que se obtienen al utilizar un expediente médico electrónico además de la utilidad de una aplicación que hace uso del cómputo en la nube, donde el paciente pueda administrar dicho expediente. La aplicación presentada en este trabajo hace uso de servicios web; de manera similar a lo que se pretende realizar en el proyecto presentado en esta propuesta. Sin embargo, la propuesta permitirá que un médico pueda visualizar la información de sus pacientes.

2.2 Referencias internas

Prototipo de una aplicación móvil para la gestión de síntomas de un paciente [1]. Prototipo de una aplicación móvil que permite al usuario de manera gráfica indicar en una figura del cuerpo humano que parte de éste tiene algún síntoma e ingresarlo. Además, hace uso de un servicio web para respaldo de la información. Este servicio web será utilizado en la aplicación a desarrollar en esta propuesta, de manera que un médico pueda visualizar los síntomas de un paciente.

FotoMed: Prototipo de aplicación Android para la gestión de los estudios de un paciente [2]. Este prototipo de aplicación permite a un paciente ingresar los estudios médicos que le han realizado. FotoMed cuenta con un servicio web para el respaldo de la información, mismo que se utilizará para recuperarla y presentarla al médico en la aplicación a realizar en esta propuesta.

Prototipo de aplicación Android para la gestión de medicamentos [3]. El objetivo de esta aplicación es que un usuario pueda administrar los medicamentos que se encuentra consumiendo, también proporcionando alertas para el horario en que se deben consumir. Además, permite compartir esta información con otro usuario para que le ayude a administrar los medicamentos en caso de que el paciente se encuentre incapacitado. Esta aplicación provee la información necesaria sobre los medicamentos que consume un usuario, que corresponde al módulo de medicamentos consumidos en la aplicación de la presente propuesta.

Aplicación Móvil para la Administración del Historial Médico [4]. La aplicación que se considera realizar en este trabajo es que un usuario pueda realizar la administración de su historial médico (enfermedades, cirugías, hospitalizaciones, alergias) y mantenerlo en un dispositivo móvil con la capacidad de respaldar los datos en un servidor. Esta aplicación proveería los datos necesarios para el módulo de historial médico de la presente propuesta.

3. Justificación

Existen muy pocas aplicaciones que consideran la información necesaria para un expediente médico, entre ellas HealthVault [9], donde los médicos de una institución son los únicos que pueden gestionar la información de un paciente, además de ser una aplicación con licencia comercial que necesita ser comprada.

Si consideramos que en México, los pacientes no acostumbran a ser atendidos por un solo médico a lo largo de su vida; sino que son atendidos por distintos médicos en distintas instituciones de salud, cada una de ellas puede contar con un expediente médico propio de cada paciente, y la información en cada uno de estos expedientes puede cambiar.

Al centralizar la información de las aplicaciones [1], [2], [3] y [4] como proyecto de esta propuesta, el paciente podrá autorizar a los médicos a visualizar su información, donde él mismo la gestionará, permitiendo que el médico autorizado siempre tenga la información completa del paciente, aunque se haya atendido en distintas instituciones.

El médico tendrá una manera ágil de visualizar la información actualizada del paciente a través de un dispositivo móvil. Además, con la información completa de un expediente médico, un médico puede brindar un diagnóstico más acertado.

4. Objetivos

En esta sección se presentarán tanto el objetivo general como los objetivos específicos que se buscaron cumplir al realizar el proyecto.

4.1 Objetivo general

Desarrollar una aplicación móvil para visualizar el expediente médico de un paciente, a saber: historial médico, medicamentos consumidos, síntomas presentados y estudios realizados, utilizando servicios web.

4.2 Objetivos específicos

1. Diseñar e implementar un servicio web para registrar a los usuarios que deseen hacer uso del expediente médico de un paciente.
2. Diseñar e implementar servicio web para la invocación de los servicios web de las aplicaciones existentes.
3. Diseñar e implementar una base de datos para guardar la información del médico que corresponderá a su información personal y la relación de pacientes que tiene.
4. Diseñar e implementar una aplicación móvil para que el médico pueda visualizar el expediente médico de sus pacientes.

5. Marco teórico

En esta sección se darán conceptos básicos necesarios que forman la base del desarrollo del proyecto.

5.1 Plataforma Android

Android se define como un sistema operativo para dispositivos móviles que es considerado el más utilizado en el mundo. La plataforma se caracteriza por tener un concepto abierto, es decir, que el código fuente del sistema operativo es distribuido de manera libre para que usuarios puedan realizar modificaciones a éste.

Las distintas versiones que existen del sistema operativo tienen sus propios kits de desarrollo o SDK por sus siglas en inglés. Cada uno de los kits brinda las herramientas necesarias para manipular el hardware de los dispositivos, como por ejemplo, las comunicaciones a través de internet utilizando Wi-Fi.

Para realizar el desarrollo de una aplicación bajo la plataforma Android se utiliza el lenguaje de programación Java [10].

El objetivo de la plataforma es brindar un sistema operativo que se acople fácilmente al hardware de distintos fabricantes haciendo uso óptimo de los recursos. Gracias a esto las aplicaciones desarrolladas para esta plataforma tienen un gran alcance en la población pues gracias a la flexibilidad de hardware que ofrece el sistema, la mayor parte de la población con acceso a un dispositivo móvil tiene un sistema operativo Android.

5.2 Características básicas de una aplicación Android

Como toda aplicación que se desarrolle sin importar la plataforma o lenguaje de programación, una aplicación consta de un ciclo de vida. El ciclo de vida de una aplicación se refiere a las distintas etapas por las que pasa desde el punto en que esta es mostrada al usuario, hasta que deja de ser utilizada.

En Android cada aplicación consta de un conjunto de *Activities* que se define como el punto de entrada para la interacción de un usuario en la aplicación [11], que se refiere a la interfaz de usuario y las acciones que se pueden realizar sobre esta.

Las actividades en Android utilizan interfaces gráficas para ser mostradas en pantalla, que previamente se definen en archivos con extensión xml. Estos archivos contienen todos los elementos visuales con los que podrá interactuar el usuario.

Si el usuario comienza a interactuar con otra actividad, ya sea de la misma aplicación o de una distinta, se ejecuta el método *onPause()* que se encarga de mantener la información necesaria de la actividad si es que el usuario volverá a utilizarla.

Si el sistema operativo requiere de recursos adicionales mientras la actividad se encuentra en pausa, esta actividad será destruida y su ciclo de vida volvería a comenzar por *onCreate()*. Si el usuario regresa a esta actividad, se vuelve a ejecutar el método *onResume()*.

Cuando se deja de utilizar la actividad por completo, se ejecuta el método *onStop()* que indica que esta actividad no se encuentra visible para el usuario. Si se requiere regresar a la actividad se ejecuta el método *onRestart()*, pero si por el contrario ya no se vuelve a utilizar por un tiempo determinado o el usuario cierra la aplicación, se ejecuta el método *onDestroy()*.

5.3 Uso de base de datos MySQL

Debido a que la información que utiliza la aplicación puede ser muy grande, se necesita que esta información se encuentre en un medio externo al dispositivo móvil. Por esto se da la solución de crear una base de datos que se encuentre en un servidor accesible a través de internet.

MySQL es un administrador de bases de datos que cuenta con una distribución no comercial. El administrador de base de datos se encarga de obtener instrucciones e interpretarlas para manipular los datos que almacena. Estas instrucciones son proporcionadas por los desarrolladores en un lenguaje de manipulación de datos SQL (Structured Query Language). Las sentencias SQL son diseñadas por los desarrolladores para manipular los datos según lo requiera la aplicación.

A través de su producto MySQL Community Server se da la posibilidad de crear un servidor de base de datos. Esto quiere decir que se puede cargar una base de datos en un servidor y que aplicaciones que deseen utilizar esta información puedan acceder a la base de datos a través de internet.

La carga de base de datos se realiza a través de un archivo sql que contiene sentencias SQL que indican al administrador de base de datos las instrucciones necesarias para la creación del esquema de base de datos.

Mediante el uso del servidor de base de datos es posible almacenar una gran cantidad de datos y solamente obtener una parte de ella, para así optimizar el funcionamiento en dispositivos que no cuentan con una gran cantidad de recursos, como lo son los dispositivos móviles bajo los cuales se ejecuta Android.

5.4 Servicios web REST

Los servicios web son aplicaciones con arquitectura cliente – servidor que se comunican a través de internet utilizando protocolos HTTP [12]. Los servicios web pueden verse como operaciones que se pueden llevar a cabo utilizando internet.

La ventaja de utilizar servicios web es que las operaciones no se realizan en el dispositivo desde el que se están ejecutando (en este caso un dispositivo móvil Android), lo que lleva a ahorrar recursos en el dispositivo.

Las operaciones realizadas por los servicios web se ejecutan en un servidor utilizando los recursos de este, enviando simplemente los resultados a la aplicación que los ejecutó.

Los servicios web RESTful se refieren a un estándar por el cual la comunicación a través de internet se lleva a cabo. Este estándar es más actual y preferido sobre estándares antiguos como SOAP, ya que la comunicación se hace de manera más rápida por ser más ligero.

A través del esquema mostrado en la Figura 2 se puede observar cómo se lleva a cabo la ejecución de un servicio web a través del estándar REST.



Figura 2. Comunicación llevada a cabo por un servicio web REST.

Como podemos observar el consumidor de servicio solicita a través de una URL la ejecución del servicio web en el proveedor del servicio que se refiere a un servidor. Posteriormente a que el proveedor del servicio finalice con las operaciones asignadas, se envía la respuesta al consumidor utilizando el protocolo HTTP.

6. Desarrollo del proyecto

6.1 Metodología empleada en el desarrollo del proyecto

Para el desarrollo del proyecto se utilizó la metodología UP (Unified Process), que se centra en etapas iterativas e incrementales.

6.2 Diseño del sistema

En las secciones posteriores se presentarán los artefactos de diseño del proyecto. Estos incluyen diagramas de casos de uso, casos de uso de texto, diagramas de clases, diagrama de arquitectura del sistema y la estructura de la base de datos.

Cabe mencionar que en algunas secciones solamente se abordarán algunos de los artefactos. La compilación completa de artefactos se encuentra en la [memoria de diseño](#).

6.2.1 Diagrama de casos de uso

En la Figura 3 se muestra el diagrama de caso de uso para el sistema.

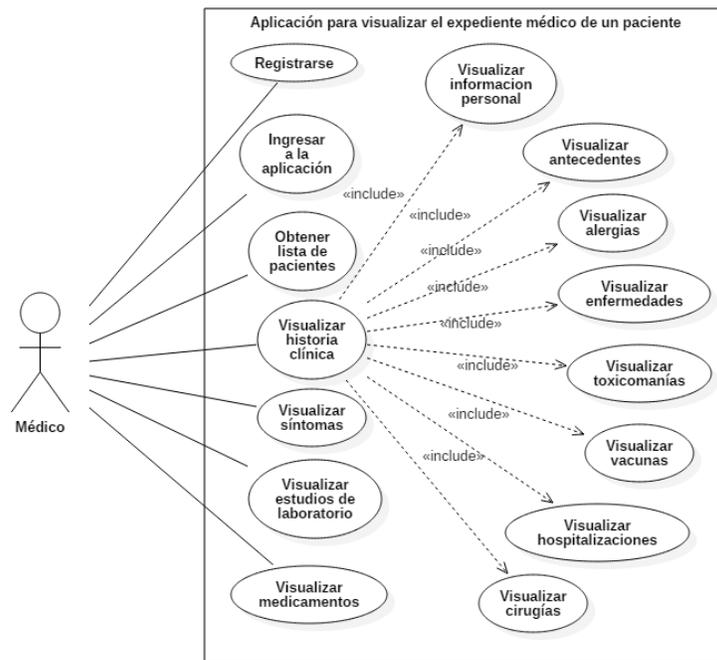


Figura 3. Diagrama de casos de uso del sistema.

A continuación, se dará una breve explicación de cada uno de los casos de uso que se presentan en la Figura 3.

Registrarse

Permite al médico llenar un formulario con sus datos personales para posteriormente utilizarlos para ingresar al sistema

Ingresar a la aplicación

Permite que un médico previamente registrado, utilice sus datos de ingreso para ser autenticado como usuario de la aplicación e ingresar para comenzar a utilizarla.

Obtener lista de pacientes

Muestra al médico la lista de pacientes que tiene asignados y permite la búsqueda de un paciente en específico.

Visualizar historia clínica

Permite al médico visualizar la historia clínica de un paciente, que en este caso se compone de las siguientes áreas:

- **Visualizar información personal:** Muestra información personal relevante para el diagnóstico de un paciente.
- **Visualizar antecedentes:** Muestra antecedentes familiares del paciente y si es que se encuentran con vida o muerte.
- **Visualizar alergias:** Muestra las alergias que presenta el paciente.
- **Visualizar enfermedades:** Muestra la lista de enfermedades relevantes que ha presentado el paciente.
- **Visualizar toxicomanías:** Muestra la lista de las toxicomanías que presenta el paciente.
- **Visualizar vacunas:** Muestra la lista de las vacunas que han sido aplicadas al paciente.
- **Visualizar hospitalizaciones:** Muestra la lista de hospitalizaciones que ha tenido un paciente.
- **Visualizar cirugías:** Muestra la lista de las cirugías a las cuales ha sido sometido el paciente.

Cabe mencionar que para visualizar esta área, el paciente debe haber dado previamente permiso de visualización al médico.

Visualizar síntomas

Presenta al médico la lista de síntomas recientes que ha presentado el paciente. También muestra la descripción de estos síntomas.

Visualizar estudios de laboratorio

Muestra al médico la lista de estudios de laboratorio que el paciente ha ingresado. Muestra también el estudio como una imagen y un resumen médico de éste.

Visualizar medicamentos

Muestra al médico la lista de medicamentos que ha consumido el paciente, así como información adicional sobre la forma de ingesta de cada medicamento en específico.

6.2.2 Casos de uso de texto

En esta sección se presentarán los casos de texto más relevantes en la aplicación.

6.2.2.1 Caso de uso para el registro de la aplicación

Identificador: 01Registrarse
Nombre: Registrarse
Descripción: El médico ingresara datos personales que serán almacenados para posteriormente utilizarlos para ingresar a la aplicación.
Actor principal: Médico
Precondiciones: <ul style="list-style-type: none">• El médico ha presionado la palabra <i>¡Regístrate!</i> desde la pantalla inicial de la aplicación.
Flujo principal: <ol style="list-style-type: none">1. El sistema despliega el formulario correspondiente al registro de usuario.2. El médico presiona el campo de texto etiquetado como <i>Nombre</i>.3. El médico ingresa su nombre cuando aparece el teclado en pantalla.4. El médico presiona el campo de texto etiquetado como <i>Apellido Paterno</i>.5. El médico ingresa su apellido paterno cuando aparece el teclado en pantalla.6. El médico presiona el campo de texto etiquetado como <i>Apellido Materno</i>.7. El médico ingresa su apellido materno cuando aparece el teclado en pantalla.8. El médico presiona el campo de texto etiquetado como <i>Cédula Profesional</i>.9. El médico ingresa su cédula profesional cuando aparece el teclado en pantalla.10. El médico presiona el campo de texto etiquetado como <i>Contraseña</i>.

11. El médico ingresa su contraseña cuando aparece el teclado en pantalla.
12. El médico presiona el campo de texto etiquetado como *Nombre de la universidad*.
13. El médico ingresa el nombre de su universidad de procedencia cuando aparece el teclado en pantalla.
14. El médico presiona el campo de texto etiquetado como *Clave*.
15. El médico ingresa la clave de su universidad cuando aparece el teclado en pantalla.
16. El médico presiona el campo de texto etiquetado como *Nombre de la especialidad*.
17. El médico el nombre de su especialidad cuando aparece el teclado en pantalla.
18. El médico presiona el campo de texto etiquetado como *Nombre de la institución*.
19. El médico ingresa el nombre de la institución donde labora cuando aparece el teclado en pantalla.
20. El médico presiona el campo de texto etiquetado como *Consultorio*.
21. El médico ingresa el consultorio donde se ubica dentro de la institución cuando aparece el teclado en pantalla.
22. El médico presiona el campo de texto etiquetado como *Estado*.
23. El médico ingresa el estado donde se localiza la institución cuando aparece el teclado en pantalla.
24. El médico presiona el campo de texto etiquetado como *Municipio*.
25. El médico ingresa el municipio donde se localiza la institución cuando aparece el teclado en pantalla.
26. El médico presiona el campo de texto etiquetado como *Colonia*.
27. El médico ingresa la colonia donde se localiza la institución cuando aparece el teclado en pantalla.
28. El médico presiona el campo de texto etiquetado como *Calle*.
29. El médico ingresa la calle donde se localiza la institución cuando aparece el teclado en pantalla.
30. El médico presiona el campo de texto etiquetado como *Número telefónico*.
31. El médico ingresa el número telefónico de la institución cuando aparece el teclado en pantalla.
32. El médico presiona el campo de texto etiquetado como *Código postal*.
33. El médico ingresa el código postal donde se localiza la institución cuando aparece el teclado en pantalla.
34. El médico presiona el botón *Crear Cuenta*.
35. El sistema muestra un diálogo indicando que se está creando la cuenta.
36. El sistema crea un diálogo indicando que la cuenta se creó correctamente

Flujo alternativo:

- En cualquier momento el médico puede presionar el botón *Crear Cuenta* sin haber ingresado los datos necesarios. En donde se le presentara al usuario un texto de ayuda en los campos que aún no han sido ingresados correctamente.
- El médico puede retroceder en la aplicación presionando el botón de *Retroceder* proporcionado por el dispositivo móvil para salir del registro sin guardar los datos.

Post-condiciones: El médico ha sido registrado en el sistema y puede dar clic en la palabra *¡Ingresar!* para comenzar a hacer uso del sistema, o bien salir de la aplicación.

Requerimientos no funcionales:

- El formulario de registro proveerá al médico con ayuda en caso de que alguno de los campos a ingresar sea incorrecto.
- Las secciones del formulario se dividirán para que sea más clara que información es la que se solicita.

6.2.2.2 Caso de uso para ingresar a la aplicación

Identificador: 01IngresarAplicación
Nombre: Ingresar a la aplicación
Descripción: El médico ingresara su información de autenticación para ingresar a la aplicación
Actor principal: Médico
Pre-condiciones: <ul style="list-style-type: none">• El médico se ha registrado previamente [01Registrarse].• El médico se encuentra en la pantalla inicial de la aplicación.
Flujo principal: <ol style="list-style-type: none">1. En la pantalla principal el sistema indica el ingreso de los datos del usuario correspondientes a <i>Cédula profesional</i> y <i>Contraseña</i>.2. El médico presiona el campo de texto etiquetado como <i>Cédula profesional</i>.3. El médico ingresa su cédula profesional cuando aparece el teclado en pantalla.4. El médico presiona el campo de texto etiquetado como <i>Contraseña</i>.5. El médico ingresa su contraseña cuando aparece el teclado en pantalla.6. El médico presiona el botón llamado <i>Ingresar</i>.7. El sistema crea un diálogo indicando que se está llevando a cabo la operación de ingresar.8. El sistema cierra el diálogo y abre una nueva pantalla de bienvenida a la aplicación.
Flujo alternativo: <ol style="list-style-type: none">6a. La cédula profesional o la contraseña del médico son incorrectas.<ol style="list-style-type: none">1. El sistema creará un dialogo indicando que se está llevando a cabo la operación de autenticación.2. El sistema indicará que hubo un problema al autenticar al médico y puede intentar ingresar al sistema de nuevo.• En cualquier momento que el médico no ingrese los datos correspondientes en los campos <i>Cédula profesional</i> o <i>Contraseña</i> el sistema creara textos de ayuda para indicar donde se encuentra el problema en estos campos.
Post-Condicion: El médico ha sido autenticado correctamente en el sistema y se le presenta la pantalla de bienvenida al sistema.
Requerimientos no funcionales: <ul style="list-style-type: none">• Crear un diálogo que indique al médico que se está llevando a cabo la operación de autenticación y no ha surgido ningún problema.

- En caso de haber un error en los campos de texto *Cédula Profesional* o *Contraseña* y el médico presione el botón *Ingresar*, indicarle en dónde está el error en estos campos.

6.2.2.3 Caso de uso para obtener lista de pacientes

Identificador: 01ObtenerListaPacientes
Nombre: Obtener lista de pacientes
Descripción: El médico podrá obtener la lista de pacientes que se le tiene asignada
Actor principal: Médico
Pre-condiciones: <ul style="list-style-type: none"> • El médico ha ingresado exitosamente al sistema [01IngresarAplicación].
Flujo principal: <ol style="list-style-type: none"> 1. El médico presiona el botón <i>Pacientes</i>. 2. El sistema muestra una pantalla con la lista de los pacientes asignados al médico. 3. El médico presiona sobre un paciente en específico. 4. El sistema presenta al médico la pantalla de resumen del paciente indicando instrucciones para visualizar sus elementos.
Flujo alternativo: <ol style="list-style-type: none"> 2.a. El médico busca a un paciente en específico. <ol style="list-style-type: none"> 1. El médico presiona el botón que tiene un ícono de lupa en la parte derecha de la <ActionBar>. 2. El sistema muestra la etiqueta <i>Search...</i> y abre el teclado en pantalla. 3. El médico hace una búsqueda de un paciente en base a su nombre. 4. El sistema muestra a los pacientes que concuerdan con la búsqueda automáticamente.
Post-condiciones: El médico obtiene la lista de los pacientes que tiene asignados.
Requerimientos no funcionales: <ul style="list-style-type: none"> • En cada uno de los pacientes, hacer más visible el nombre y resaltarlo para que sea más fácil elegir el paciente deseado.

6.2.2.4 Caso de uso para la visualización de la historia clínica

Identificador: 01VisualizarHistoria
Nombre: Visualizar historia clínica
Descripción: El médico podrá visualizar la historia clínica del paciente elegido anteriormente.
Actor principal: Médico
Pre-condiciones: <ul style="list-style-type: none"> • El médico ha elegido un paciente en específico [01ObtenerListaPacientes].
Flujo principal: <ol style="list-style-type: none"> 1. El médico presiona el botón en la parte superior izquierda de la <ActionBar>.

2. El sistema le presentará al médico un menú para visualizar los elementos del expediente médico.
3. El médico presiona sobre el elemento *Historial Clínico*.
4. El sistema presenta una pantalla donde se encuentran los elementos que conforman al historial clínico.
5. El médico presiona el botón *Consultar* para alguna de los elementos.
 - 5.1. Visualiza la información personal de un paciente [vía **01InformaciónPersonal**].
 - 5.2. Visualiza los antecedentes personales [vía **01AntecedentesPersonales**].
 - 5.3. Visualiza las alergias del paciente [vía **01Alergias**].
 - 5.4. Visualiza las enfermedades del paciente [vía **01Enfermedades**].
 - 5.5. Visualiza las toxicomanías del paciente [vía **01Toxicomanias**].
 - 5.6. Visualiza las vacunas del paciente [vía **01Vacunas**].
 - 5.7. Visualiza las hospitalizaciones de un paciente [vía **01Hospitalizaciones**].
 - 5.8. Visualiza las cirugías de un paciente [vía **01Cirugias**].

Flujo Alternativo:

- 2.a El médico no cuenta con permisos para visualizar el historial clínico.
 1. El sistema presenta el elemento de historial clínico de color opaco.
 2. El médico presiona el elemento del historial clínico.
 3. El sistema no da respuesta, indica que el médico no tiene los permisos del paciente para visualizar el elemento de historial clínico.
 - En cualquier momento el médico puede presionar el botón *Regresar* para volver a la pantalla de elección de categoría del historial clínico.

Requerimientos no funcionales:

- Indicar con un color opaco en caso de que el médico no cuente con los permisos de visualizar el historial clínico.
- Integrar íconos representativos de cada categoría del historial clínico para hacer más sencilla la visualización de las categorías.

6.2.3 Diagrama de clases

En esta sección se presentan los diagramas de clases más relevantes del sistema. Cabe mencionar que los diagramas se dividen en dos, los diagramas de la aplicación en Android y los diagramas de los servicios web.

También para facilitar el uso de los servicios web, su descripción se puede obtener en el anexo que contiene el API de los servicios.

6.2.3.1 Diagrama de clases de la aplicación Android

Debido a que la cantidad de clases era muy grande, se dividió el diagrama de clases en dos. En la Figura 4 se muestra la primera parte del diagrama de clases del paquete activities.

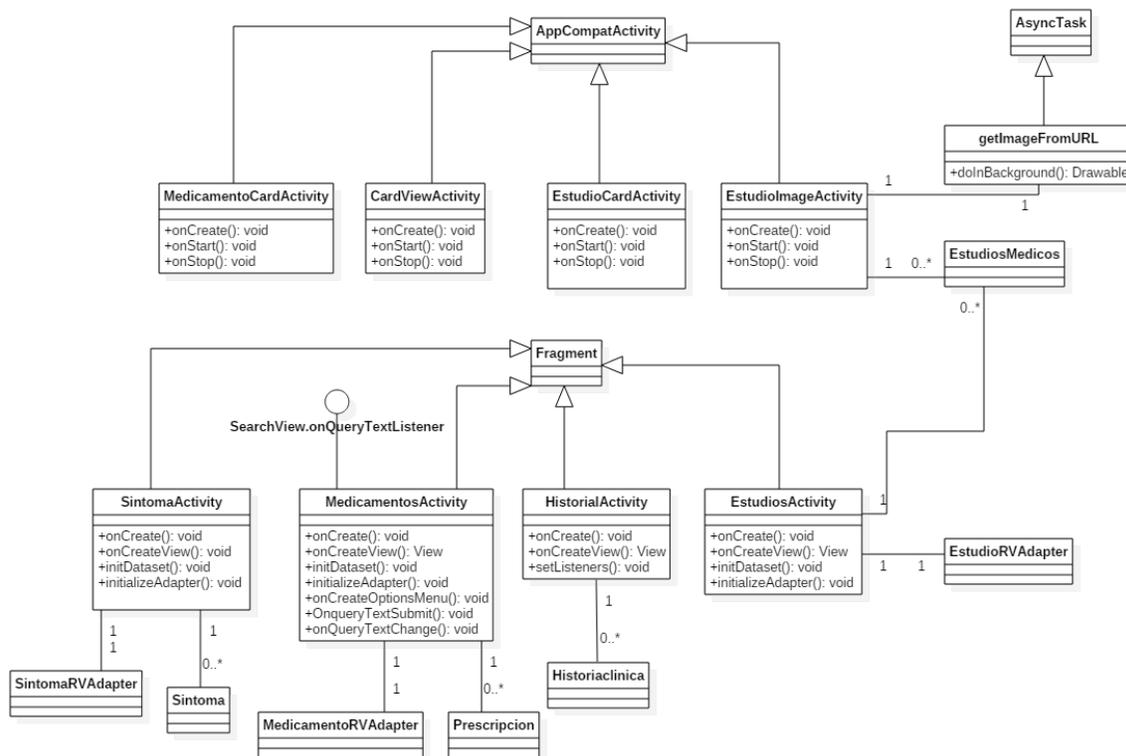


Figura 4. Diagrama de clases del paquete activities (1/2).

Como se observa en la Figura 4, las clases contenidas en ese paquete principalmente funcionan como controladores visto desde un modelo MVC (Modelo – Vista – Controlador), pues la mayoría de estas clases tienen la función de controlar las acciones del usuario sobre la interfaz gráfica de la aplicación.

En este caso, las clases **SintomaRVAdapter**, **Sintoma**, **MedicamentoRVAdapter**, **Prescripcion**, **Historiaclinica**, **EstudioRVadapter** y **EstudiosMedicos**, forman parte de paquetes externos. En la sección correspondiente en la memoria de diseño se da la descripción correspondiente.

También cabe mencionar, que para el desarrollo de la aplicación se utiliza un elemento de diseño de Android, llamado *Cards*. Este elemento se utiliza para mostrar listas de información de manera ordenada, permitiendo además resaltar fragmentos de interés de esta información. Además permite que cada *Card* sea presionada y (en este caso), presentar toda la información que se tiene sobre el elemento. También tiene como característica importante, el permitir implementar de manera fácil una interfaz de búsqueda para listas extensas.

AppCompatActivity

Como parte de Android, Google recomienda ciertos lineamientos para el diseño de las aplicaciones. Uno de estos es separar la vista(o interfaz de usuario), de las operaciones asignadas a los datos (modelo). Por esto, todas las clases que presentarán un elemento en pantalla tienen la necesidad de extender de una clase relacionada con **Activity**, en este caso, se utilizan las clases **AppCompatActivity** y **Fragment**.

Esto hace que las subclases hereden los métodos *onCreate()*, *onStart()* y *onStop()*. Estos métodos se heredan como parte del ciclo de vida de una actividad en Android, que podríamos identificar como el controlador en un modelo MVC.

El método *onCreate()*, permite hacer una serie de operaciones definidas por el usuario, antes de que la interfaz sea mostrada al usuario.

El método *onStart()*, hace operaciones justo antes de que sea mostrada la interfaz de usuario. La diferencia entre este método y *onCreate()*, es que se asegura que al finalizar el método *onStart()*, la interfaz se estará mostrando.

El método *onStop()*, permite realizar operaciones cuando la actividad junto con su interfaz de usuario ya no se encuentra visible.

Fragment

Esta es una variante de una actividad en Android que sirve el mismo propósito como **AppCompatActivity**, la diferencia es que los fragmentos son un espacio en la pantalla de un dispositivo, que puede ser sustituido por distintas actividades, sin cambiar otros elementos que no sean los definidos para el fragmento.

Esto es especialmente útil en dispositivos con una pantalla grande, pues se pueden mantener elementos que se utilicen constantemente en un área fija de la pantalla y que la información se cargue dinámicamente en otra área. Así la navegación en distintos módulos de la aplicación se hace más fácil ya que siempre esta visible ésta área.

Este tipo de diseño se utiliza para visualizar el expediente de un paciente, ya que este expediente se compone de cuatro módulos: historial, síntomas, medicamentos y estudios de laboratorio. Por esto, se mantiene un área de navegación en la parte lateral donde el médico puede acceder de manera fácil a cada uno de los módulos y, la información se carga dinámicamente en lo que resta de la pantalla sin afectar al área de navegación.

Así todas las clases que extienden de **Fragment**, tendrán el comportamiento descrito anteriormente.

MedicamentoCardActivity

Esta clase funciona como presentador de cada una de las *Cards* que compondrán la lista de los elementos en pantalla. La clase encargada de la creación de la listas, utilizará esta clase para poder presentar cada una de los medicamentos contenidos en la lista en la interfaz gráfica.

CardViewActivity

Se utiliza para presentar las *Cards* que componen a la lista de los pacientes asignados a un médico. La clase que se encarga de obtener la lista de los pacientes, utiliza esta clase para poder crearlos y presentarlos en pantalla.

EstudioCardActivity

Esta clase se utiliza para presentar las *Cards* que componen la lista de estudios de laboratorio de un paciente. La clase encargada de obtener la lista, utiliza esta clase para poder presentar los elementos en pantalla.

EstudioImageActivity

Esta clase se encarga de mostrar los estudios de laboratorio como imágenes en pantalla. Como parte del diseño se consideró que los estudios de laboratorio se componen de imágenes de dicho estudio. Esta clase presenta al estudio en pantalla, además de los datos relevantes junto con una descripción provista por el médico. Mediante su método *onCreate()* se obtiene el objeto de tipo **EstudiosMedicos** para ser visualizado.

getImageFromURL

Como lineamientos de desarrollo en Android, no se puede realizar operaciones de red en el hilo principal de una aplicación. Por esto, se utiliza la clase **getImageFromURL**, que se encarga de obtener la dirección URL de la imagen de un estudio, acceder a ella y presentarla

en la pantalla de este dispositivo. La clase hereda de **AsyncTask**, para que se realice en un hilo de procesamiento distinto al de **EstudioImageActivity**, y sea procesado correctamente.

Utiliza también la clase de EstudiosMedicos que representa el estudio de laboratorio, para obtener la información de cada estudio y presentarla.

EstudiosActivity

Esta clase se encarga de obtener los estudios de laboratorio de un paciente y presentarlos en forma de *Cards* en la pantalla. El uso de la clase **EstudioRVAdapter** es parte del uso de las *Cards* y se dará su descripción en la sección correspondiente.

HistorialActivity

Esta clase se utiliza para obtener el historial clínico del paciente y utilizar la clase **Historiaclinica**, para guardar la información del historial para posteriormente ser presentada. También muestra los distintos elementos que conforman al historial y da la posibilidad de acceso a cada uno de ellos para visualizarlos de manera individual.

MedicamentosActivity

Muestra en pantalla la lista de los medicamentos que ha consumido un paciente. Obtiene esta lista de medicamentos y la inicializa a través del método *initDataset ()*, también inicializa el adaptador de las *Cards* para posteriormente presentarlas en pantalla a través del método *initializeAdapter()*.

Para facilitar la navegación de la lista de los medicamentos, se provee la posibilidad de buscar y filtrar los medicamentos en base a su nombre. Para poder realizar esto, la clase implementa la interfaz provista por Android **SearchView.onQueryTextListener**. Así el método *onCreateOptionsMenu()*, hace que se presente la barra de búsqueda en la parte superior de la pantalla, *onQueryTextSubmit()*, hace que la lista se filtre al presionar el botón de buscar en el teclado y *onQueryTextChange()*, hace que la lista se filtre automáticamente al estar escribiendo el nombre de algún medicamento, mostrando las *Cards* que concuerden con el término de búsqueda.

SintomaActivity

Presenta la lista de síntomas que tiene un paciente, en forma de *Cards*. Utiliza la clase **Sintoma** para obtener cada objeto y su información correspondiente y también utiliza la clase **SintomaRVAdapter** para hacer uso de las *Cards*.

El método *initDataset()* obtiene la lista de los síntomas y los convierte en objetos de tipo **Sintoma**, para que el adaptador **SintomaRVAdapter**, los utilice posteriormente para presentarlos en forma de *Cards*.

En la Figura 5 podemos observar las clases restantes en el paquete y sus relaciones. Aquí podemos observar como al igual que en la Figura 4, todas las clases que son actividades extienden a la clase **AppCompatActivity**.

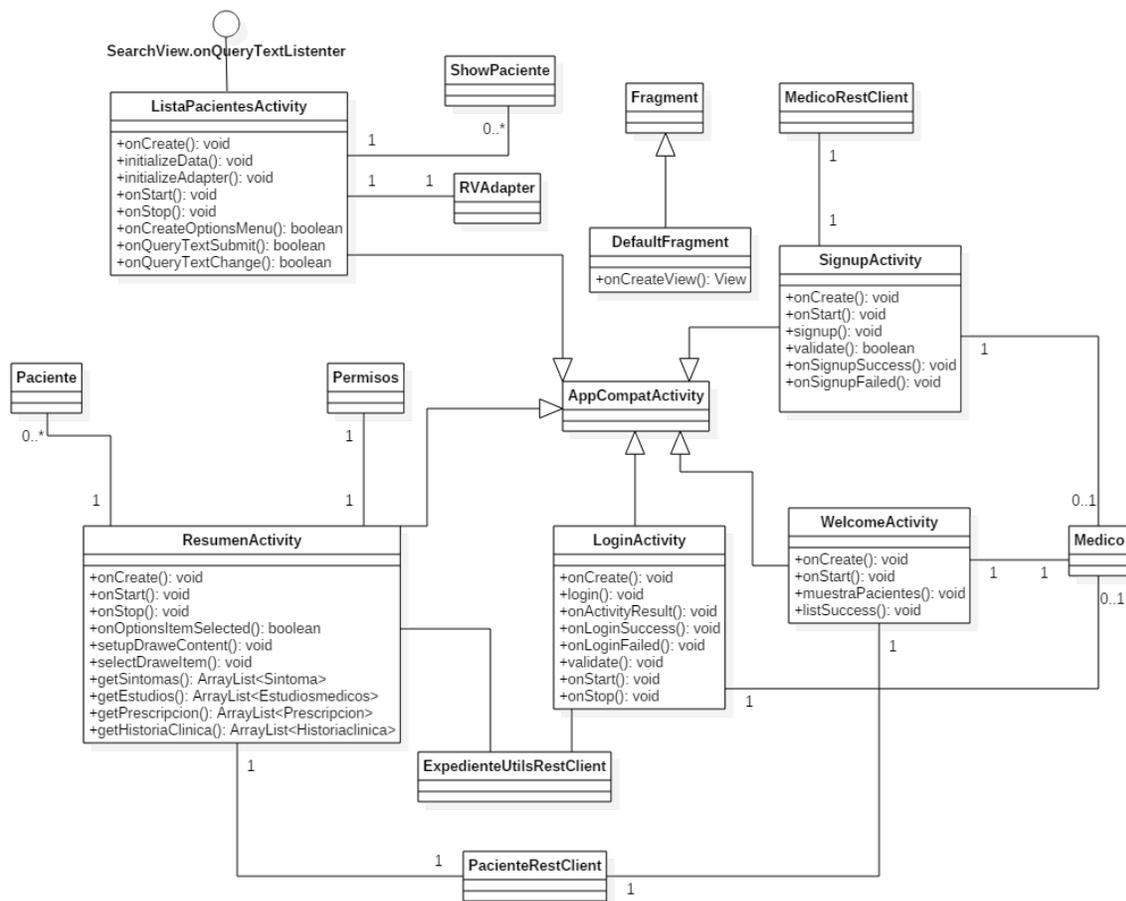


Figura 5. Diagrama de clases del paquete actividades (2/2).

También tenemos que las clases **Paciente**, **PacienteRestClient**, **Permisos**, **RVAdapter**, **ShowPaciente**, **Medico**, **MedicoRestClient** y **ExpedienteUtilsRestClient**, son clases que se encuentran definidas en otros paquetes, por esto, su definición y resumen de su funcionamiento serán descritos en la memoria de diseño.

A partir de la Figura 5, el funcionamiento de las clases y sus relaciones es el siguiente.

ListaPacientesActivity

Su principal objetivo es el obtener la lista de pacientes asignados a un médico y mostrarlas en pantalla en forma de *Cards*.

El método *initializeData()* inicializa una lista de objetos de tipo **ShowPaciente** que muestra de manera rápida el paciente y su identificación para ser seleccionado.

El método *initializeAdapter()* inicializa el adaptador de tipo **RVAdapter** que utilizan las *Cards* para ser mostradas en pantalla, se inicializa con los datos obtenidos de la lista de los pacientes.

La clase **ListaPacientesActivity** hace la implementación de la interfaz **SearchView.onQueryTextListener** para poder realizar la implementación de la función de búsqueda en la lista de pacientes. Al implementar la interfaz mencionada, se realiza la implementación de los métodos *onCreateOptionsMenu()*, cuya función es asignar una barra de búsqueda en la parte superior de la interfaz, *onQueryTextSubmit()* que realiza el filtrado de la lista de los pacientes al presionar el botón de buscar en el teclado y *onQueryTextChange()* que realiza el filtrado automáticamente cuando el texto en la barra de búsqueda cambia.

ResumenActivity

La función principal de esta clase es mostrar el área principal para la visualización del expediente médico. Esto se refiere a crear el espacio donde los fragmentos (que extienden a la clase **Fragment** como se observa en la Figura 4 y Figura 5 van a ser cargados, además de definir el área de navegación.

A través del método *onCreate()* recibe el objeto **Paciente** y **Permisos** previamente obtenidos para ser mostrados.

Para indicar el área de navegación para el expediente se utilizan los métodos *setUpDraweContent()*, para mostrar los elementos de la navegación, *selectDraweItem()* crea los fragmentos que se cargaran en el área en específico dependiendo de la selección en la navegación y *onOptionsItemSelected()* indica la inicialización del menú de navegación.

También provee los métodos para enviar los objetos de tipo específico a cada uno de los módulos que componen el expediente médico. Es decir, los métodos *getSintomas()*, *getEstudios()*, *getPrescripcion()*, *getHistoriaClinica()*, envían las listas de objetos de tipo **Sintoma**, **Estudiosmedicos**, **Prescripcion** e **HistoriaClinica**, para ser utilizados por los módulos correspondientes.

DefaultFragment

Este es una actividad que extiende a la clase **Fragment** para ser cargada por omisión al ser inicializada la interfaz que provee **ResumenActivity**. Es solamente un mensaje para indicar la manera de acceso a cada uno de los elementos del expediente médico.

SignupActivity

La función principal es el presentar la pantalla de registro de médicos para el uso de la aplicación, además de ofrecer las operaciones para registrarlo a través de un servicio web.

El método *signup()* es el que se encarga de realizar el proceso de registro. Dentro de este método se utiliza también *validate()* que hace una revisión de los datos ingresados en los campos para el registro, en caso de haber algún error, el método hace aparecer en la interfaz gráfica una pequeña ayuda para el campo en el cual se está teniendo el error.

Hace uso de la clase **MedicoRestClient** para realizar la llamada al servicio web que registra a los usuarios, en caso de ser exitosa la llamada se accede al método *onSignupSuccess()* que envía el objeto de tipo **Medico** para ser mostrado a través de la clase **WelcomeActivity**. En caso de que la llamada falle, se utiliza el método *onSignupFailed()*, que se encarga de mostrar un mensaje en pantalla sobre el error en cuestión.

WelcomeActivity

Su función es que a partir de un objeto de tipo **Medico**, se muestran los datos para confirmar el acceso a la aplicación, además de dar la opción de visualizar la lista de pacientes del médico.

El método *muestraPacientes()* hace uso de la clase **PacienteRestClient** para hacer la llamada al servicio web para obtener la lista de los pacientes. En caso de ser exitosa se envían tanto la lista de los pacientes como los permisos para ser utilizados por la clase **ListaPacientesActivity** a través del método *listSuccess()*.

LoginActivity

La función principal de esta clase es la autenticación de un médico para el uso de la aplicación a través de su usuario y contraseña

La autorización se realiza en el método *login()* que a su vez se apoya en los métodos *validate()* para verificar que los datos ingresados en los campos de usuario y contraseña tengan las características adecuadas y de *onActivityResult()* que utiliza la clase **ExpedienteUtilsRestClient** para hacer la llamada al servicio web de autenticación y obtener si el usuario y contraseña son correctos.

En caso de que el acceso sea exitoso, se utiliza el método *onLoginSuccess()* que obtiene la instancia de la clase **Medico** que representa al usuario y es enviada a la clase **WelcomeActivity**. En caso de que el acceso no sea exitoso, se utiliza el método *onLoginFailed()* que se encarga de mostrar un mensaje referente al error en pantalla

6.2.3.2 Diagrama del paquete webServices de los servicios web

La importancia de las clases presentadas en la Figura 6 radica en que en ellas se encuentran los métodos que se ponen a disposición a través de los servicios web REST. La clase **ApplicationConfig** es creada automáticamente por Netbeans para registrar las clases que funcionaran para brindar los servicios web.

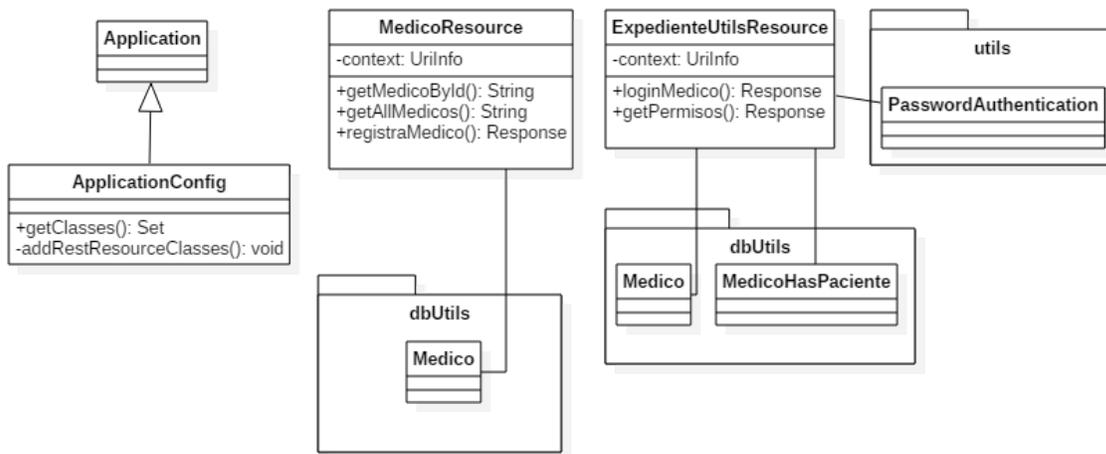


Figura 6. Clases contenidas en el paquete webServices (1/2).

Tanto la clase **MedicoResource** y **ExpedienteUtilsResource** tienen relación con clases importantes de paquetes externos como podemos observar en la Figura 6. Ambas clases ofrecen métodos accesibles a través de los servicios web.

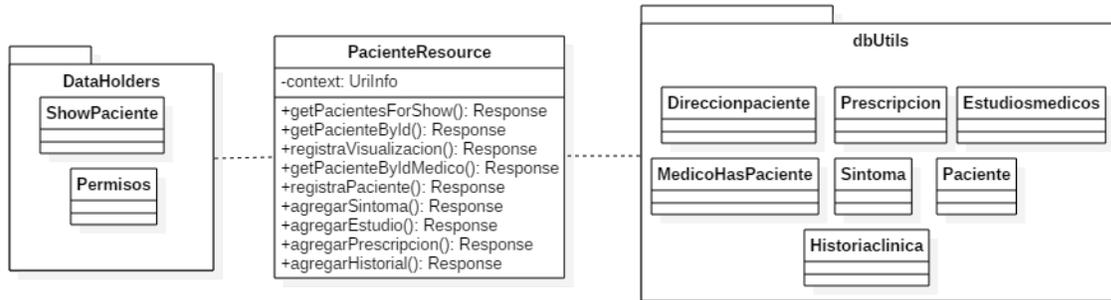


Figura 7. Clases contenidas en el paquete webServices (2/2).

En la Figura 7 se muestra la clase **PacienteResource**. Esta clase es de las más utilizadas pues brinda una gran cantidad de métodos para consultar la información del paciente. También cabe mencionar la gran cantidad de clases con las que se relaciona en los paquetes DataHolders y dbUtils.

La descripción de los métodos ofrecidos por las clases **MedicoResource**, **ExpedienteUtilsResource** y **PacienteResource**, se encuentran en el JavaDoc adjunto a este reporte. Ahí se pueden consultar los parámetros, el retorno y el objetivo de cada uno de los métodos.

6.2.4 Arquitectura del sistema

Para el diseño del sistema se utilizó el patrón MVC (Modelo – Vista – Controlador) que esencialmente indica la separación de la interfaz de usuario (vista) de la lógica de la aplicación (modelo) utilizando un controlador para procesar las solicitudes entre el modelo y la vista.

Se puede dividir el desarrollo del proyecto en dos categorías, la implementación de la aplicación en la plataforma Android y la implementación de los servicios web REST.

En este caso podemos considerar que la vista de la aplicación se realiza en la plataforma Android presentando la interfaz al médico para visualizar el expediente médico de un paciente. Dentro de la aplicación también consideramos al controlador como las Activities (que se explicaron en la sección de diagramas de clases) como el intermediario entre la interfaz y los servicios web.

También podemos considerar el uso de los servicios web como el modelo de la aplicación, ya que estos son los encargados de obtener los objetos y las representaciones de la base de datos, así como brindar las operaciones necesarias sobre estos objetos.

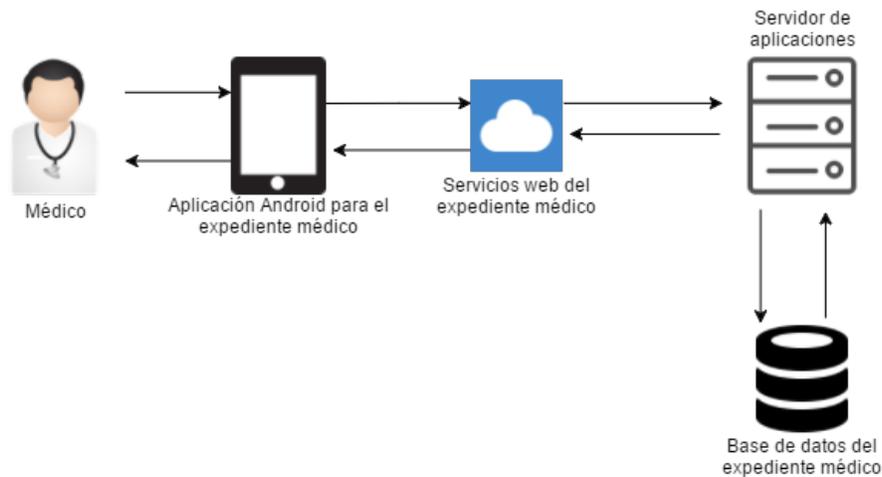


Figura 8. Arquitectura general del proyecto

Como observamos en la Figura 8, el proyecto se compone de dos categorías. La aplicación para visualizar el expediente médico de un paciente en Android y la implementación de los servicios web para acceder a la información del expediente.

En base a esta figura, podemos observar como el médico se encargará de interactuar con la aplicación en un dispositivo Android. Esta interacción implicará la obtención de datos que se realizará a través de los servicios web implementados para el expediente médico que serán alojados en un servidor de aplicaciones.

También en este servidor se alojará la base de datos del expediente, sobre la cual los servicios web realizarán las consultas necesarias para presentar la información solicitada por el médico en la aplicación.

6.2.5 Estructura de la base de datos

En base a la Figura 9 podemos observar el modelo entidad relación de la base de datos de la aplicación del expediente médico. Se observa que se creó una base de datos completa con las tablas referentes a todos los módulos de la aplicación (medicamentos, síntomas, estudios e historial clínico). Esto se realizó con la idea de que aplicaciones futuras que utilicen datos de alguno de estos módulos puedan utilizar las tablas para ofrecer una única base de datos para las aplicaciones y estandarizar los datos a ser almacenados.

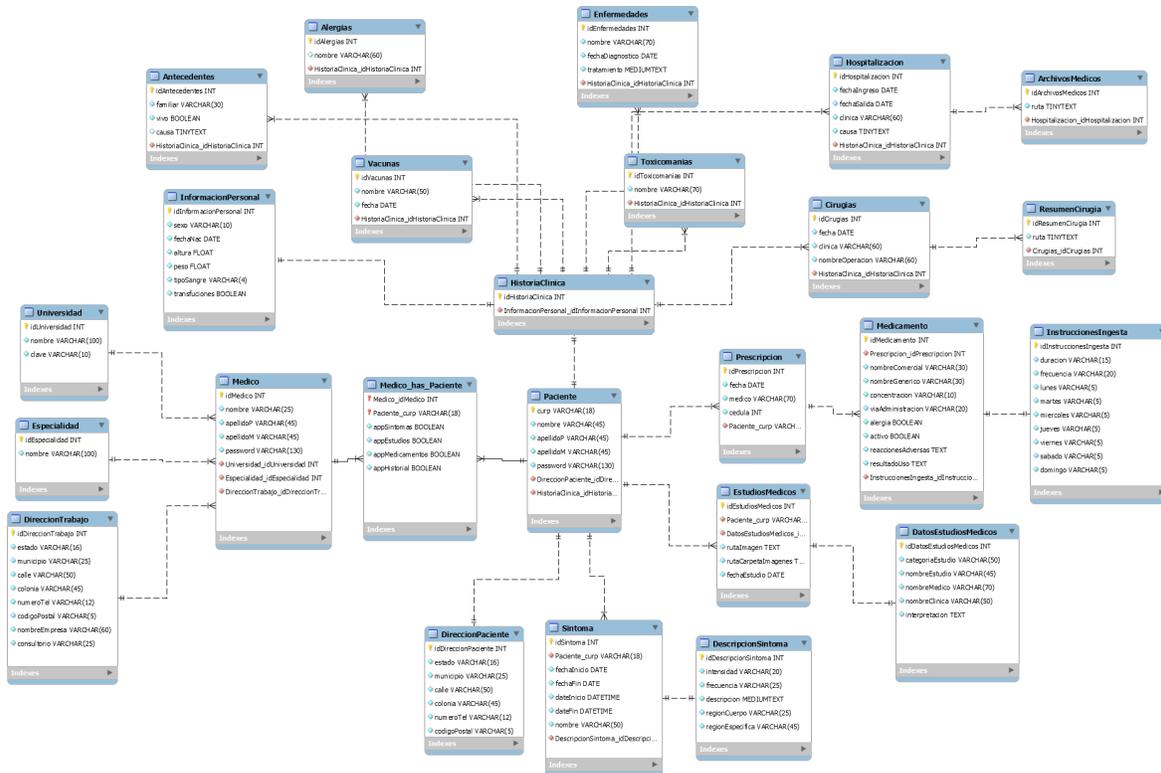


Figura 9. Diagrama entidad - relación de la base de datos del sistema

También se agregó la entidad **Paciente** como un usuario, así otras aplicaciones pueden implementar la autenticación única de los usuarios de las aplicaciones para poder compartir información con los médicos.

A continuación se explica la estructura de cada una de las tablas en base a la Figura 9. Cabe señalar que los campos marcados en negritas son las llaves foráneas.

Tabla Medico

Es la entidad que representa al usuario médico como el actor principal de la aplicación Android. Se consideran datos personales para su registro, así como usuario y contraseña para su autenticación. Utiliza las entidades **Universidad**, **Especialidad** y **DirecciónTrabajo** para completar la información necesaria para su representación. Su estructura es la siguiente:

- Nombre: Primer nombre del médico.
- apellidoP: Apellido paterno del médico.
- apellidoM: Apellido materno del médico.
- idMedico: Se considera la cedula profesional como identificador único del médico así como su usuario para ingresar a la aplicación.
- Password: Contraseña para ingresar a la aplicación, se guarda una cadena encriptada en base a la contraseña ingresada por el médico.

- **Universidad:** Identificador de la universidad a la que perteneció el médico.
- **Especialidad:** Identificador de la última especialidad del médico.
- **DireccionTrabajo:** Identificador de la dirección de trabajo del médico.

Tabla Universidad

Es la entidad que representa la institución en la cual el médico obtuvo su último grado académico. Su estructura es la siguiente:

- idUniversidad: Identificador interno para cada registro.
- nombre: Nombre de la institución educativa.
- clave: Clave de registro de la institución ante algún órgano educativo (e.j. SEP).

Tabla Especialidad

Nombre de la última especialidad obtenida por el médico, considerado así ya que la mayoría de los médicos solamente identifican la última especialidad obtenida. También se puede considerar el caso de que la especialidad sea general, para representar a un médico sin especialidad (médico general). Su estructura es la siguiente:

- idEspecialidad: Identificador interno para cada registro.
- nombre: Nombre de la especialidad obtenida.

Tabla DireccionTrabajo

Entidad que representa el lugar de trabajo de un médico. Su estructura es la siguiente:

- idDireccionTrabajo: Identificador interno para cada registro.
- estado: Estado de la republica donde se encuentra el lugar de trabajo.
- Municipio: Municipio donde se encuentra el lugar de trabajo.
- Calle: Calle donde se encuentra el lugar de trabajo.
- Colonia: Colonia donde se encuentra el lugar de trabajo.
- numeroTel: Número telefónico correspondiente a su lugar de trabajo.
- codigoPostal: Código postal de su dirección de trabajo.
- nombreEmpresa: Nombre de la institución donde trabaja.
- consultorio: Numero o identificador de consultorio donde se encuentra dentro de su lugar de trabajo.

Tabla Medico_has_Paciente

Esta entidad es creada por la relación muchos a muchos que existen entre las entidades **Medico** y **Paciente**. Esta relación se crea ya que un médico puede tener muchos pacientes que han dado su permiso para ser visualizados por él, y al mismo tiempo, un paciente puede tener muchos médicos a los que ha otorgado su permiso para visualizarlos.

También considerando la privacidad del paciente, se agregan algunos campos que indicarán que módulos del expediente médico un paciente ha otorgado permisos para visualizar a un médico.

La estructura es la siguiente:

- **Medico:** Identificador del medico.
- **Paciente:** Identificador del paciente.
- **appSintomas:** Indica si es que el médico tiene permisos para visualizar los síntomas del paciente.
- **appEstudios:** Indica si es que el médico tiene permisos para visualizar los estudios de laboratorio de un paciente.
- **appMedicamentos:** Indica si es que el médico tiene permisos para visualizar los medicamentos consumidos por un paciente.
- **appHistorial:** Indica si es que el médico tiene permisos para visualizar el historial clínico del paciente.

Tabla Paciente

Entidad que representa a un paciente para ser visualizado por el médico, o bien, para poder ser registrado como usuario de aplicaciones externas a la descrita en este proyecto. La estructura es la siguiente:

- **Curp:** Clave otorgada por el gobierno a cada ciudadano utilizada como identificador único de cada uno de los pacientes. También utilizada como el nombre de usuario de las aplicaciones.
- **Nombre:** Primer nombre del paciente.
- **apellidoP:** Apellido paterno del paciente.
- **apellidoM:** Apellido materno del paciente.
- **password:** Contraseña del paciente para las aplicaciones. Se realiza mediante una cadena encriptada en base a una cadena proporcionada por el paciente en su registro.
- **DireccionPaciente:** Entidad de la dirección de residencia de este paciente.
- **HistoriaClinica:** Historia clínica que presenta este paciente.

Tabla DireccionPaciente

Entidad que representa la dirección de residencia de un paciente. Su estructura es la siguiente:

- **idDireccionPaciente:** Identificador interno para cada registro.
- **estado:** Estado de la republica donde reside el paciente.
- **municipio:** Municipio de la republica donde reside el paciente.
- **Calle:** Nombre de la calle donde reside el paciente.
- **Colonia:** Colonia donde reside el paciente.
- **numeroTelefono:** Número de teléfono de contacto del paciente.
- **codigoPostal:** Código postal del lugar de residencia del paciente.

Tabla Sintoma

Entidad que representa un síntoma que presenta el paciente. Se compone además de **DescripcionSintoma** para una explicación más detallada del síntoma. Su estructura es la siguiente:

- idSintoma: Identificador interno para cada registro.
- **Paciente:** Identificador del paciente al cual corresponde este síntoma.
- fechaInicio: Fecha de inicio del síntoma del paciente.
- fechaFin: Fecha de finalización del síntoma de un paciente.
- dateInicio: Fecha y hora de inicio del síntoma de un paciente.
- dateFin: Fecha y hora de finalización del síntoma de un paciente.
- nombre: Nombre que se le asigna al síntoma presentado.
- **DescripcionSintoma:** Identificador de la descripción de este síntoma.

Tabla DescripcionSintoma

Descripción más detallada sobre un síntoma en específico. Tiene una relación uno a uno con la entidad **Sintoma**. Su estructura es la siguiente:

- idDescripcionSintoma: Identificador interno para cada registro.
- intensidad: Intensidad del síntoma presentado.
- frecuencia: Intervalo de tiempo en el que se presenta el síntoma.
- descripción: Descripción del síntoma ingresada por el paciente.
- regionCuerpo: Región general del cuerpo donde se presenta el síntoma.
- regionEspecífica: Región específica del cuerpo donde se presenta el síntoma.

Tabla EstudiosMedicos

Entidad que representa el estudio de laboratorio de un paciente. Se considera que los estudios están guardados en un servidor para poder ser obtenidos a través de su ruta de acceso. También se compone de una descripción más detallada del estudio. Su estructura es la siguiente:

- idEstudiosMedicos: Identificador interno para cada registro.
- **Paciente:** Identificador del paciente al cual es asignado este estudio de laboratorio.
- **DatosEstudiosMedicos:** Identificador de la entidad de datos de estudios médicos a la cual corresponde este estudio médico.
- rutaImagen: Ruta específica del estudio del laboratorio en el servidor.
- rutaCarpetaImágenes: Ruta de la carpeta de las imágenes donde se alojan los estudios del paciente en el servidor.
- fechaEstudio: Fecha en la que se realizó el estudio de laboratorio.

Tabla DatosEstudiosMedicos

Entidad que representa una descripción más detallada sobre un estudio de laboratorio en específico. Tiene una relación uno a uno con la entidad **EstudiosMedicos**. Su estructura es la siguiente:

- idDatosEstudiosMedicos: Identificador interno para cada registro.
- categoriaEstudio: Categoría del estudio de laboratorio.
- nombreEstudio: Nombre específico del estudio realizado.
- nombreMedico: Nombre del médico a cargo de realizar el estudio de laboratorio.
- nombreClinica: Nombre de la clínica donde se realizó el estudio de laboratorio.
- interpretación: Interpretación del estudio de laboratorio proporcionada por un médico.

Tabla Prescripcion

Entidad que presenta la prescripción emitida por un médico para el consumo de medicamentos, así como las instrucciones de su consumo. Tiene una relación uno a muchos con la entidad **Medicamento** para representar la lista de medicamentos asignados en una prescripción. Su estructura es la siguiente:

- idPrescripcion: Identificador interno para cada registro.
- fecha: Fecha de emisión de la prescripción médica.
- medico: Nombre del médico que emitió la prescripción.
- cedula: Cedula profesional del médico que emitió la prescripción.
- **Paciente:** Identificador del paciente al cual es asignada esta prescripción.

Tabla Medicamento

Entidad que representa a un medicamento en específico y su ingesta por parte del paciente. Su estructura es la siguiente:

- idMedicamento: Identificador interno para cada registro.
- **Prescripcion:** Identificador de la prescripción a la que pertenece cada medicamento.
- nombreComercial: Nombre comercial del medicamento.
- nombreGenerico: Nombre de la fórmula genérica del medicamento.
- concentración: Nivel de concentración del medicamento a consumir.
- viaAdministracion: Vía de administración del medicamento (oral, intravenosa, etc.).
- alergia: Valor que representa si el paciente ha presentado una alergia al medicamento.
- activo: Valor que representa si actualmente el paciente se encuentra consumiendo el medicamento en cuestión.
- reaccionesAdversas: En caso de haber existido reacciones adversas, este campo es utilizado para describir las reacciones presentadas por el paciente.

- resultadoUso: Este campo se utiliza para que el paciente ingrese su experiencia con el tratamiento y los resultados de su uso.
- **InstruccionesIngesta:** Identificador de instrucciones de ingesta para este medicamento.

Tabla InstruccionesIngesta

Entidad que representa las instrucciones provistas por el médico para la ingesta del medicamento por parte del paciente. Tiene una relación uno a uno con **Medicamento** para representar la instrucción de ingesta para un medicamento en específico. Su estructura es la siguiente:

- idInstruccionesIngesta: Identificador interno para cada registro.
- duración: Tiempo por el cual estará activo el tratamiento.
- frecuencia: Frecuencia en la ingesta del medicamento.
- lunes: Cantidad de medicamento a consumir el día lunes.
- martes: Cantidad de medicamento a consumir el día martes.
- miércoles: Cantidad de medicamento a consumir el día miércoles.
- jueves: Cantidad de medicamento a consumir el día jueves.
- viernes: Cantidad de medicamento a consumir el día viernes.
- sábado: Cantidad de medicamento a consumir el día sábado.
- domingo: Cantidad de medicamento a consumir el día domingo.

Tabla HistoriaClinica

Entidad que representa la historia clínica de un paciente. Esta a su vez se compone de distintos módulos representados por las entidades **InformacionPersonal**, **Antecedentes**, **Vacunas**, **Alergias**, **Enfermedades**, **Toxicomanias**, **Hospitalizacion** y **Cirugia**. Su estructura es la siguiente:

- idHistoriaClinica: Identificador interno para cada registro.
- **InformacionPersonal:** Identificador de la información personal contenida en la historia clínica.

Tabla InformacionPersonal

Entidad que representa información personal relevante para el diagnóstico médico. Su estructura es la siguiente:

- idAntecedentes: Identificador interno para cada registro.
- Sexo: Sexo del paciente.
- fechaNac: Fecha de nacimiento del paciente.
- altura: Altura del paciente en metros.
- peso: Peso del paciente en kilogramos.
- tipoSangre: Tipo de sangre del paciente (O, O+, etc.).

- transfusiones: Valor que representa si es que el paciente ha recibido transfusiones sanguíneas.
- **HistoriaClínica:** Identificador de la historia clínica a la cual pertenece esta entidad.

Tabla Antecedentes

Entidad que representa los antecedentes familiares de un paciente, para conocer su estado de salud o causas de muerte. Su estructura es la siguiente:

- idAntecedentes: Identificador interno para cada registro.
- familiar: Parentesco del familiar (Padre, Madre, etc.).
- vivo: Valor que indica si el familiar está vivo o ha fallecido.
- causa: En caso de que el familiar haya fallecido, se indica la causa de muerte.
- **HistoriaClínica:** Identificador de la historia clínica a la cual pertenece esta entidad.

Tabla Vacunas

Entidad que representa las vacunas que se ha aplicado el paciente. Su estructura es la siguiente:

- idVacunas: Identificador interno para cada registro.
- nombre: Nombre de la vacuna aplicada.
- fecha: Fecha en que se aplicó la vacuna.
- **HistoriaClínica:** Identificador de la historia clínica a la cual pertenece esta entidad.

Tabla Enfermedades

Entidad que representa las enfermedades relevantes que ha presentado un paciente. Su estructura es la siguiente:

- idEnfermedades: Identificador interno para cada registro.
- nombre: Nombre de la enfermedad sufrida.
- fechaDiagnostico: Fecha en que se diagnosticó esta enfermedad.
- tratamiento: Tratamiento indicado por el médico para combatir la enfermedad.
- **HistoriaClínica:** Identificador de la historia clínica a la cual pertenece esta entidad.

Tabla Toxicomanias

Entidad que representa la toxicomanía de un paciente. Una toxicomanía se define como la dependencia física y psicológica de una sustancia. La estructura de esta entidad es la siguiente:

- idToxicomania: Identificador interno para cada registro.
- nombre: Nombre de la toxicomanía padecida.

- **HistoriaClinica:** Identificador de la historia clínica a la cual pertenece esta entidad.

Tabla Hospitalizacion

Entidad que representa una hospitalización sufrida por un paciente por alguna causa. La estructura de la entidad es la siguiente:

- idHospitalizacion: Identificador interno para cada registro.
- fechaIngreso: Fecha en la cual se ingresó a la institución médica.
- fechaSalida: Fecha en la cual se salió de la institución médica.
- clínica: Nombre de la institución en la cual el paciente fue ingresado.
- causa: Resumen de la causa por la cual el paciente sufrió la hospitalización.
- **HistoriaClinica:** Identificador de la historia clínica a la cual pertenece esta entidad.

Tabla ArchivosMedicos

Entidad que representa a los archivos médicos proporcionados por un médico como resultado de una hospitalización en formato digital. Tiene una relación uno a muchos con **Hospitalizacion** para representar la lista de archivos derivados de la hospitalización. Su estructura es la siguiente:

- idArvhivosMedicos: Identificador interno para cada registro.
- ruta: Ruta del archivo médico en el servidor.
- **Hospitalizacion:** Identificador de la hospitalización a la cual pertenece esta entidad.

Tabla Cirugias

Entidad que representa las cirugías que ha sufrido el paciente. Su estructura es la siguiente:

- idCirugias: Identificador interno para cada registro.
- fecha: Fecha en la cual se realizó la cirugía.
- clínica: Nombre de la clínica en la cual se realizó la cirugía.
- nombreOperacion: Nombre del procedimiento quirúrgico que se realizó.
- **HistoiaClinica:** Identificador de la historia clínica a la cual pertenece esta entidad.

Tabla ResumenCirugias

Entidad que representa a los archivos que contienen el resumen de la cirugía de un paciente. Esta entidad cuenta con una relación uno a muchos con **Cirugias** para representar la lista de los archivos que representan los resúmenes de cirugía. Su estructura es la siguiente:

- idResumenCirugia: Identificador interno para cada registro.
- ruta: Ruta en donde se encuentra el archivo del resumen dentro del servidor.
- **Cirugias:** Identificador de la cirugía a la cual pertenece este entidad.

6.3 Uso del sistema

A continuación se presentan los casos de uso más importantes de la aplicación. El lector puede consultar el [manual de usuario](#) para consultar todas las funcionalidades del sistema.

6.3.1 Registro de usuario

Partiendo desde la pantalla principal de la aplicación, el usuario presiona sobre el texto que dice *¡Regístrate!*

Posteriormente se visualizará la pantalla de registro para el uso de la aplicación, como se muestra en la Figura 10.



Figura 10. Pantalla de registro de la aplicación.

Como se muestra en la Figura 10, los datos de registro se dividen en cuatro grupos: datos personales, datos de la cuenta, datos de la universidad y datos laborales.

Al presionar sobre cada uno de los campos que se muestran en la Figura 10, el sistema abrirá el teclado para que el usuario pueda ingresar los valores deseados.

Cuando el usuario haya finalizado de ingresar los valores solicitados en cada uno de los campos, presionará sobre el botón *Crear Cuenta* donde se mostrará una pequeña ventana que indicará que la operación se está llevando a cabo como se muestra en la Figura 11.

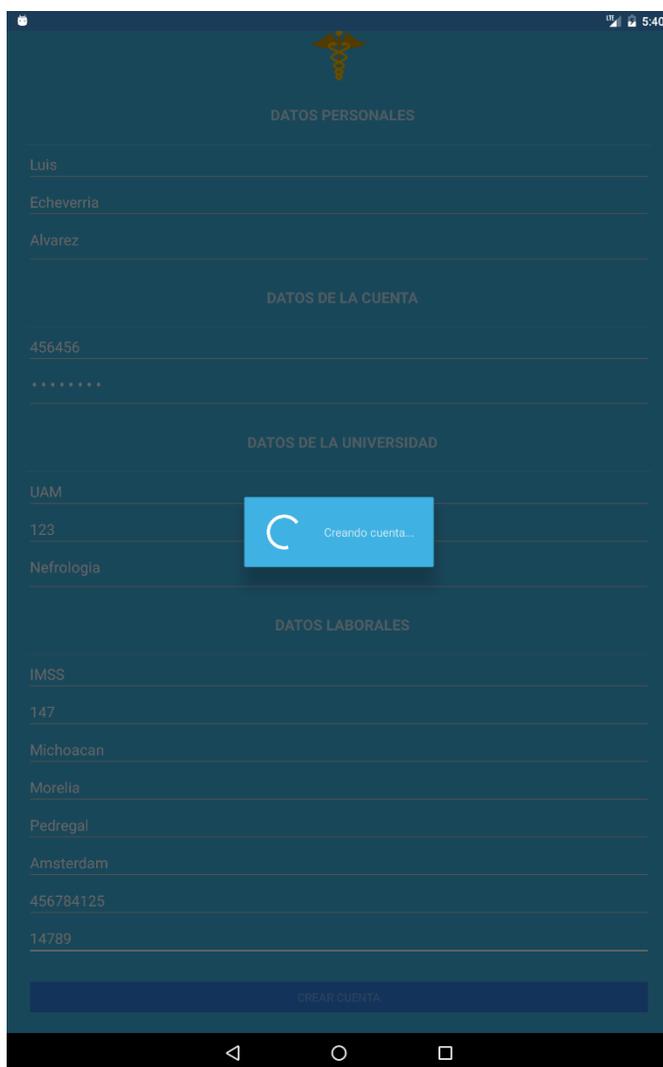


Figura 11. Pantalla de operación de registro

Esta pantalla se mantendrá mientras se realizan las operaciones necesarias. En caso de que el registro sea exitoso, se mostrará un mensaje en pantalla y se llevará al usuario a la pantalla principal.

En caso de que exista algún error en alguno de los campos, un pequeño texto de ayuda aparecerá junto al campo que presenta el problema para que el usuario pueda llenarlo más fácilmente.

En la Figura 12 se muestra un ejemplo del texto de ayuda para el usuario cuando se deja en blanco el campo de *Nombre*.



The image shows a registration form with two input fields: 'Nombre' and 'Apellido Paterno'. The 'Nombre' field is highlighted with a red border and a red exclamation mark icon, indicating an error. A tooltip with the text 'Ingrese su nombre' is displayed next to the error icon.

Figura 12. Texto de ayuda para el registro

6.3.2 Ingreso a la aplicación

Para comenzar a utilizar las funciones de la aplicación es necesario que el usuario se autentifique mediante su cédula profesional y su contraseña. La pantalla de ingreso a la aplicación es la misma que la pantalla principal.

En esta pantalla el usuario debe ingresar sus credenciales de acceso presionando sobre los campos correspondientes, que se indican como *Cédula Profesional* y *Contraseña*.

Cuando el usuario ingrese la información correspondiente en ambos campos se le presentará una pequeña ventana indicando que se está llevando a cabo la operación, como se muestra en la Figura 13.

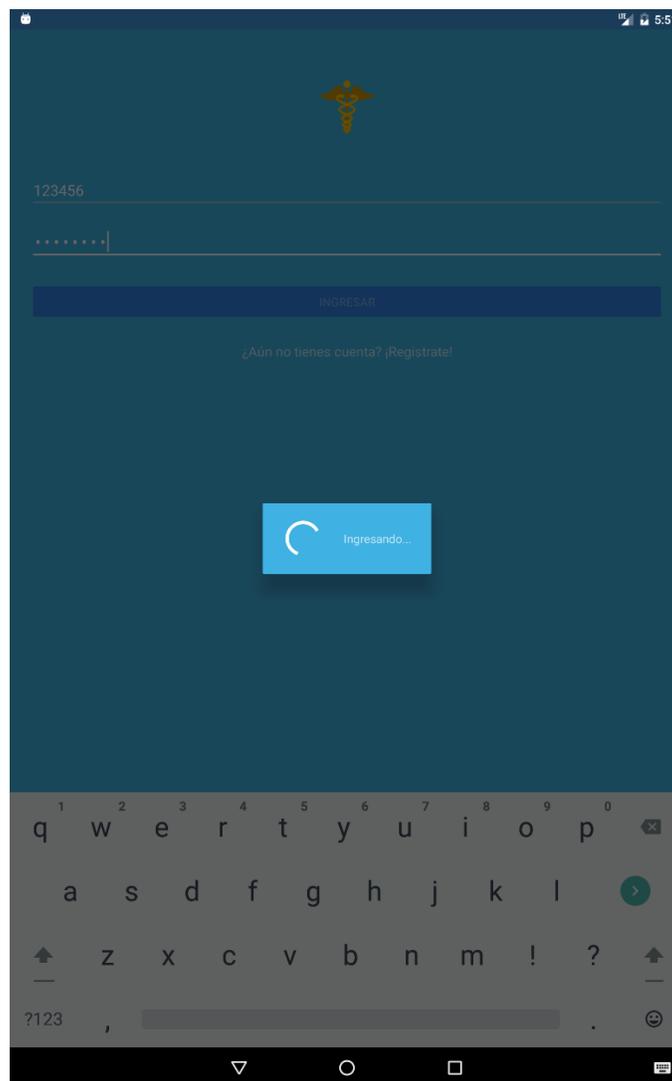


Figura 13. Pantalla de operación de autenticación.

La pantalla mostrada en la Figura 13 se mantendrá mientras se esté llevando a cabo la operación de autenticación.

En caso de exista un error en las credenciales de acceso, se le presentará al usuario un pequeño mensaje en la parte inferior de la pantalla que se muestra en la Figura 14.



Figura 14. Texto de error al ingresar a la aplicación.

Cuando el ingreso a la aplicación se realice de manera exitosa, la pantalla de la aplicación cambiará a la pantalla de bienvenida.

Esta pantalla se presenta como retroalimentación al usuario para confirmar su acceso exitoso al sistema. En ésta se muestra un pequeño resumen de la información del usuario además de proveer la capacidad de mostrar la lista de los pacientes asignados, cuyo funcionamiento se presentará en su sección correspondiente.

La pantalla de bienvenida se muestra en la Figura 15.

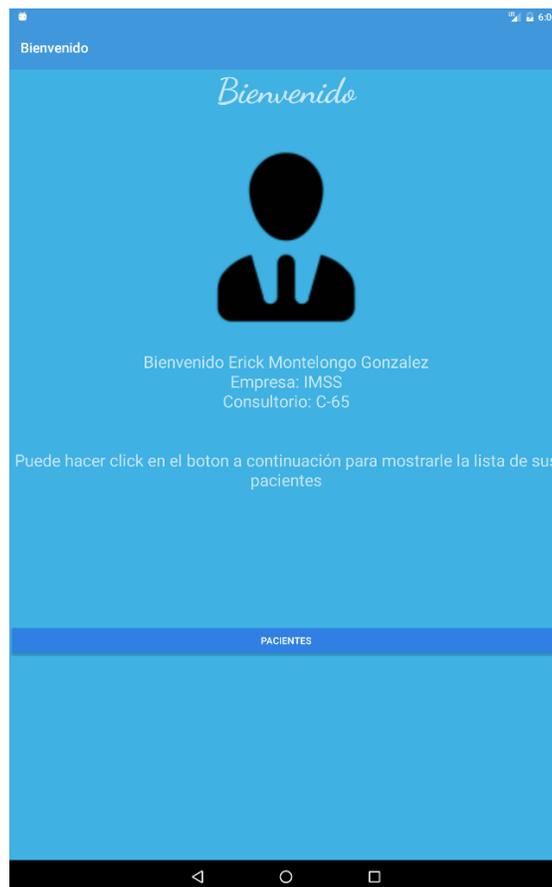


Figura 15. Pantalla de bienvenida a la aplicación.

6.3.3 Visualización y búsqueda de la lista de pacientes

Por la naturaleza de la aplicación, el usuario tiene asignado una lista de pacientes que previamente han dado los permisos para visualizar los distintos módulos del expediente médico. La cantidad de pacientes podría ser muy grande, por lo que también se tiene la posibilidad de realizar una búsqueda por nombre del paciente.

Para poder obtener la lista de pacientes asignados, se procede desde la pantalla de bienvenida mostrada en la Figura 15, donde el usuario presiona sobre el botón *Pacientes*.

A continuación se mostrará una pequeña pantalla indicando que se está realizando la operación de obtener los pacientes. Esta pantalla se muestra en la Figura 16.

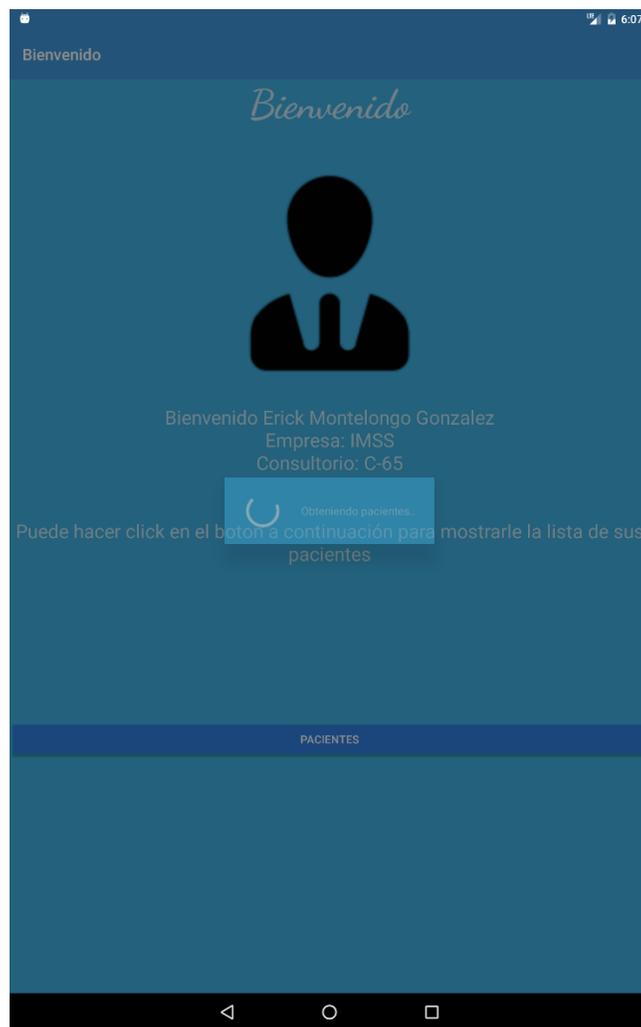


Figura 16. Pantalla de operación para obtener lista de pacientes.

La pantalla mostrada en la Figura 16 se mantendrá el tiempo necesario para que se obtengan todos los pacientes asignados al usuario. El tiempo que se tarde en obtener la lista de

pacientes será proporcional a la cantidad de pacientes que el usuario tenga asignado, es decir, entre más pacientes se tengan asignados más tiempo se tardará en que se obtenga la lista.

Al finalizar la obtención de la lista se le presentará al usuario la pantalla mostrada en la Figura 17 donde se encuentra un pequeño resumen de cada uno de sus pacientes.

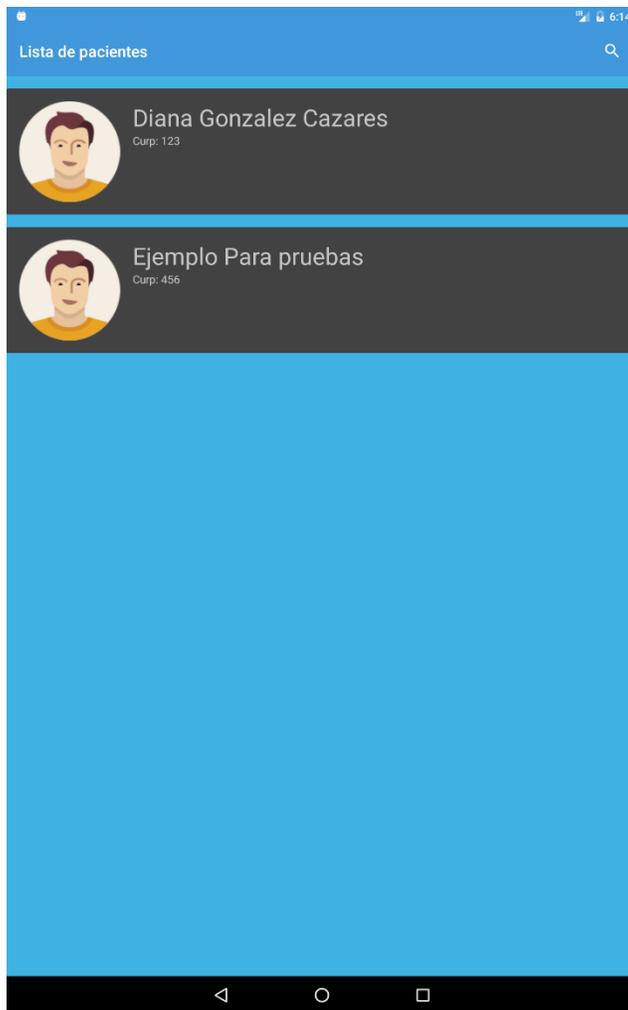


Figura 17. Pantalla de visualización de lista de pacientes.

En la lista de pacientes mostrada en la Figura 17 también se puede realizar la búsqueda de un paciente en base a su nombre. Para realizar la búsqueda el usuario debe presionar sobre el ícono de lupa en la parte superior derecha de la pantalla. Al presionar sobre este ícono se abrirá una barra de texto como se muestra en la Figura 18 donde el usuario puede ingresar el nombre del paciente al que desee buscar.

La búsqueda realizada se realiza de manera automática sin que el usuario requiera presionar el ícono de *Buscar* en el teclado y se realiza mientras se está escribiendo en la barra de texto.

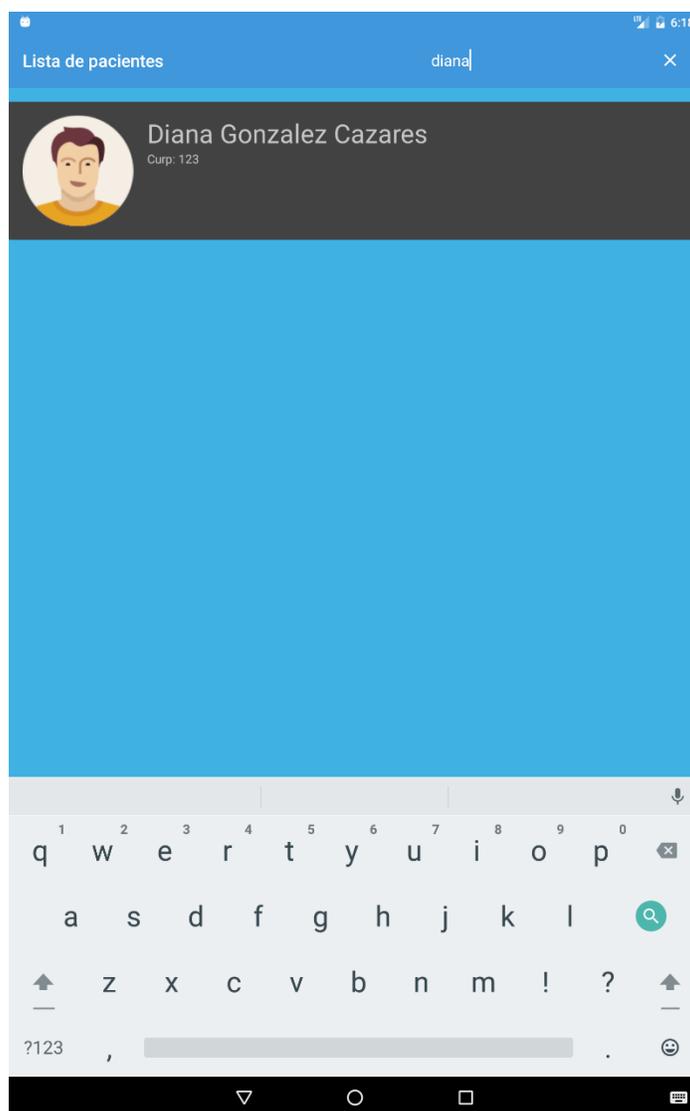


Figura 18. Búsqueda de un paciente.

Ahora bien para comenzar la visualización del expediente médico de un paciente, el usuario debe presionar sobre la parte oscura del paciente que desee visualizar. Cuando el usuario presione sobre el paciente deseado, se llevará a la pantalla principal de visualización del expediente que se muestra en la Figura 19.



Figura 19. Pantalla de resumen de expediente.

6.3.4 Visualización de historia clínica

Para comenzar con la visualización de la historia clínica del paciente, se comienza desde que el usuario se encuentra en la pantalla principal de visualización del expediente médico del paciente, mostrada en la Figura 19.

Para acceder al módulo de la historia clínica el usuario presiona sobre el ícono de menú en la parte superior izquierda, mismo que se muestra en la Figura 20.



Figura 20. Ícono de menú lateral del expediente médico

Al presionar este ícono se abrirá el menú lateral para la navegación entre cada uno de los módulos del expediente médico, como se muestra en la Figura 21.

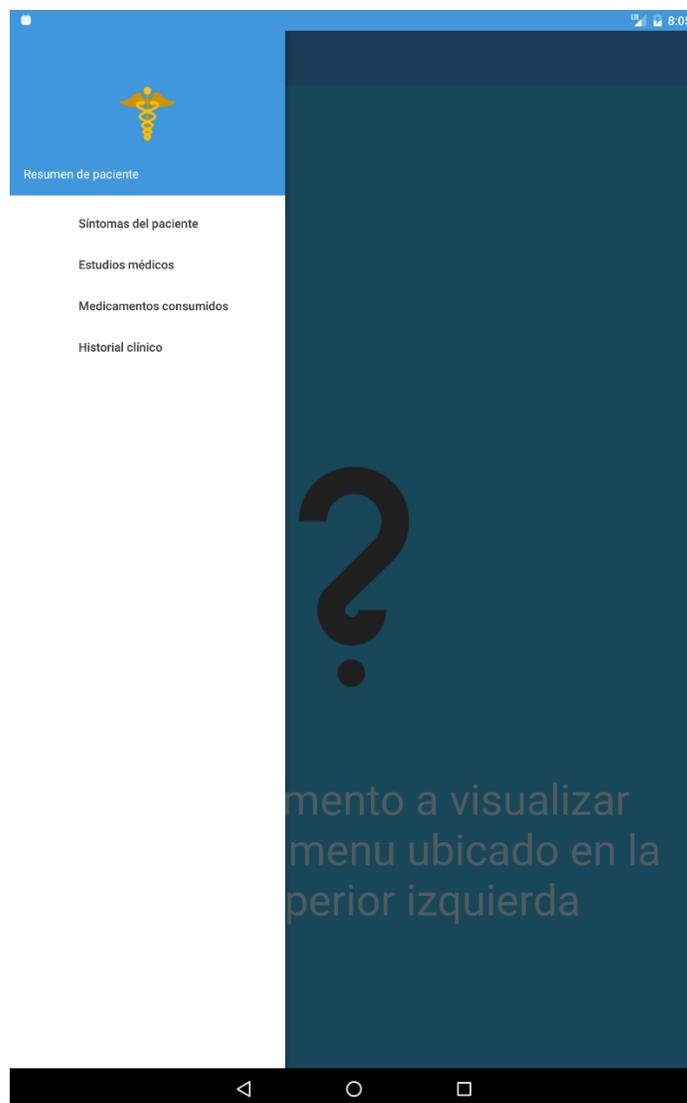


Figura 21. Menú lateral de navegación del expediente médico.

Cuando se presente este menú, el usuario puede acceder a los módulos indicados por el mismo. En este caso se requiere visualizar la historia clínica del paciente, por lo que el usuario debe presionar sobre el texto que indica *Historial Clínico*.

Al presionar sobre este botón se le presentará al usuario la pantalla principal de visualización de la historia clínica del paciente, mostrada en la Figura 22.



Figura 22. Pantalla principal del historial clínico del paciente.

En base a la Figura 22 se puede acceder a cada uno de los módulos que componen el historial clínico de un paciente, estos módulos son: información personal, antecedentes personales, alergias, enfermedades, toxicomanías, vacunas, hospitalizaciones y cirugías.

Para acceder a cada uno de los módulos el usuario presiona sobre el botón *Consultar* que acompaña a cada ícono de los módulos.

6.4 Hardware y software necesarios

A continuación se describe el hardware y software que fueron necesarios tanto para el desarrollo del proyecto como para la puesta en marcha de la aplicación. El lector puede consultar el [manual técnico](#) para mayor información.

6.4.1 Hardware

El hardware utilizado para el desarrollo de la aplicación y para pruebas del servidor fue un equipo de cómputo de escritorio con las siguientes características:

- Procesador: Intel Core i5 4670K @ 3.8GHz.
- Memoria RAM: 8Gb DDR3.
- Disco Duro: SSD 128GB y 1TB HDD.
- Tarjeta de video: Nvidia GTX 750 Ti.

Para realizar las pruebas de la aplicación Android se utilizó un hardware emulando una tablet con las siguientes características:

- Tamaño de pantalla de 10”.
- Procesador Dual Core a 1.7 GHz.
- Wi-Fi 802.11 b/g/n.
- 3 Gb RAM.

6.4.2 Software

El software utilizado para el desarrollo de la aplicación fue el siguiente:

- Sistema Operativo: Windows 10.
- Lenguaje de programación: [Java versión 8](#).
- IDE para desarrollo en Android: [Android Studio 2.2.3](#).
- IDE para el desarrollo de servicios web: [NetBeans 8.2](#).
- Framework para el mapeo de objetos – entidades de base de datos: [Hibernate 4.3.x](#).
- Framework para el desarrollo de servicios web REST: [Jersey 2.5.1](#).
- API para llamadas asíncronas a métodos HTTP en Android: [Android Asynchronous Http Client](#).
- API para convertir una estructura JSON en objetos java: [Jackson 2.7.8](#).
- Servidor de aplicaciones: [Tomcat 8.5.11](#).
- Editor visual para el modelado de la base de datos: [MySQL Workbench](#).

- Manejador de base de datos: [MySQL Community Server 5.7](#).
- Script para la construcción de la base de datos.

La versión mínima del sistema operativo Android fue 6.0 Marshmallow pero versiones posteriores a ésta pueden funcionar correctamente.

El software necesario para poner en marcha el servidor de aplicaciones es el siguiente:

- Sistema operativo: Windows o Linux.
- Servidor Tomcat 8.5.11.
- MySQL Community Server 5.7.
- Script de la base de datos del sistema¹.

¹ El archivo que contiene el script sql se anexa al documento en digital.

7. Resultados

Al realizar este proyecto se obtuvieron productos de software que fueron divididos en dos grupos, aquellos que se refieren a la aplicación en Android para la visualización del expediente médico de un paciente y los obtenidos al implementar los servicios web para el expediente médico en el servidor.

Los módulos obtenidos por el diseño e implementación de la aplicación Android para visualizar el expediente médico son los siguientes:

- Módulo para el registro del médico.
- Módulo para la autenticación del médico.
- Módulo para la obtención de la lista de pacientes del médico.
- Módulo para la visualización de los síntomas de un paciente.
- Módulo para la visualización de los estudios de laboratorio de un paciente.
- Módulo para la visualización de los medicamentos consumidos por un paciente.
- Módulo para la visualización de la historia clínica de un paciente.

Los módulos obtenidos por el diseño e implementación de los servicios web del expediente médico son los siguientes:

- Módulo para el registro de médicos y pacientes.
- Módulo para obtención y registro de los síntomas de un paciente.
- Módulo para la obtención y registro de estudios de laboratorio de un paciente.
- Módulo para la obtención y registro de medicamentos consumidos por un paciente.
- Módulo para la obtención y registro de la historia clínica de un paciente.

Cabe mencionar que para la obtención de la información por parte de los servicios web, se diseñó e implementó una base de datos para guardar la información de los usuarios tanto pacientes como médicos y los datos que conforman el expediente médico.

8. Análisis y discusión de resultados

En esta sección se describen las distintas características del sistema, así como las pruebas realizadas para comprobar el funcionamiento correcto del proyecto.

Para la prueba del rendimiento del servidor se utilizaron los siguientes datos:

- Registro de cinco pacientes distintos en la base de datos con dos distintos médicos asignados.
- Registro de dos síntomas distintos para los pacientes.
- Registro de un historia clínica para los pacientes.
- Registro de tres estudios de laboratorio para los pacientes.
- Registro de dos medicamentos distintos para cada paciente.
- En el área del historia clínica se ingresaron distinta cantidad de datos a cada uno de los pacientes registrados.

Con la información en la base de datos se pudo realizar la visualización desde la aplicación Android utilizando los servicios web para obtener dicha información. Se realizaron las siguientes acciones en la aplicación Android:

- Registro de cinco médicos.
- Autenticación de dos médicos distintos.
- Visualización del expediente médico de dos pacientes distintos.

8.1 Discusión de resultados

En base a las pruebas que se realizaron se tiene que tanto la aplicación Android, los servicios web y la base de datos funcionan conforme a lo esperado.

El registro de los médicos se realiza de manera exitosa utilizando el servicio web de registro.

La autenticación se realizó exitosamente utilizando las credenciales de acceso del médico. Esta información se guarda de manera segura en la base de datos pues las contraseñas son transformadas en cadenas encriptados.

La obtención de la lista de los pacientes se realizó exitosamente, aunque conforme aumente la cantidad de pacientes que tenga el médico, el tiempo que le toma a la aplicación obtener la lista completa aumenta.

La visualización del expediente médico se realiza correctamente ya que todas las áreas accesibles para el médico pueden ser visualizadas sin problema.

9. Conclusiones

El objetivo general de este proyecto fue el desarrollar una aplicación móvil para visualizar el expediente médico de un paciente, a saber: historial médico, medicamentos consumidos, síntomas presentados y estudios realizados, utilizando servicios web.

Basándonos en este objetivo se cumplió de manera exitosa, pues se logró el desarrollo de la aplicación en Android para la visualización del expediente médico, que a su vez utiliza una base de datos y servicios web para obtener dicha información.

A continuación se analizan los objetivos particulares y la medida en que se cumplieron:

1. Diseñar e implementar un servicio web para registrar a los usuarios que deseen hacer uso del expediente médico de un paciente.

Se cumplió completamente ya que se obtuvo el módulo para el registro de usuarios, que contempla tanto al médico como al paciente.

2. Diseñar e implementar servicio web para la invocación de los servicios web de las aplicaciones existentes.

Este objetivo se cumplió completamente, pues se realizó el desarrollo de los servicios web necesarios para obtener la información de cada uno de los elementos de los cuales se compone el expediente médico.

3. Diseñar e implementar una base de datos para guardar la información del médico que corresponderá a su información personal y la relación de pacientes que tiene.

Este objetivo se cumplió completamente ya que se obtuvo la base de datos del expediente médico. Además se extendieron sus características para poder almacenar información de cada uno de los módulos del expediente médico para estandarizar la información a almacenar.

4. Diseñar e implementar una aplicación móvil para que el médico pueda visualizar el expediente médico de sus pacientes.

Este objetivo se cumplió completamente pues se obtuvo la aplicación Android para visualizar el expediente médico de un paciente.

9.1 Experiencia al realizar el proyecto

Al realizar el proyecto uno de los grandes retos que se presentó al inicio fue el aprender una nueva tecnología para el desarrollo de aplicaciones. En este caso fue necesario el aprendizaje de desarrollar aplicaciones bajo la plataforma Android.

También durante el desarrollo de la aplicación surgieron algunos problemas al intentar utilizar características que Android no permite o bien, tiene una manera distinta de utilizarlas. Por esto también fue necesario la investigación y aprendizaje de distintas APIs que pudieran resolver los problemas que se iban presentado.

Gracias al tiempo dedicado y la constante investigación fue que se logró la implementación de este proyecto. Con esto obtuve una amplia experiencia al poder trabajar con tecnologías nuevas que seguramente utilizare en un futuro.

Otra área importante que considero fue reforzada fue en el área de diseño de sistemas, gracias al desarrollo del proyecto puse en práctica conocimientos adquiridos a lo largo de mis estudios, además de comprender mejor el uso y beneficios de cada uno de los artefactos de diseño realizados.

10. Perspectivas del proyecto

Si bien el proyecto cumple con los objetivos planteados, hay características que se pueden extender o mejorar para ayudar a la aplicación a crecer y ser más útil para los usuarios.

Las características que propongo para extender la aplicación o mejorarla son las siguientes:

- Pedir ayuda de un experto en diseño gráfico para mejorar la interfaz gráfica de la aplicación.
- Implementar un sistema de mensajes entre los pacientes y el médico. De esta manera ambos podrían estar en constante comunicación de ser necesario.
- Implementar características extras para el médico como filtrar a los pacientes bajo distintos campos para que sea aún más fácil su búsqueda.
- Explotar los datos que se almacenan en la base de datos para obtener estadísticas y apoyar aún más al diagnóstico de un paciente.

12. Referencias bibliográficas

- [1] F. García, «Prototipo de una aplicación móvil para la gestión de síntomas de un paciente,» Universidad Autónoma Metropolitana Azcapotzalco, Ciudad de México, 2014.
- [2] M. Juárez, «FotoMed: Prototipo de aplicación Android para la gestión de los estudios de un paciente,» Universidad Autónoma Metropolitana Azcapotzalco, Ciudad de México, 2015.
- [3] J. Valle, «Prototipo de aplicación Android para la gestión de medicamentos,» Universidad Autónoma Metropolitana Azcapotzalco, Ciudad de México, 2014.
- [4] J. Trinidad, «Aplicación Móvil para la Administración del Historial Médico,» Universidad Autónoma Metropolitana Azcapotzalco, Ciudad de México, 2015.
- [5] Google Inc., «Google Play Store,» Computer Software, 29 Mayo 2013. [En línea]. Available: <https://play.google.com/store/apps/details?id=com.klouddata.mmr.main&hl=en>. [Último acceso: 18 Marzo 2017].
- [6] Google Inc., «Google Play Store,» Computer Software, 2 Agosto 2015. [En línea]. Available: <https://play.google.com/store/apps/details?id=com.hyax.mymedical&hl=en>. [Último acceso: 18 Marzo 2017].
- [7] J. B. A. N. a. J. W. N. Kalogriopoulos, «Electronic Medical Record Systems for Developing Countries: Review,» *IEEE Engineering in Medicine and Biology Society*, pp. 1730 - 1733, 2009.
- [8] T. P. a. I. M. C. Doukas, «Mobile Healthcare Information Management utilizing Cloud Computing and Android OS,» *Engineering in Medicine and Biology Society (EMBC)*, pp. 1037 - 1040, 2010.
- [9] M. Corporation, «HealthVault,» Software Company, Octubre 2007. [En línea]. Available: <https://www.healthvault.com/mx/es>. [Último acceso: 18 Marzo 2017].
- [10] Oracle Corporation, «Java,» Software Company, [En línea]. Available: <https://www.java.com/es/>. [Último acceso: 19 Marzo 2017].
- [11] Google Inc., «Android Developers,» Software Company, [En línea]. Available: <https://developer.android.com/guide/components/activities.html>. [Último acceso: 19 Marzo 2017].

[12] Oracle Corporation, «Java EE,» Software Company, [En línea]. Available: <http://docs.oracle.com/javaee/6/tutorial/doc/gijvh.html>. [Último acceso: 19 Marzo 2017].

13. Apéndice A. Entregable: Listado del API del código fuente de los servicios web desarrollados

Debido a que el API generado a través de javaDoc se presenta a través de archivos HTML para visualización en navegadores, se presenta un enlace para la visualización más ordenada de este contenido a continuación.

Presionar sobre [este enlace](#) para visualizar la documentación generada del código fuente.

14. Apéndice B. Entregable: Listado del API del código fuente de la aplicación

Debido a que el API generado a través de javaDoc se presenta a través de archivos HTML para visualización en navegadores, se presenta un enlace para la visualización más ordenada de este contenido a continuación.

Presionar sobre [este enlace](#) para visualizar la documentación generada del código fuente.