

Asistencia, Seguimiento y Desarrollo de App Catrep

Lina Fernanda Menjura Rodríguez

Juan Carlos Gómez Méndez

Universidad de Boyacá

Facultad de Ciencias e Ingeniería

Ingeniería de Sistemas

Tunja

2021

Asistencia, Seguimiento y Desarrollo de App Catrep

Lina Fernanda Menjura Rodríguez

Juan Carlos Gómez Méndez

Trabajo de Grado de Semillero de Investigación para optar al título de
Ingeniero(a) de Sistemas

Directora

Carmen Constanza Uribe Sandoval

Ingeniera de Sistemas

Universidad de Boyacá

Facultad de Ciencias e Ingeniería

Ingeniería de Sistemas

Tunja

2021

Nota de aceptacion

Firma presidente del jurado

Firma del jurado

Tunja, 16 de noviembre de 2021

“Únicamente el graduando es responsable de las ideas expuestas en el presente trabajo”.

(Universidad de Boyacá. Reglamento Estudiantil de la Universidad de Boyacá del 02 de junio de 2011, Artículo 97).

Este proyecto está dedicado en especial a Dios, porque gracias a Él tuvimos la sabiduría, la inteligencia, la entrega y la voluntad para poder desarrollar cada parte del proyecto. En segunda instancia a nuestros padres, porque de no haber sido por su esfuerzo de sacarnos adelante, su constante e incondicional apoyo y sus palabras de ánimo para luchar y alcanzar nuestros sueños, hoy día no estaríamos presentando este proyecto. También a la Universidad de Boyacá, la cual fue nuestra guía permanente para su desarrollo, brindándonos los conocimientos y los recursos necesarios para solucionar y sacar adelante nuestro proyecto, y a todos aquellos ingenieros, amigos y compañeros que nos aportaron su granito de arena para el continuo desarrollo del proyecto.

Agradecimientos

Agradecemos primeramente a Dios por permitirnos sacar adelante el proyecto y por nunca dejarnos rendir en este transcurso. Gracias a los ingenieros del programa de Ingeniería de Sistemas porque gracias a sus enseñanzas y sus ideas se realizó un correcto diseño, implementación y control de nuestro prototipo. Gracias a la Ingeniera Carmen Constanza Uribe Sandoval porque gracias a sus enseñanzas y constante orientación y apoyo en el desarrollo de las etapas del prototipo pudimos dar término correcto a nuestro proyecto.

Contenido

	Pág.
Introducción.....	13
Medición de Signos Vitales.....	14
Requisitos de la aplicación	14
Razonamiento detrás de los requisitos.....	14
Windows.....	14
Android Studio	15
KOTLIN	15
JAVA.....	15
Android	17
Configuración del Entorno	18
Configuración y Desarrollo de Proyectos	19
Construcción, depuración y pruebas	19
Publicación	19
Instalación Android Estudio	19
SDK de Java	19
Android Studio	20
Diseño del prototipo de la aplicación	22
Objetivos del sistema	22
Gestión del usuario.....	22
Gestión de alertas	22
Gestión de Signos Vitales	22
Requisitos funcionales del sistema	22
Requisitos no funcionales del sistema	24
Desarrollo de la Aplicación	27
Desarrollo HUAWEI	27
Acceso al dispositivo.....	27
Adquisición de datos	27
El Prototipo.....	31

Iniciar Sesión.....	31
Home	32
Frecuencia Cardiaca	33
Oxígeno en la sangre	34
Conexión Pulsera.....	34
Perfil	36
Cerrar sesión.....	37
Conclusiones.....	38
Recomendaciones	39
Referencias	40
Anexos	41

Lista de Tablas

	Pág.
Tabla 1. Selección Pulsera.....	16
Tabla 2. Requisito funcional 1.....	23
Tabla 3. Requisito funcional 2.....	23
Tabla 4. Requisito funcional 3.....	23
Tabla 5. Requisito funcional 4.....	23
Tabla 6. Requisito funcional 5.....	24
Tabla 7. Requisito funcional 6.....	24
Tabla 8. Requisito funcional 7.....	24
Tabla 9. Requisito no funcional 1.....	25
Tabla 10. Requisito no funcional 2.....	25
Tabla 11. Requisito no funcional 3.....	25
Tabla 12. Requisito no funcional 4.....	25

Lista de Figuras

	Pág.
Figura 1. Arquitectura Android	18
Figura 2. Arquitectura de desarrollo.....	28
Figura 3. ID LOGIN	29
Figura 4. ID CATREP	31
Figura 5. Inicio de sesión.....	32
Figura 6. Menú	33
Figura 7. Frecuencia cardiaca.....	33
Figura 8. Saturación.....	34
Figura 9. Conexión Pulsera 1	35
Figura 10. Conexión Pulsera 2	35
Figura 11. Conexión Pulsera 3	36
Figura 12. Perfil.....	37

Lista de Anexos

	Pág.
Anexo A. Anteproyecto.....	42
Anexo B. Artículo de revisión estado del arte.....	69

Resumen

Menjura Rodríguez, Lina Fernanda

Asistencia, seguimiento y desarrollo de tratamientos respiratorios de enfermedades pulmonares a través de la app CATREP / Lina Fernanda Menjura Rodríguez, Juan Carlos Gómez Méndez. - - Tunja : Universidad de Boyacá, Facultad de Ciencias e Ingeniería, 2021.

84 h. : il. + 1 CD ROM. - - (Trabajos de Grado de Semillero de Investigación UB, Ingeniería de Sistemas ; n°)

Trabajo de Semillero de Investigación (Ingeniero(a) de Sistemas). - - Universidad de Boyacá, 2021.

El prototipo de la aplicación de asistencia, seguimiento y desarrollo de la app CATREP, está diseñado para monitorear los signos vitales de pacientes con cuadros respiratorios, información primaria para alimentar la historia clínica del paciente de forma que contribuya a la toma de decisiones por parte del personal de salud. El prototipo se desarrolló en conjunto con las APIs de HUAWEI (Health-Kit y Huawei Id), las cuales permiten la lectura de datos y la identificación de la aplicación.

Para la elaboración de este prototipo fue necesario adquirir conocimientos e investigar las funcionalidades y restricciones de herramientas de software como Android Studio, Health-Kit y Huawei Id y los lenguajes de desarrollo Kotlin y Java.

La investigación es de tipo aplicada y descriptiva, donde se aplican los conocimientos adquiridos durante la investigación al desarrollo de una herramienta tecnológica. Para el desarrollo de la aplicación se usó la metodología de prototipos.

La investigación permitió interactuar con APIs que manejan tecnologías ofrecidas por el mercado, ampliar conocimientos y establecer relaciones para conseguir permisos estudiantiles para el desarrollo de aplicaciones, como la que se le ofrece al profesional de la salud para monitorear a sus pacientes, tomar decisiones y dar recomendaciones ágiles y apropiadas.

Introducción

La mayoría de las enfermedades respiratorias se presentan debido a la acumulación de secreciones o flemas en los pulmones que, al no ser expulsadas, generan obstrucción en las vías respiratorias; los tratamientos utilizados para controlar y expulsar dichas secreciones consisten en la utilización de medicamentos e intervención terapéutica, que permiten al paciente mejorar su estado de salud en un intervalo de tiempo prolongado.

Actualmente se cuenta con un prototipo elaborado por egresados del programa de Ingeniería Mecatrónica de la Universidad de Boyacá, el cual es un mecanismo tipo camilla, que le permite a los terapeutas tener la movilidad del paciente a determinadas posiciones y grados de lateralidad, gracias a los anterior se decidió implementar una aplicación complementaria que permite la digitalización y toma de signos vitales del paciente para junto al mecanismo aplicar un diagnóstico y tratamiento asertivo según indicaciones del terapeuta respiratorio.

A lo largo del documento se encontrará el desarrollo del proyecto, es decir, desde las características para realizar un diseño hasta tener el prototipo final de la aplicación, además se detallarán las etapas más importantes o sobresalientes, comprendidas entre el diseño, la implementación y el control. Estas estarán divididas en los capítulos del documento, los cuales fueron debidamente descritos para su desarrollo, sumado a lo anterior se mostrará los resultados de cada uno junto con los anexos.

Este documento comprende la parte metodológica de todo el proyecto, se describe el transcurso de cada etapa, el cómo se investigó para llegar a una idea clara y concisa de lo que se debía realizar y cómo se tendrían buenos resultados. Se asumieron diferentes técnicas y procedimientos para realizar cada etapa, obtener fallas y realizar mejoras.

Medición de Signos Vitales

Luego de realizar una revisión de la tecnología que el mercado ofrece para la toma de los signos vitales básicos que se requieren monitorear en los pacientes con enfermedades respiratorias, se decide aprovechar los instrumentos que destacadas empresas tecnológicas han puesto a disposición del mundo para el monitoreo de los signos vitales, es decir las pulseras que, además de lo anteriormente expuesto, son muy fáciles de usar, constituyéndose en un método no invasivo que permite monitorear constantemente al paciente.

Las pulseras que se encuentran en el mercado pueden poseer diversas características y capacidades y el software asociado a ellas, también son diversos; por esto se acotan estas características dentro de los siguientes requerimientos.

Requisitos de la aplicación

Para el desarrollo de la app CATREP se tienen los siguientes requisitos:

- Sistema Operativo Windows.
- Android Studio.
- Lenguajes de programación KOTLIN.
- Lenguaje de programación JAVA.
- Pulsera con capacidades de lectura de ritmo cardiaco y porcentaje de oxígeno en la sangre (Huawei Smart band 4 Pro).
- Smartphone basado en el sistema operativo Android 8 en adelante.

Razonamiento detrás de los requisitos

Windows

Entre los sistemas operativos contemplados para el desarrollo de la aplicación CATREP se tienen en cuenta Windows y MacOS.

Por motivos de presupuesto se descarta inmediatamente MacOS ya que requiere de una gran inversión inicial para su desarrollo la cual no es justificable teniendo en cuenta que la

capacidad de adquisición de un equipo con Windows es mucho mayor, adicional a esto como desarrolladores ya contamos con los equipos de cómputo con el sistema operativo Windows instalado.

Android Studio

Las opciones para el entorno de desarrollo son múltiples, entre los cuales se puede optar por desarrollo en ANGULAR, REACT NATIVE, xCODE, FLUTTER, entre otras.

No escogemos los frameworks de desarrollo web por limitaciones en sus capacidades para la lógica detrás de la adquisición de datos a través de las pulseras de terceros, el entorno desarrollo xCode se descarta por ser un entorno exclusivo del sistema operativo MacOS.

KOTLIN

Este es el lenguaje de programación preferido por Android Studio, es de fácil aprendizaje y optimizado para el desarrollo de aplicaciones móviles, está inspirado en JAVA y Python, por lo que su sintaxis es comprensible y es de funcionalidad robusta.

JAVA

Es necesario el manejo de JAVA para el desarrollo de aplicaciones móviles pues este fue el lenguaje nativo desde su lanzamiento, por esta razón varias funcionalidades avanzadas se realizan bajo este lenguaje de programación, adicional a esto, los sensores comprados para la adquisición de datos manejan su documentación en este lenguaje.

No se presentan conflictos entre Kotlin y Java para el desarrollo ya que estos lenguajes pueden ser interpretados simultáneamente por el entorno de desarrollo y, por ende, la aplicación final.

Smartphone Android 8

Este es el requisito que más flexibilidad ofrece, ya que la amplia gama de teléfonos inteligentes en el mercado que usen un sistema operativo mayor a Android 8 nos permite hacer pruebas y desplegar la aplicación en casi cualquier smartphone que haya sido lanzado a partir del año 2017.

Selección pulsera

En la actualidad en el mercado existe un sin número de pulseras inteligentes que cuentan con la capacidad de entregarle al usuario a través de una interfaz el ritmo cardiaco, ya sea en tiempo real o un promedio calculado en un tiempo determinado. Sin embargo, es necesario que la pulsera tenga la capacidad de leer el porcentaje de oxigenación en la sangre.

Además de estos requerimientos básicos, se indagó la disponibilidad de documentación que facilitara el manejo y comprensión de estas pulseras para el desarrollo del prototipo de aplicación que se propuso. Como se puede ver en la tabla 1, todas las pulseras cumplen los requisitos mencionados y cuentan con documentación, pero, en algunos casos, con restricciones causadas por el idioma.

Tabla 1.

Selección Pulsera

Pulsera	¿Ritmo Cardiaco?	¿Oxímetro?	¿Documentación?	¿Precio?
Huawei Smart band 4 Pro	Sí	Sí	Sí, inglés	COP\$154.900
Xiaomi Mi Band 5	Sí	Sí	Sí, mandarín	COP\$119.900
Fitbit Charge 3	Sí	Sí	Sí, inglés	COP\$899.900
Honor Band 5	Sí	Sí	Sí, mandarín	COP\$259.900

Fuente: Los autores.

Se verificó que estas, no solo midieran el ritmo cardiaco, sino también el porcentaje de oxigenación en la sangre, información que permite establecer situaciones de alarma en un paciente respiratorio.

Las pulseras Xiaomi y Honor se descartaron porque su documentación se encuentra completamente en mandarín. Se le dio gran importancia a la pulsera de la marca Fitbit ya que su documentación era muy amplia, detallada y bien redactada lo cual facilita significativamente el desarrollo, pero su precio está muy por encima del presupuesto, por tal razón se decidió por la pulsera Huawei y su Smart band 4 Pro, si bien la documentación no es tan detallada como se esperaba fue la mejor opción calidad/precio.

Android

El lenguaje sobre el cual se desarrollan aplicaciones nativas para la plataforma Android es Java. Desde el año 2013 fue declarado por Google como el entorno de desarrollo oficial para el desarrollo de aplicaciones nativas en Android, buscando ejercer un mejor control sobre las aplicaciones que los desarrolladores producían. Sumados a los problemas de eficiencia que el IDE Eclipse producía debido a su funcionamiento basado en plugins. Para el año 2014, Android Studio se consolida como IDE, al eliminarse el Android Developer Tool (ADT). Dentro de sus características encontramos:

- Nueva interfaz para el desarrollo Android
- Vista previa de la interfaz creada, proporcionando opciones de visualización de acuerdo al dispositivo y versión de Android escogida
- Soporte para aplicación que no corresponden a un Smartphone
- Posibilidad de importar proyectos construidos en Eclipse
- Control de versiones a través de herramientas complementarias (como git, github, mercurial, etc)
- Detección en tiempo real de errores de sintaxis, rendimiento y compatibilidad.

Para desarrollar aplicaciones en Android se utilizan una serie de herramientas incluidas en Android Studio. Así mismo, como desarrolladores, podemos acceder a la mayoría de las herramientas de SDK desde la línea de comandos. Sin embargo, el desarrollo con Android

Studio es el método preferido porque puede invocar directamente las herramientas que necesitamos, mientras que este codifica nuestras aplicaciones. La Figura 1 muestra la arquitectura de Android, la cual permite contextualizar lo que aquí se expone.

Los pasos básicos para el desarrollo de aplicaciones (con o sin Android Studio) abarcan cuatro fases de desarrollo, estas incluyen:

Configuración del entorno

Durante esta fase se instala y configura el entorno de desarrollo. También se crean dispositivos virtuales Android (AVDS) y se conectan dispositivos físicos en los que se pueden instalar las aplicaciones.

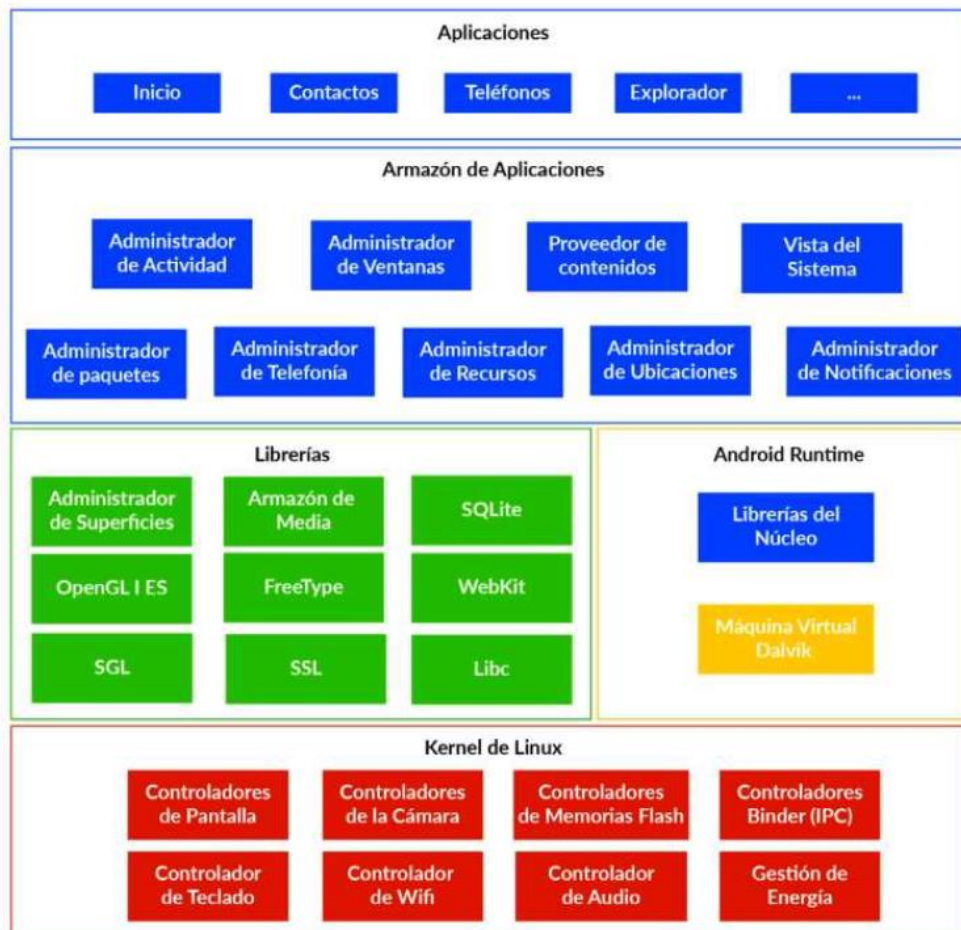


Figura 1. Arquitectura Android, tomada de (Blanco, 2009)

Configuración y desarrollo de proyectos

Durante esta fase se configura y desarrolla el proyecto con sus respectivos módulos, los cuales contienen todos los archivos del código fuente y de recursos para la aplicación. En esta fase se aplican los principios de ingeniería de software.

Construcción, depuración y pruebas

Durante esta fase se construye su proyecto en un paquete apk depurable que se puede instalar y ejecutar en el emulador o en un dispositivo con Android. Android Studio utiliza un sistema de construcción basado en Gradle que proporciona flexibilidad, variantes de construcción a medida, la resolución de las dependencias, y mucho más. Por último, se prueba la aplicación utilizando diversas herramientas de prueba SDK de Android.

Publicación

Durante esta fase se configura y construye liberación y distribución de aplicaciones hacia los usuarios.

Instalación de Android Studio

SDK de Java

Para el funcionamiento de Android Studio es necesario contar con el SDK de Java el cual puede instalar con los siguientes pasos:

1. Diríjase a la siguiente URL: <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>
2. Seleccione la versión del JDK disponible de acuerdo a su sistema operativo, en la siguiente imagen, corresponde a Windows de 64 Bits.
3. Acepte las condiciones del sitio web4.4.

4. Para realizar la descarga, debe iniciar sesión con la cuenta Oracle. Puede crear una cuenta de forma gratuita.

5. Si lo desea, puede revisar los términos y condiciones, a continuación, debe hacer click en Next.

6. Después de la descarga, se iniciará el asistente de instalación, especifique la ruta de instalación del JDK o permita que se utilice la versión por defecto.

7. Al final, recibirá la confirmación del proceso.

Android Studio

La instalación propiamente de Android Studio se detalla a continuación:

1. Diríjase a la URL: <https://developer.android.com/> y seleccione la opción para descargar Android Studio. A continuación, seleccione la opción de descarga.

2. Revise y acepte las condiciones para descargar Android Studio.

3. Si lo requiere, verifique los requisitos mínimos del sistema para un correcto funcionamiento.

4. Después de la descarga, contará con la opción de iniciar el asistente de instalación.

5. Deje seleccionados los componentes por defecto.

6. Utilice la ruta de instalación por defecto.

7. Utilice la opción por defecto para el menú de inicio.

8. Al finalizar la instalación, haga clic en la opción Next.

9. Al finalizar la instalación recibirá el mensaje de confirmación.

10. Una vez instalado, podrá iniciar la aplicación, no es necesario importar configuración, se recomienda no enviar información de uso y luego recibirá un mensaje de bienvenida.

11. Escoja la configuración estándar.

12. Seleccione el estilo de visualización del software de acuerdo a su preferencia.

13. Confirme la configuración para dar inicio a la instalación. Cada paquete se descargará de Internet y será instalado junto al IDE.

14. Al finalizar recibirá el mensaje de confirmación.

15. La interfaz de inicio, le permitirá acceder en la parte inferior a través de la opción: More actions, al SDK Manager, desde donde podrá instalar componentes adicionales dependiendo la versión de Android. Debe ingresar al SDK Manager.

Diseño del Prototipo de la Aplicación

En este capítulo se presentan las tablas y diagramas que orientan el desarrollo de la aplicación propuesta.

Objetivos del sistema

Los objetivos que se tiene previstos para la aplicación se describen a continuación.

Gestión del usuario

La aplicación deberá gestionar los datos del usuario: registro, inicio de sesión, modificar datos, añadir HR, etc. Estabilidad: Media/Alta.

Gestión de alertas

La aplicación deberá ser capaz de gestionar las alertas: editando parámetros, actualizando en BD, estudiar datos de HR, etc. Estabilidad: Alta.

Gestión de Signos Vitales

La aplicación deberá gestionar los datos de la pulsera inteligente recibidos a través de la pulsera: añadir datos, mostrar datos, etc. Estabilidad: Baja/Media.

Requisitos funcionales del sistema

Los Requisitos Funcionales (RF) reúnen los servicios que ofrece el sistema, la reacción que debe tener en situaciones particulares y, en ciertos casos, especificará qué no debe hacer.

La información que se debe aportar de cada requisito funcional se presenta en las tablas de la 2 a la 8.

Tabla 2.

Requisito funcional 1

Identificador	RF -1
Nombre	Registrar usuario
Descripción	La aplicación debe ser capaz de registrar al usuario en la BD, previa conexión a Internet.

Fuente: Los autores.

Tabla 3.

Requisito funcional 2

Identificador	RF -2
Nombre	Iniciar Sesión
Descripción	La aplicación debe ser capaz de iniciar sesión de cualquier usuario en la BD, previa conexión a Internet

Fuente: Los autores.

Tabla 4.

Requisito funcional 3

Identificador	RF -3
Nombre	Modificar datos del usuario
Descripción	La aplicación debe ser capaz de modificar los datos del perfil del usuario. Si no tiene conexión a internet, se almacenarán en local y cuando se conecte actualizará en la BD.

Fuente: Los autores.

Tabla 5.

Requisito funcional 4

Identificador	RF -4
Nombre	Cerrar Sesión
Descripción	La aplicación debe ser capaz de cerrar sesión en el dispositivo y olvidar sus datos de acceso.

Fuente: Los autores.

Tabla 6.

Requisito funcional 5

Identificador	RF -5
Nombre	Añadir HR
Descripción	La aplicación debe ser capaz de almacenar distintos signos vitales en cada lectura de la pulsera.

Fuente: Los autores.

Tabla 7.

Requisito funcional 6

Identificador	RF -6
Nombre	Calcular media diaria
Descripción	La aplicación debe ser capaz de calcular la media de HR del día cuando éste finalice y almacenarla en la BD.

Fuente: Los autores.

Tabla 8.

Requisito funcional 7

Identificador	RF -7
Nombre	Alarma
Descripción	La aplicación debe ser capaz de mandar una alerta al móvil cuando se detecten signos de alerta en el paciente.

Fuente: Los autores.

Requisitos no funcionales del sistema

Los Requisitos No Funcionales (RNF) no hacen referencia a funciones del sistema, sino que describen propiedades que éste debe cumplir. De este modo, se restringe de forma indirecta la aplicación.

La información que se debe aportar de cada requisito no funcional se presenta en las tablas de la 9 a la 12.

Tabla 9.

Requisito no funcional 1

Identificador	RF -1
Nombre	Ámbito
Descripción	La aplicación está dirigida a profesionales de la salud que atienden pacientes con enfermedades pulmonares para un mayor control de sus signos vitales

Fuente: Los autores.

Tabla 10.

Requisito funcional 2

Identificador	RF -2
Nombre	Hardware para la ejecución
Descripción	El paciente necesitará tener una pulsera Huawei Smart band 4 Pro y un dispositivo móvil Android.

Fuente: Los autores.

Tabla 11.

Requisito funcional 3

Identificador	RF -3
Nombre	Hardware para el desarrollo
Descripción	Durante el desarrollo será necesario un PC con SO Windows.

Fuente: Los autores.

Tabla 12.

Requisito funcional 4

Identificador	RF -4
Nombre	Software para la ejecución
Descripción	Durante el desarrollo será necesario la correcta instalación y configuración de Android Studio.



Fuente: Los autores.

Desarrollo de la Aplicación

Desarrollo HUAWEI

Acceso al dispositivo

El dispositivo de hardware se conectará a través del protocolo Bluetooth estándar proporcionado por la aplicación CATREP (Health Kit)

- En la pantalla principal de tu teléfono, desliza el dedo hacia abajo desde la barra de estado y abre el panel de accesos directos.
- Presiona  para habilitar o deshabilitar Bluetooth.
- Mantén presionado  para abrir la pantalla de ajustes de Bluetooth.

Emparejar un dispositivo:

1. Habilita Bluetooth en la pantalla de ajustes de Bluetooth.
2. Asegúrate de que el dispositivo que vas a conectar con tu teléfono esté visible en la sección Dispositivos Disponibles.
3. Selecciónalo y sigue las instrucciones en pantalla para finalizar.

Adquisición de datos

Debido al dispositivo seleccionado es de marca HUAWEI, se debe hacer uso del kit de Salud (Health Kit) ofrecido por la empresa, el cual permitirá que la aplicación CATREP accedan a los datos de salud y estado físico de sus usuarios, en función de un ingreso y autorización de HUAWEI.

HUAWEI proporciona un mecanismo para el almacenamiento y el intercambio de datos de salud y fitness basado en una autorización flexible.

Por medio de Health Kit, HUAWEI proporciona una plataforma de datos y capacidades abiertas, para que desarrolladores como nosotros podamos crear aplicaciones, en este caso

CATREP. Esta extensión conecta los dispositivos de hardware y las aplicaciones para brindar a los consumidores atención médica, orientación para el entrenamiento y una mejor experiencia de servicio.

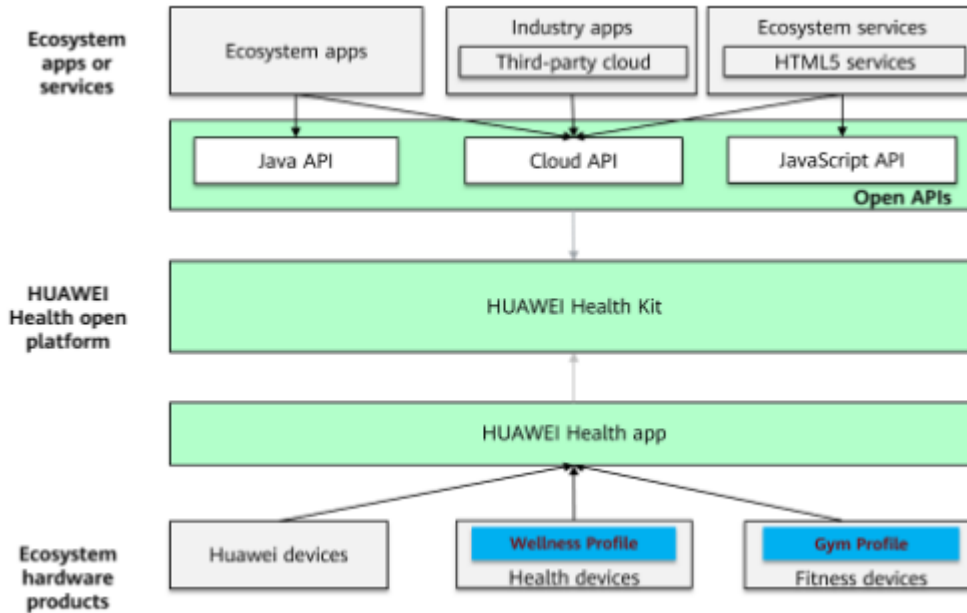


Figura 2. Arquitectura de desarrollo, tomada de (HUAWEI Developers. All rights reserved., 2021).

Como se muestra en la figura 2, Health Kit proporciona una serie de capacidades abiertas para aplicaciones / servicios del ecosistema y dispositivos de hardware.

- API de Java: permite a los desarrolladores de aplicaciones móviles leer y escribir datos de salud y estado físico.
- API en la nube: admite el acceso de los desarrolladores de aplicaciones web y móviles para leer y escribir datos de salud y estado físico.
- API de JavaScript: admite el acceso de desarrolladores de socios para proporcionar servicios de valor agregado para la aplicación Huawei Health.

El kit de Salud permite que las aplicaciones accedan a la base de datos para que puedan agregar, eliminar, modificar y consultar datos del usuario, proporcionando servicios de datos de salud y estado físico para los usuarios, continuando con la solicitud de permisos que HUAWEI le proporcionará a la persona que esté ingresando a la aplicación

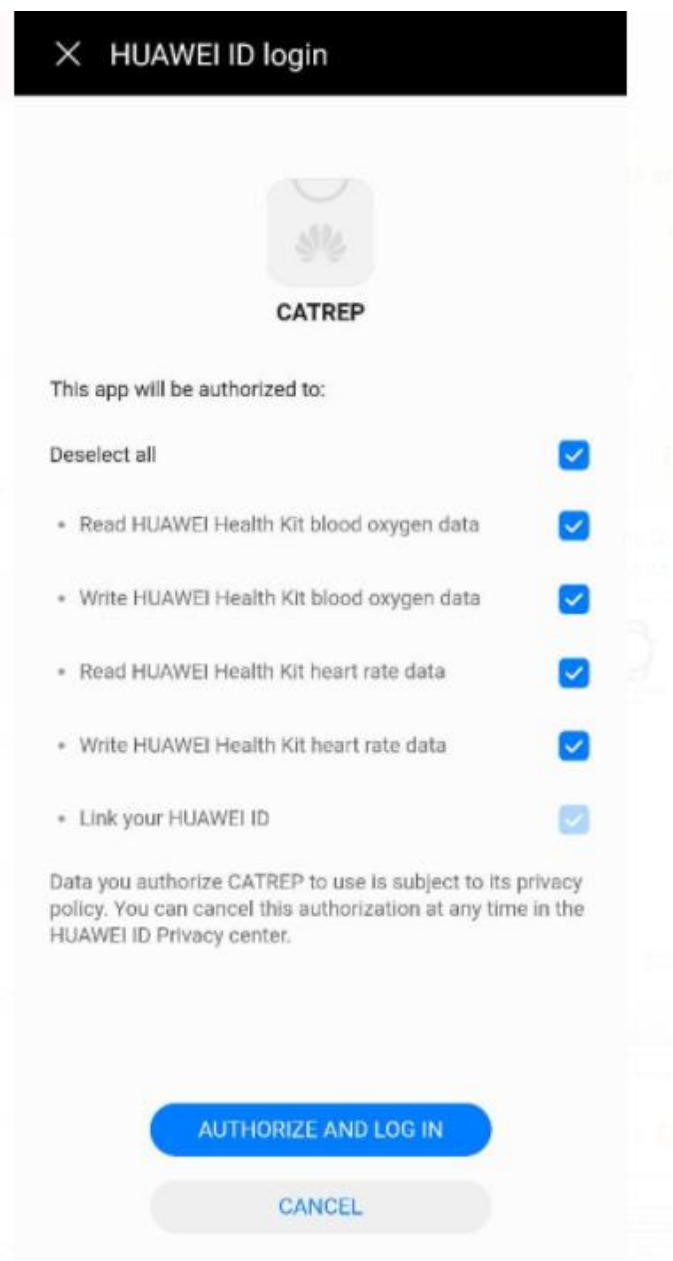


Figura 3. ID LOGIN. Fuente: Los autores.

El usuario puede otorgar los permisos de acceso a los datos correspondientes, mediante una selección de los mismos. Los datos a los que puede acceder la aplicación están dentro del rango de datos aprobados por la aplicación Huawei Health y también dentro del alcance de tratamiento de datos otorgado por el usuario.

Bitácora de desarrollo

Se destinaron dos meses para el aprendizaje del lenguaje de programación Kotlin, el cual permite a los desarrolladores, trabajar de manera más fluida en un lenguaje con el que sientan más cómodos. El curso es provisto de manera gratuita por Google Developers en la url: [Android Basics in Kotlin course](#).

Una vez terminado el tiempo de estudio se inició la creación del prototipo del frontend en la aplicación web Framer, en la cual se detallaron las vistas a utilizar.

Se inició la transición del prototipo frontend de Framer a Android Studio, se crea una lógica sencilla para el inicio de sesión y se finaliza la programación de la interfaz gráfica en Android Studio, se despliega la aplicación en teléfonos inteligentes y se prueba que todas las pantallas están conectadas entre sí basado en el prototipo diseñado.

Se presentan problemas con la implementación de las funcionalidades de la pulsera, pues, debido a que Google le negó a Huawei hacer uso del sistema operativo Android, el soporte es limitado. Es necesaria la instalación de la aplicación Huawei Health para la conexión con la pulsera; dentro de la aplicación, la lectura de datos es correcta, pero no es posible hacer la conexión con la pulsera de manera nativa en la aplicación CATREP, es necesario recolectar los datos a través de la aplicación Huawei Health para poder mostrarlos dentro de la aplicación CATREP.

Se solicitó a HUAWEI un permiso estudiantil para continuar con la investigación haciendo uso de información sensible que se está capturando por la API Health-Kit, para lo que fue necesario demostrar que es un proyecto de investigación de carácter educativo, sin ánimo de lucro. La asignación de la app ID se muestra en la figura 4. Una vez recibido el permiso, se inicia el desarrollo de la aplicación, como tal.

HUAWEI DEVELOPERS

Hi developer,

Thank you for your support for Health Kit!

We are pleased to inform you that your **Health Kit permission application** for your app **CATREP** (app ID: 104759719) has been **approved**. You can now develop and test your app with HUAWEI Health Kit integrated.

Note that the current review is only for development and test purposes, and the number of users is limited. Only the first 100 users can use Health Kit in your app. After your app is tested, you can submit a verification application to remove the limit on the number of users. For details, see the [Applying For Health Kit](#).

In order to ensure a speedy development process, please find the following information for your reference:

1. [Health Kit documentation](#), including the **development guide**, **API reference**, **sample code**, and **SDK download guide**.

Go to **HUAWEI Developers > Develop > HMS Core > Health Kit** to view documents.

2. If you need assistance during development, try the following:

a. Go to **HUAWEI Developers > Support > Intelligent Assistant**.

b. Go to **HUAWEI Developers > Support > Submit a ticket > HMS Core > Health Kit**.

c. Go to **HUAWEI Developers > Community > HUAWEI Developer Forum**.

3. You are welcome to share your experience and thoughts by [submitting a ticket online](#) or on the [HUAWEI Developer Forum](#). For business cooperation, feel free to contact us via hihealth@huawei.com.

Wish you a great experience in app development.

HUAWEI Health Team

Figura 4. ID CATREP. Fuente: Los autores.

El prototipo

A continuación, se muestran las pantallas de mayor importancia del prototipo.

Iniciar sesión

Cuando la aplicación se ejecuta por primera vez, el usuario (profesional médico) ingresa sus credenciales de HUAWEI por primera y única vez. Estas creencias quedan guardadas en la nube de HUAWEI hasta que se revoque sus permisos, de manera manual por el usuario.

La lógica de dicha actividad se encuentra en LoginActivity.java, mientras que el diseño está implementado en activity_login.xml.

La figura 5 muestra la vista de inicio de sesión.

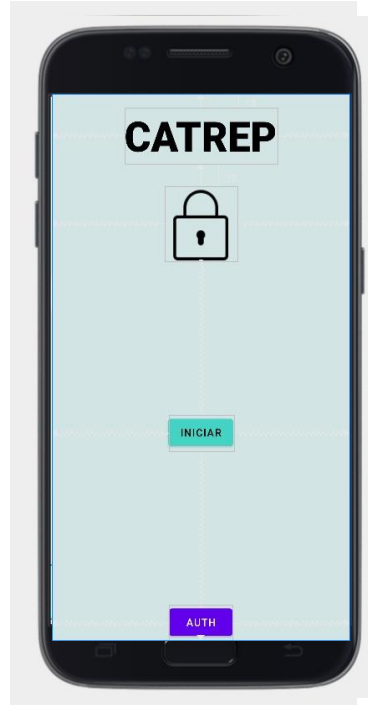


Figura 5. Inicio de sesión. Fuente: Los autores.

Home

Una vez que el usuario inicia sesión, independientemente de la pantalla, aparece en la actividad de la figura 6, cuya lógica se encuentra implementada en HomeActivity.java mientras que su diseño está en activity_home.xml.

En este menú el usuario deberá seleccionar el signo vital del que desea hacer la medición en el paciente.



Figura 6. Menú. Fuente: Los autores.

Frecuencia cardiaca

Muestra el dato recogido en pulsaciones por minuto (bpm) del paciente. Las actualizaciones de la pantalla se hacen cada segundo y se visualizan en tiempo real, como se muestra en la Figura 7.



Figura 7. Frecuencia cardiaca. Fuente: Los autores.

En el botón de información, el usuario podrá comparar los resultados obtenidos en la medición con una tabla de referencia con información de la frecuencia cardiaca; se resalta que estos valores son teóricos y no se asocian a las patologías de los pacientes.

Oxígeno en la sangre

Se muestra el valor de la saturación de oxígeno del paciente en porcentaje (%), como se ve en la figura 8.



Figura 8. Saturación. Fuente: Los autores.

En el botón de información, el usuario podrá comparar los resultados obtenidos en la medición con una tabla de referencia con información de la saturación de oxígeno en la sangre; se resalta que estos valores son teóricos y no se asocian a las patologías de los pacientes.

Conexión de la pulsera

La conexión con la pulsera se realiza a través de la aplicación Huawei Health, como se ve en las figuras 9, 10 y 11.

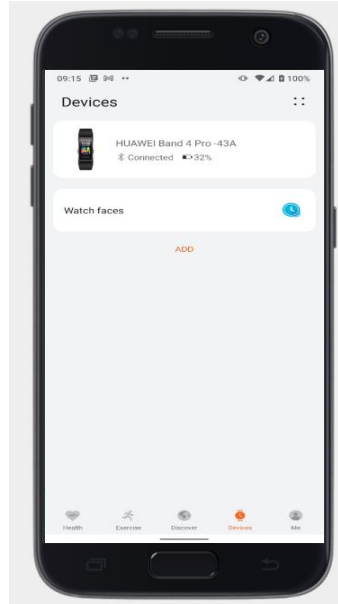


Figura 9. Conexión Pulsera 1. Fuente: Los autores.

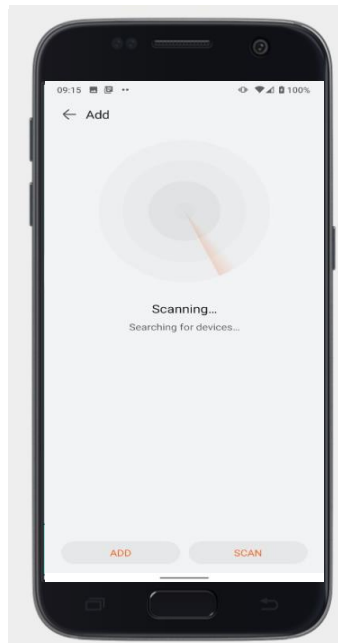


Figura 10. Conexión Pulsera 2. Fuente: Los autores.

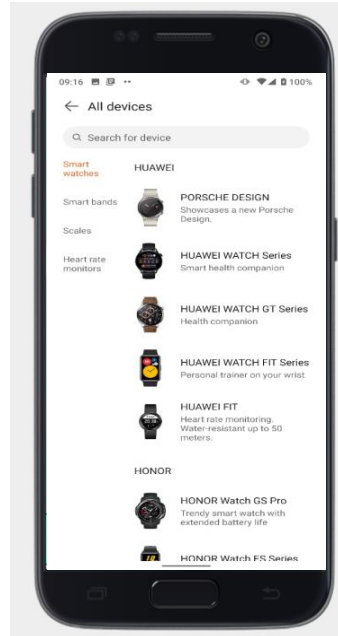


Figura 11. Conexión Pulsera 3. Fuente: Los autores.

Perfil

Esta pantalla se diseñó para modificarán los datos personales del usuario. La actividad está implementada en ProfileActivity.java mientras que el apartado gráfico se encuentra en activity_profile.xml.

De acuerdo a la autorización recibida de HUAWEI para la aplicación, por cada celular solo se tiene permiso para un usuario, quien ingresa sus datos una única vez; por lo tanto, esta pantalla no es funcional en la aplicación final.



Figura 12. Perfil, Fuente: Los autores.

Cerrar sesión

En todas las pantallas a las que podemos acceder con la sesión iniciada, se puede realizar un cierre de sesión. Cuando esto ocurre se desconecta el dispositivo (en caso de estar previamente conectado), se eliminan las preferencias y se le redirige hacia la pantalla principal de la figura 5.

Conclusiones

La formación como ingenieros de sistemas permite interactuar con herramientas de software de terceros, disponibles en el mercado, para crear soluciones novedosas a problemas del entorno.

Dado que se decidió utilizar la manilla de HUAWEI, se hizo necesario cumplir requerimientos para obtener permisos y hacer uso de APIs de forma gratuita, con lo cual se obtuvo un ID específico para la aplicación CATREP.

La metodología de prototipos fue adecuada, debido a que se requería ajustar la aplicación constantemente, para adaptarla a las restricciones de marca, a los permisos que se obtuvieron y al tipo de variables que manejaran las APIs de HUAWEI.

Relacionar la medicina con el área de la ingeniería de sistemas facilita la futura implementación de nuevas tecnologías. Del mismo modo, incrementa la confianza del público en general, al utilizar herramientas automatizadas para la asistencia médica.

Recomendaciones

El prototipo que se entrega en esta investigación es susceptible de ser mejorado. Se puede implementar una funcionalidad que permita al personal médico conocer los registros que la aplicación haya tomado para un paciente determinado con sus fechas, análisis de la información y gráficas de datos, basado en requerimientos propuestos por el personal médico.

La aplicación podrá evolucionar a una versión en que actualice automáticamente la historia clínica de un paciente en el centro médico donde se encuentre. Esto implicaría obtener un permiso por parte de HUAWEI para el tratamiento de los datos y el diseño de una base de datos acorde a la tipología de estos datos.

Dado que la empresa HUAWEI está en constante actualización de las versiones con las que se trabaja la aplicación CATREP, es recomendable manejar un control de versiones, puesto que la actualización entre ellas suele ser tediosa.

Referencias

- Blanco, P. C. (2009). Metodología de desarrollo ágil para sistemas móviles. Introducción al desarrollo con Android y el iPhone. *Doctorado en Ingeniería de Sistemas Telemáticos*, 1-30.
- HUAWEI Developers. All rights reserved. (2021). *About HUAWEI Health Kit*. Recuperado de <https://developer.huawei.com/consumer/en/doc/development/HMSCore-Guides/health-introduce-0000001053684429>