

*Universidad Autónoma Metropolitana
Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería*

**Reporte Final del Proyecto de Integración
Licenciatura en Ingeniería en Computación**

Modalidad Proyecto tecnológico

**ReminderMed: Prototipo de aplicación Android para la gestión de
medicamentos**

Alumno

Juan Carlos Valle Olmedo

Matrícula 206204345

Asesor

Beatriz Adriana González Beltrán

Co-Asesor

María Lizbeth Gallardo López

Trimestre lectivo 17-I

31 de marzo de 2017

Yo, González Beltrán Beatriz Adriana, declaro que aprobé el contenido del presente 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.



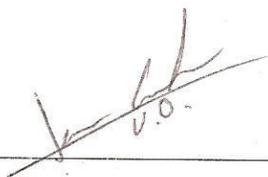
Firma

Yo, Gallardo López María Lizbeth, declaro que aprobé el contenido del presente 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.



Firma

Yo, Valle Olmedo Juan Carlos, 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.



Firma

Resumen

El objetivo de este proyecto de integración es desarrollar una aplicación móvil que permita al usuario gestionar los medicamentos que va a ingerir de manera correcta y en los horarios establecidos, para un mayor aprovechamiento de las recetas médicas. Esta aplicación tiene la característica de recordarnos la ingesta de una pastilla, inyección, jarabe o similar, mediante alarmas que te notifican cuando llegue el momento.

Se puede encontrar software relacionado al proyecto de integración que se presenta en este reporte, como por ejemplo:

- **Pastillero para Pebble.** Esta aplicación hace las funciones de pastillero para monitorear la toma de medicamentos. Ha sido creada con la idea de que sea sencilla de utilizar.
- **Pastillero.** Esta aplicación que actúa como un pastillero, lleva un calendario en el que se puede seleccionar un día y muestra la información de los medicamentos que quedan por tomar.
- **Wedjat.** Esta aplicación para teléfonos inteligentes ayuda a los pacientes a evitar los errores de administración de medicamentos, puede recordar a sus usuarios tomar los medicamentos correctos a tiempo.

La metodología utilizada para el desarrollo del proyecto de integración fue el Proceso Unificado. Este proceso se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental. Se compone de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición, cada una de estas fases es a su vez dividida en una serie de iteraciones mismas que ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.

La aplicación se desarrolló mediante elementos o módulos diseñados por separado e integrados en una aplicación Android. Dichos módulos llevan a cabo una tarea específica y fueron probados por separado antes de integrarlos a la aplicación final.

El diseño los módulos por separado es un excelente método para realizar este tipo de software ya que permite actualizar, cambiar o agregar más funcionalidades a la aplicación sin afectar todo el diseño del software.

Tabla de contenido

Índice de figuras	5
1 Introducción.....	6
2 Antecedentes.....	6
2.1 Referencias internas.....	6
2.2 Referencias externas	7
3 Justificación	7
4 Objetivos.....	8
4.1 Objetivo general.....	8
4.2 Objetivos particulares	8
5 Marco teórico.....	9
6 Desarrollo del proyecto.....	11
6.1 Metodología empleada en el desarrollo del proyecto.	11
6.2 Diseño del sistema	12
6.2.1 Diagrama de casos de uso.....	12
6.2.2 Casos de uso de texto.....	13
6.2.3 Diagrama de clases del sistema	16
6.2.4 Arquitectura del sistema	18
6.2.5 Estructura de la base de datos	20
6.3 Uso del sistema.....	21
6.3.1 Registrar usuario.....	21
6.3.2 Modificar datos del perfil	22
6.3.3 Añadir un medicamento.....	23
6.3.4 Eliminar un medicamento.....	24
6.3.5 Posponer alarma.....	25
6.4 Hardware y software necesario.....	26
6.4.1 Tecnología para el desarrollo de la aplicación	26
6.4.2 Tecnología para la instalación y puesta en marcha de la aplicación	26
7 Resultados.....	26
8 Análisis y discusión de resultados	27
9 Conclusiones.....	27
10 Perspectivas del proyecto	27
Referencias bibliográficas	28

Índice de figuras

Figura 1. Ciclo de vida de una aplicación Android.	10
Figura 2. Diagrama de casos de uso.	12
Figura 3. Diagrama de clases.....	16
Figura 4. Diagrama de clases con atributos.....	17
Figura 5. Base de datos de la aplicación.....	20
Figura 6. Registrar usuario.	21
Figura 7. Pantalla principal.....	21
Figura 8. Opción para configurar perfil.....	22
Figura 9. Modificar datos.	22
Figura 10. Botón de nuevo medicamento.....	23
Figura 11. Configurar medicamento.....	23
Figura 12. Menús de opciones.....	24
Figura 13. Lista de medicamentos.....	24
Figura 14. Diálogo de confirmación.....	25
Figura 15. Ejecución del recordatorio.....	25

1 Introducción

Un paciente debe tener un expediente médico que incluya la lista de medicamentos que está tomando y que ha tomado.

Sin embargo, en muchas ocasiones, un paciente no tiene registrada la lista de medicamentos que se está tomando y este registro ayudaría a dar mayor información a sus médicos y, por lo tanto, a apoyarles a un mejor diagnóstico. Además, un paciente puede olvidarse de tomar la dosis de un medicamento de manera correcta. Es por ello que con la aplicación ReminderMed para la gestión de medicamentos, se busca apoyar al paciente a llevar su tratamiento de manera organizada.

El objetivo es desarrollar un prototipo de aplicación Android para la gestión de medicamentos, que mediante una interfaz intuitiva y fácil de usar se puedan registrar los medicamentos recetados a un paciente y programar las alarmas necesarias para suministrar la dosis correcta.

2 Antecedentes

2.1 Referencias internas

Sistema gestor de nutrición [1]. En este proyecto se diseña e implementa un sistema de información vía web que permite hacer el registro de pacientes, guardar el historial del seguimiento de los padecimientos y dietas propuestas para estos, además de realizar estadísticas. La diferencia principal con el proyecto presentado es la tecnología, el gestor de nutrición es una aplicación web y la propuesta será una aplicación Android. La relación con la propuesta presentada en este documento, es que ambas son del área de la salud.

Empleo de comandos de voz para el registro de citas médicas mediante una aplicación para dispositivos móviles con sistema operativo Android [2]. El proyecto tiene la finalidad de permitir, a un paciente con alguna enfermedad, registrar las citas a las que su médico le indique asistir usando comandos de voz. Esta aplicación no guarda información clínica, o personal, solo cumple con la función de agendar o planificar. Lo que tiene en común con la propuesta presentada es el uso y la programación de las alarmas o notificaciones, para avisar al usuario los eventos agendados.

Prototipo para un sistema de gestión de síntomas de un paciente [3]. En este proyecto se diseña e implementa una aplicación Android que permite al paciente llevar un registro de los síntomas padecidos, puesto que estos forman parte de su historial clínico, el cual es muy valioso para su médico. La similitud de este trabajo con la propuesta a desarrollar es la gestión de la información por parte del paciente.

2.2 Referencias externas

Pastillero para Pebble [4]. Esta aplicación hace las funciones de pastillero para monitorear la toma de medicamentos. Ha sido creada con la idea de que sea sencilla de utilizar. Sólo se introduce la lista de medicamentos que se tomarán, el horario y la dosis. Cuando es el momento de tomar esa medicación se informa mediante una alarma. Pebble no permite avisar a un familiar o amigo, sobre los medicamentos que se está suministrando el paciente. Tanto la propuesta presentada como el pastillero para Pebble son aplicaciones para los dispositivos móviles.

Pastillero [5]. Esta aplicación que actúa como un pastillero, lleva un calendario en el que se puede seleccionar un día y muestra la información de los medicamentos que quedan por tomar. Esta aplicación no permite gestionar los medicamentos que han sido usados anteriormente, ni cuenta con la funcionalidad de las alarmas. La relación con la propuesta presentada en este documento es el uso del calendario en el cual se programa los medicamentos a usar.

Wedjat [6]. Esta aplicación para teléfonos inteligentes ayuda a los pacientes a evitar los errores de administración de medicamentos, puede recordar a sus usuarios tomar los medicamentos correctos a tiempo y mantener un registro para su posterior revisión por profesionales de la salud, también puede alertar a los pacientes acerca de posibles interacciones negativas entre medicamentos o alimentos - medicamentos y planificar un calendario de tomar en que evita estas interacciones adversas, Wedjat trabaja con la aplicación de calendario disponible en la mayoría de los teléfonos inteligentes para emitir los recordatorios. A diferencia con la propuesta presentada dirigida hacia el sistema operativo Android, Wedjat se ha implementado para la plataforma Windows Mobile.

3 Justificación

Existen personas que están en tratamiento médico por un periodo de tiempo y otras que lo están de por vida.

Es muy importante seguir las indicaciones de frecuencia y duración de todo tratamiento médico. Por ejemplo, el uso inadecuado de antibióticos, es la principal razón de que las bacterias que causan una infección, se vuelven inmunes a la acción del medicamento, precisamente por su mal uso. Existen varios factores específicos que desencadenan la resistencia de las bacterias a un antibiótico, como pueden ser: suspender el tratamiento antes de completarlo, incumplir la dosis prescrita, abusar de su uso, auto medicarse y tratar con antibióticos, infecciones que no requieren este tipo de medicamentos. El irrespeto a la

indicación, la dosis y duración de un tratamiento con antibióticos no solo puede generar resistencia al medicamento, sino que podría además agravar la enfermedad, poner en riesgo la recuperación del paciente y comprometer su salud general [7].

El objetivo principal de la aplicación es ayudar a un paciente a registrar los medicamentos prescritos de manera sencilla, sobre todo cuando se requiere tomar de varios medicamentos distintos. Las aplicaciones existentes relacionadas con la toma de medicamentos se enfocan en programar alertas y registrarlas en un calendario.

4 Objetivos

4.1 Objetivo general

Desarrollar un prototipo de aplicación Android para la gestión de medicamentos de un paciente.

4.2 Objetivos particulares

- Diseñar e implementar la interfaz de la aplicación.
- Diseñar e implementar el módulo para registrar el perfil de un usuario.
- Diseñar e implementar el módulo para gestionar medicamentos.
- Diseñar e implementar el módulo de alarmas para recordar el horario de las dosis.
- Integrar los módulos de la aplicación.
- Realizar pruebas de la aplicación.

5 Marco teórico

Entorno de desarrollo Android Studio

En la actualidad Android Studio es una plataforma para desarrollar aplicaciones Android basada en IntelliJ y puede ser descargado de forma gratuita a través de la licencia de Apache 2.0.

Android Studio permite ver el desarrollo de las aplicaciones y las pantallas en las que será usada la aplicación.

Android Studio permite a los desarrolladores una fácil accesibilidad a sus carpetas, archivos y lo que estén usando para crear una aplicación. Las aplicaciones estén escritas en lenguaje Java se pueden compilar y dejarlos como un archivo .apk de una manera muy simple.

Un componente inicial de una aplicación Android es una `Activity`.

Una `activity` es un componente de la aplicación que contiene una pantalla con la que los usuarios pueden interactuar para realizar una acción, como marcar un número telefónico, tomar una foto, enviar un correo electrónico o ver un mapa. A cada `activity` se le asigna una ventana en la que se puede dibujar su interfaz de usuario. La ventana generalmente abarca toda la pantalla, pero en ocasiones puede ser más pequeña que esta y quedar flotando encima de otras ventanas. Una aplicación generalmente consiste en múltiples actividades vinculadas entre sí.

Ciclo de vida de una aplicación

La siguiente figura muestra el camino que debe tomar un `Activity`. Los rectángulos representan los métodos callback se implementan para realizar operaciones cuando la actividad cambia de estado (Figura 1).

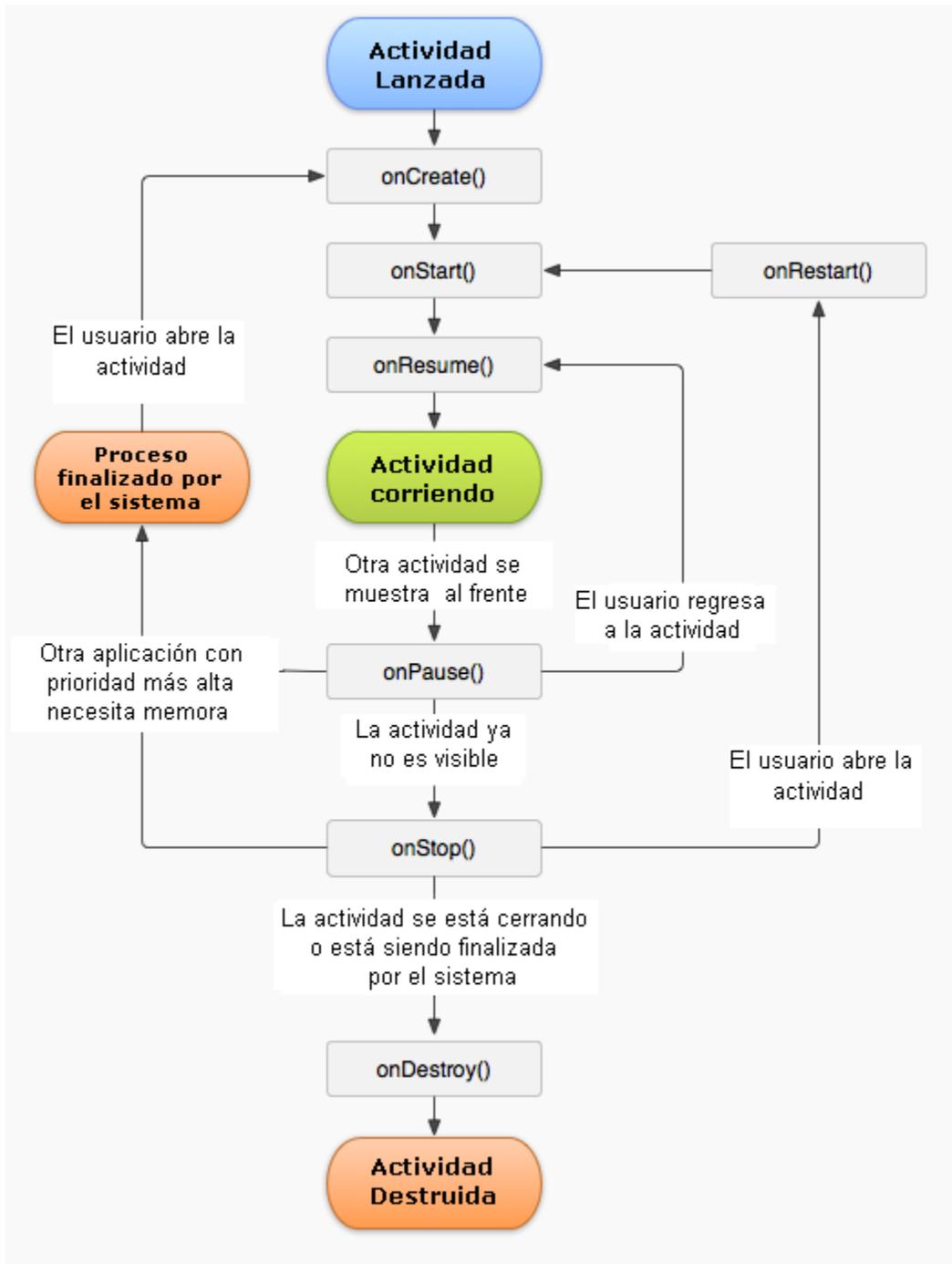


Figura 1. Ciclo de vida de una aplicación Android.

Los métodos callback que muestra la figura e se describen a continuación:

`onCreate()` : Recibe una llamada cuando se crea la actividad por primera vez. Aquí es donde se realizan todas las configuraciones normales como crear vistas, enlazar datos con listas, etc. Seguido por `onStart()`.

`onRestart()` : Recibe una llamada después de que se detiene la actividad, justo antes de que vuelva a iniciarse. Siempre seguido por `onStart()`.

`onStart()` : Recibe una llamada justo antes que la actividad se vuelva visible para el usuario. Seguido por `onResume()` si la actividad pasa a primer plano, o por `onStop()` si se oculta.

`onResume()` : Recibe una llamada justo antes de que la actividad comience a interactuar con el usuario. En este momento la actividad se encuentra en la parte superior de la pila de actividades, y recibe las entradas del usuario. Siempre seguido por `onPause()`.

`onPause()` : Recibe una llamada cuando el sistema está a punto de reanudar otra actividad. Este método generalmente se usa para confirmar los cambios sin guardar como datos persistentes, para detener animaciones y otras tareas que podrían estar consumiendo CPU, etc. La siguiente actividad no se reanuda hasta que el método regrese. Seguido por `onResume()` si la actividad vuelve al primer plano, o por `onStop()` si se vuelve invisible para el usuario.

`onStop()` : Recibe una llamada cuando la actividad ya no es visible para el usuario. Esto puede ocurrir porque se la destruyó o porque se reanudó otra actividad (ya sea una actividad existente o una nueva) y la está cubriendo. Seguido por `onDestroy()` si la actividad desaparece.

`onDestroy()` : Recibe una llamada antes de que se destruya la actividad. Esta es la última llamada que recibirá la actividad.

6 Desarrollo del proyecto

6.1 Metodología empleada en el desarrollo del proyecto.

La metodología utilizada para el desarrollo del proyecto de integración fue el Proceso Unificado. Este proceso se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental. Se compone de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición, cada una de estas fases es a su vez dividida en una serie de iteraciones mismas que ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.

6.2 Diseño del sistema

6.2.1 Diagrama de casos de uso

El siguiente diagrama muestra los casos de uso de forma general (Figura 2).

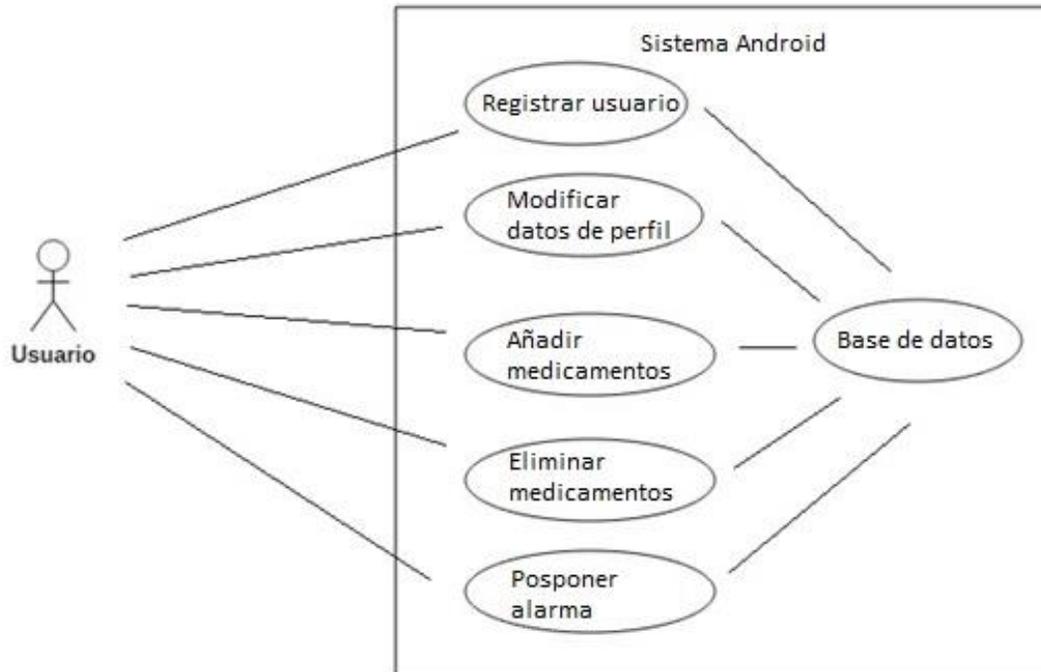


Figura 2. Diagrama de casos de uso.

Los casos de uso de la aplicación Android son los siguientes:

- **Registrar usuario:** El usuario se registra en el sistema para tener acceso a la aplicación.
- **Modificar datos de perfil:** El usuario modifica sus datos de perfil.
- **Añadir medicamento:** El usuario crea un medicamento con la descripción del medicamento, así como la dosis requerida para el paciente.
- **Eliminar medicamento:** El usuario elimina un medicamento.
- **Posponer alarma:** El usuario pospone la alarma hasta el siguiente horario establecido.

6.2.2 Casos de uso de texto

6.2.2.1 Caso de uso: Registrar usuario

Breve descripción

El usuario se registra en el sistema para tener acceso a la aplicación.

Precondición

ninguna.

Postcondición

El sistema crea una cuenta nueva con los datos proporcionados por el usuario.

Flujo de eventos

Flujo base

1 Ejecutar la aplicación

El caso de uso comienza cuando el usuario inicia la aplicación en el dispositivo móvil.

2 El sistema muestra la pantalla de registro de usuario

El sistema muestra el formulario de registro de usuario, el usuario llena todos los campos del formulario y presiona el botón entrar.

3 Enviar datos

El sistema valida los datos ingresados y crea una cuenta para el usuario, quedando registrado y almacenado en la base de datos.

4 Fin del caso de uso

Fin del caso de uso.

Flujo alternativo

3.1 enviar datos

El sistema valida los datos ingresados por el usuario, muestra un mensaje indicando llenar el campo que hace falta.

6.2.2.2 Caso de uso: Modificar Perfil de usuario

Breve descripción

El usuario modifica los datos de su registro.

Precondición

Registrarse en el sistema.

Postcondición

El sistema actualiza los datos proporcionados por el usuario.

Flujo de eventos**Flujo base****1 Ejecutar la aplicación**

El caso de uso comienza cuando el usuario da clic en la opción configurar perfil, del menú desplegable.

2 El sistema muestra la pantalla de configuración

El sistema muestra el formulario de registro de usuario, con los datos proporcionados anteriormente por el mismo.

3 Modificar datos

El usuario modifica los datos del formulario y da clic en guardar.

4 Actualizar datos

El sistema actualiza los datos del perfil.

6.2.2.3 Caso de uso: Añadir medicamento**Breve descripción**

El usuario añade un medicamento.

Precondición

Registrar usuario

Postcondición

El sistema añade un medicamento y crea una alarma nueva con los datos proporcionados por el usuario.

Flujo de eventos**Flujo base****1 Presionar el botón +**

El caso de uso comienza cuando el usuario presiona el botón +.

2 El sistema muestra la pantalla de nueva medicación El sistema muestra el formulario para registrar un nuevo medicamento con los campos descripción de la medicación, dosis, fecha de inicio, fecha final, frecuencia de la dosis y la hora de inicio de la primera medicación.

3 Enviar datos

El sistema valida los datos ingresados y añade un nuevo medicamento.

Flujo alternativo

3.1 enviar datos

El sistema valida los datos ingresados por el usuario, muestra un mensaje indicando llenar todos los campos.

6.2.2.4 Caso de uso: Eliminar medicamento

Breve descripción

El usuario elimina un medicamento.

Precondición

Tener al menos un medicamento registrado.

Postcondición

El sistema elimina un medicamento de la base de datos.

Flujo de eventos

Flujo base

1 Selección de recordatorio

El usuario presiona el botón de borrar de uno de los medicamentos listados en el sistema.

2 Confirmación del sistema

El sistema pide confirmación para eliminar el medicamento seleccionado, el usuario pulsa en confirmar.

3 Borrar medicamento

El sistema borra el medicamento de la base de datos.

Flujo alternativo

2.1 Confirmación del sistema

El usuario pulsa en la opción cancelar, no se realiza ninguna acción y se muestra nuevamente la interfaz de lista de medicamentos.

6.2.2.5 Caso de uso: Posponer la alarma

Breve descripción

El usuario pospone la alarma hasta el siguiente horario establecido.

Precondición

Configurar al menos una alarma.

Postcondición

El sistema pospone la alarma

Flujo de eventos

Flujo base

1 Activación de la alarma

El sistema activa la alarma del recordatorio configurado anteriormente.

2 Pantalla de alarma

El sistema muestra la interfaz de la alarma

3 Posponer recordatorio

El usuario pulsa sobre la imagen del reloj para posponer la alarma hasta el siguiente horario establecido.

6.2.3 Diagrama de clases del sistema

En el siguiente diagrama se muestran las clases del sistema y sus relaciones (Figura 3).

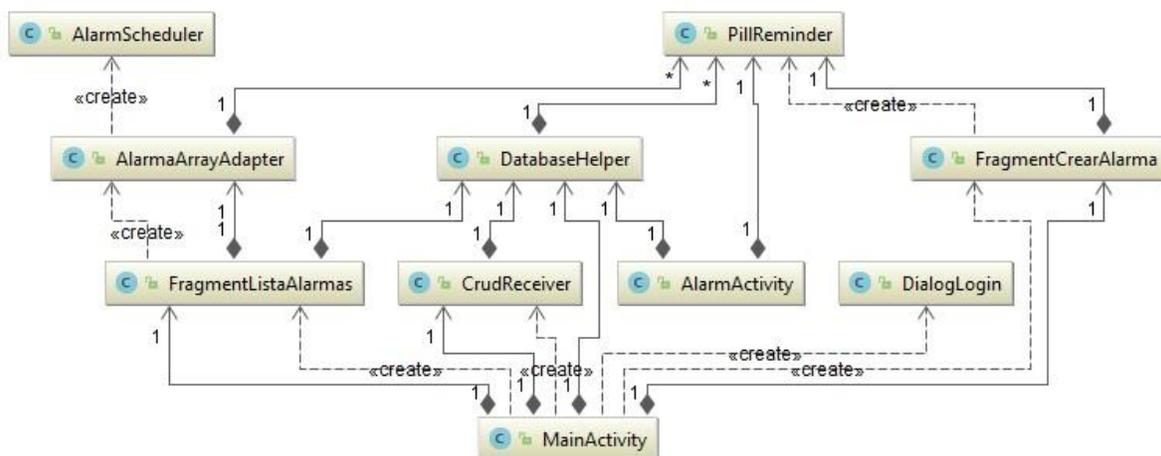


Figura 3. Diagrama de clases

El siguiente diagrama se muestra las clases del sistema sus atributos y relaciones (Figura 4).

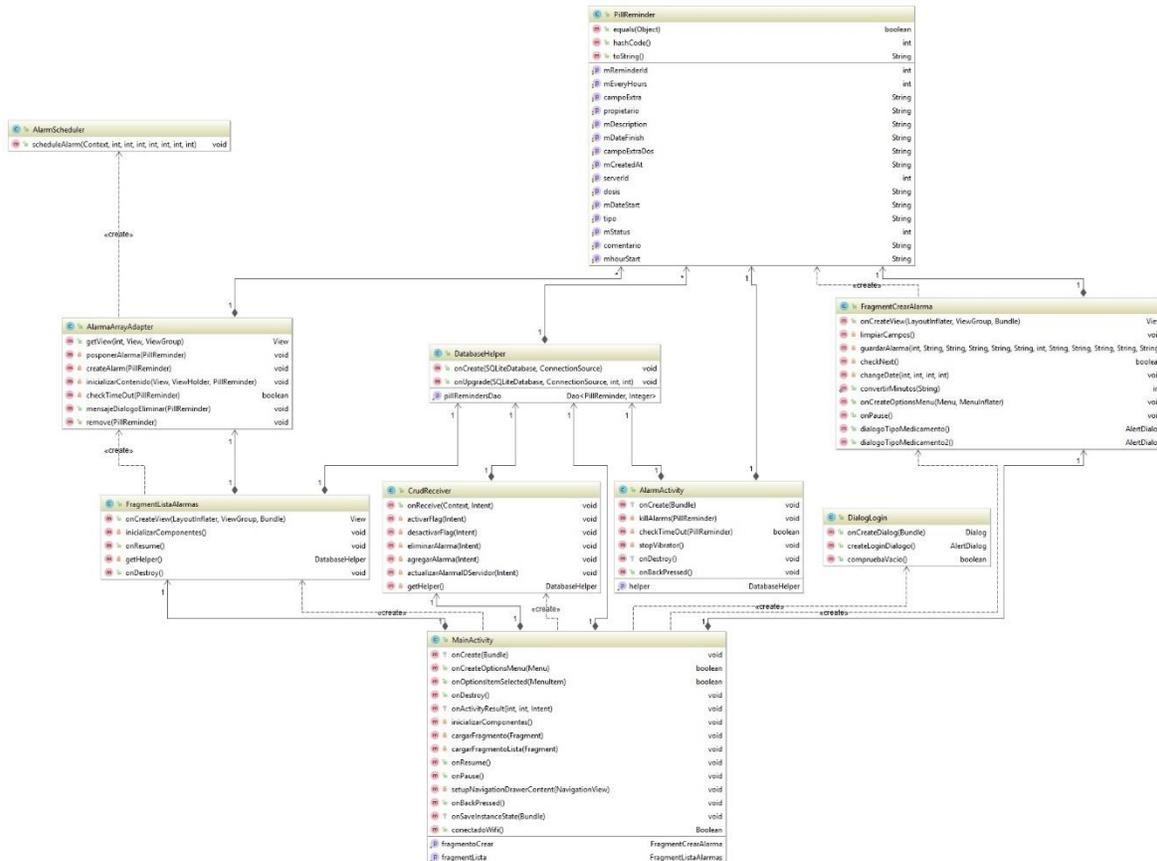


Figura 4. Diagrama de clases con atributos.

Entidades que conforman el diagrama de clases.

- **MainActivity.java**: Contienen la definición de la interfaz gráfica de la pantalla principal de la aplicación.
- **PillReminder.java**: Esta es la clase modelo, es procesable o almacenable dentro de la base de datos
- **DatabaseHelper.java**: Se encarga de todas las llamadas a la base de datos.
- **AlarmActivity.java**: Contiene el funcionamiento de nuestros recordatorios.
- **AlarmScheduler.java**: Contiene los datos de los recordatorios para la próxima alarma.
- **FragmentCrearAlarma.java**: Esta clase permite al usuario crear un nuevo medicamento y su alarma. El código de esta clase está vinculado directamente con el xml vista_alarma.xml.

- `FragmentListaAlarmas.java`: Esta clase permite al usuario ver todos los medicamentos con sus recordatorios.
- `CrudReceiver.java`: Implementa la estructura del `BroadcastReceiver` para la comunicación entre fragmentos de forma modular.
- `AlarmArrayAdapter`: Estructura los datos de los medicamentos para mostrarlos en forma de lista.

6.2.4 Arquitectura del sistema

El sistema está basado en el patrón de arquitectura de software MVC (Modelo – Vista - Controlador), que separa los datos y la lógica de negocio de una aplicación, de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario. Al aplicar esta arquitectura se facilita el desarrollo de la aplicación y su posterior mantenimiento.

En la aplicación se definieron las siguientes clases que forman parte del modelo:

- `PillReminder.java`: Esta es la clase modelo, es procesable o almacenable dentro de la base de datos
- `ConfigDataBase.java`: Clase de ayuda de archivo de configuración de base de datos que se utiliza para escribir un archivo de configuración en el subdirectorio de recursos sin procesar para acelerar la creación de DAO (objeto de acceso a datos).
- `CrudReceiver.java`: Implementa la estructura del `BroadcastReceiver` para la comunicación entre fragmentos de forma modular.
- `DatabaseHelper.java`: Se encarga de todas las llamadas a la base de datos.

En la aplicación se definieron las siguientes vistas:

- `Activity_main.xml`: Interfaz gráfica que muestra al usuario el diseño de la aplicación.
- `alarm_main.xml`: Interfaz gráfica que notifica con una alarma al usuario.
- `dialog_signin.xml`: Interfaz gráfica que permite registrarse en el sistema.

- `configuracion.xml`: Interfaz gráfica que permite modificar los datos del usuario.
- `lista_alarmas.xml`: Interfaz gráfica que muestra la lista de los medicamentos y sus alarmas registrados.
- `navigation_drawer_header.xml`: Interfaz gráfica que muestra el menú de navegación desplegable.
- `registro.xml`: Interfaz gráfica para modificar los datos del usuario.
- `row_alarma.xml`: Interfaz gráfica que muestra los datos de los medicamentos registrados.
- `vista_alarma.xml`: Interfaz gráfica para configurar un medicamento y su alarma.

En la aplicación Android se definieron los siguientes controladores:

- `FragmentCrearAlarma.java`: Esta clase permite al usuario crear un nuevo medicamento y su alarma, el código de esta clase está vinculado directamente con el xml `vista_alarma.xml`.
- `FragmentListaAlarmas.java`: Esta clase permite al usuario ver todos los medicamentos con sus recordatorios.
- `ConfigPerfilFragment.java`: Modifica los datos del perfil del usuario.
- `ConfiguracionActivity.java`: Registro de usuario.
- `AlarmArrayAdapter`: Estructura los datos de los recordatorios para mostrarlos en forma de lista.
- `DialogDate.java`: Muestra un calendario para seleccionar la fecha inicial y final en la configuración de un recordatorio.
- `DialogHour.java`: Muestra un cuadro de selección para indicar la frecuencia de las alarmas.
- `DialogHour_cero.java`: Muestra un cuadro de selección para indicar la hora inicial del primer recordatorio.
- `DialogLogin.java`: Muestra los diálogos para un correcto registro de usuario.

6.2.5 Estructura de la base de datos

El siguiente diagrama muestra el esquema de la base de datos (Figura 5).

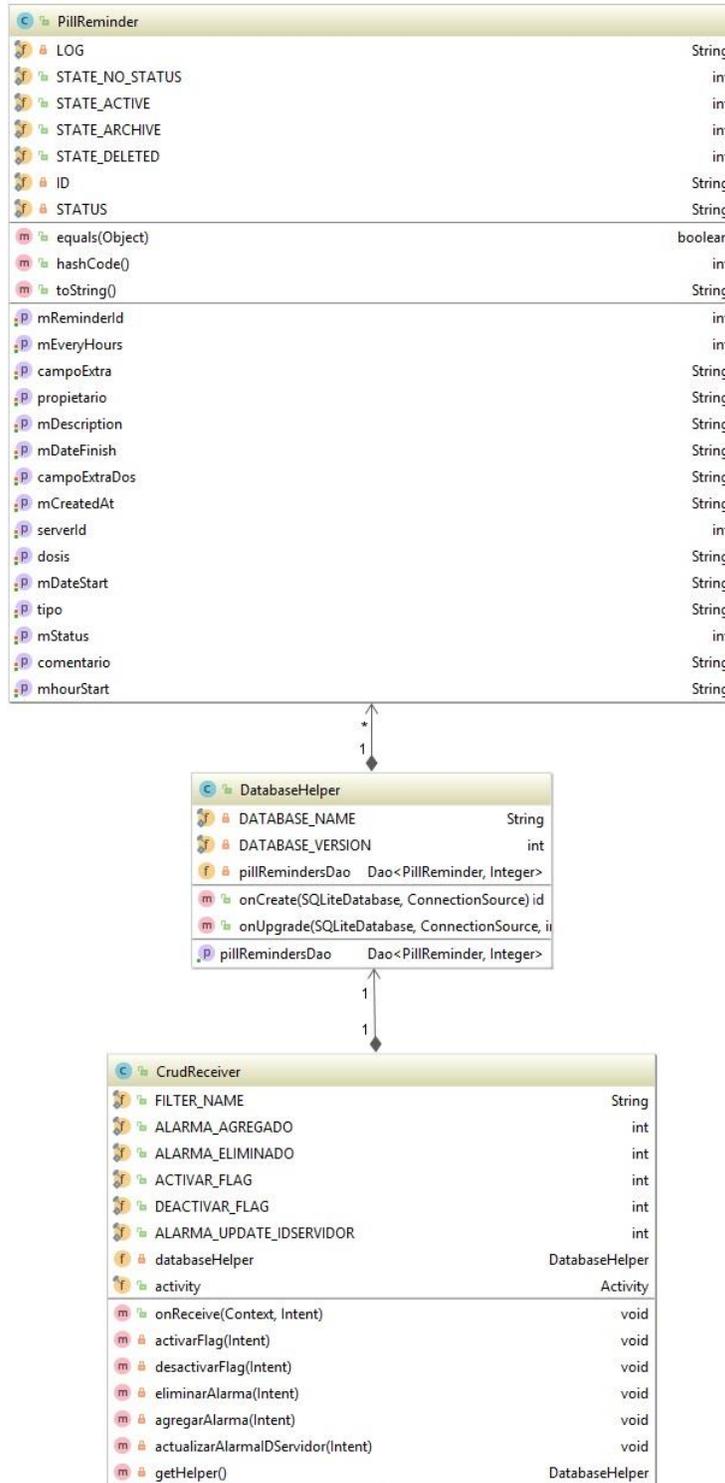


Figura 5. Base de datos de la aplicación.

6.3 Uso del sistema

6.3.1 Registrar usuario

Al ejecutar la aplicación por primera vez se muestra una pantalla inicial de bienvenida (Figura 6). Para registrarse en el sistema, se llenan los campos nombre y correo electrónico, una vez estén completos se pulsa entrar, el sistema valida los datos ingresados y crea una cuenta para el usuario.

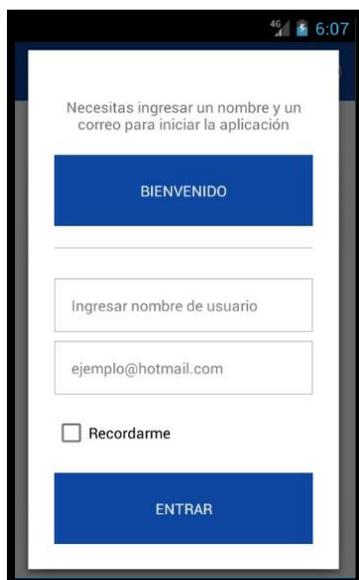


Figura 6. Registrar usuario.

Una vez registrado se muestra la pantalla principal de la aplicación (Figura 7).



Figura 7. Pantalla principal.

6.3.2 Modificar datos del perfil

Los datos del perfil se pueden modificar pulsando la opción Configurar perfil, en el menú de navegación (Figura 8).

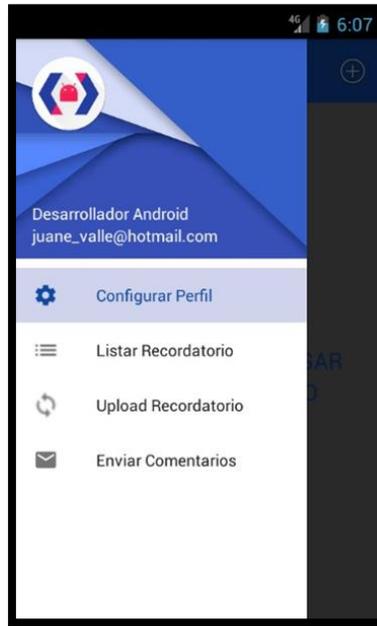


Figura 8. Opción para configurar perfil

Una vez en la pantalla de perfil se pueden modificar los datos del registro de usuario y se pulsa en guardar (Figura 9).

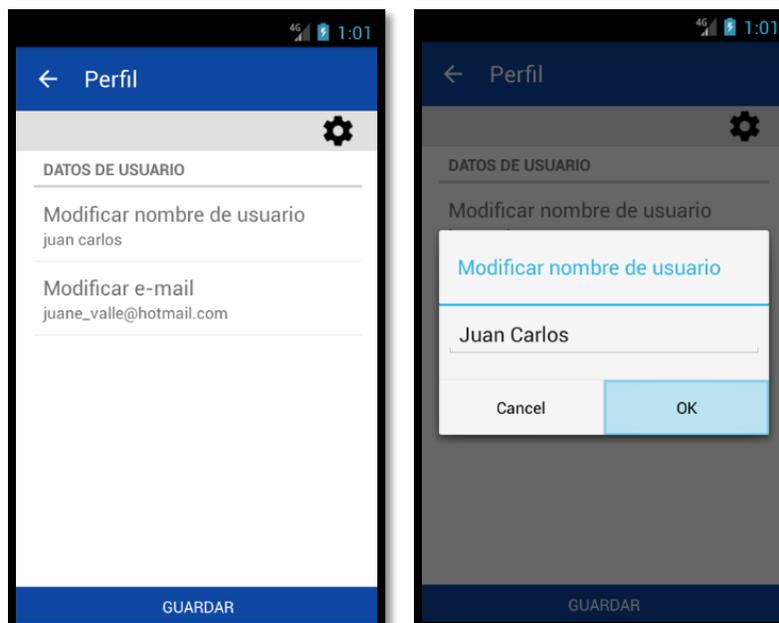


Figura 9. Modificar datos.

6.3.3 Añadir un medicamento

Se puede añadir los medicamentos que se deseen ingerir. Para agregar un nuevo medicamento se pulsa en el botón + que se encuentra en la parte superior de la pantalla (figura 10).



Figura 10. Botón de nuevo medicamento

Para añadir un medicamento se llenan los campos: descripción del medicamento y dosis, lo siguiente es seleccionar la fecha inicial de la medicación, la fecha final, la frecuencia de la dosis. Además, se pueden crear recordatorios indicando la hora inicial en que se activará el primer recordatorio (Figura 11).



Figura 11. Configurar medicamento

Las opciones de configuración de la fecha de inicio, fecha final, frecuencia de la dosis y hora de inicio de los recordatorios se realizan mediante menús de selección (Figura 12).



Figura 12. Menús de opciones

6.3.4 Eliminar un medicamento

Para eliminar un medicamento, se pulsa en la imagen de la papelera que se muestra al lado del recordatorio que se desea eliminar (Figura 13).



Figura 13. Lista de medicamentos.

Al pulsar en la imagen de la papelera, el sistema pide la confirmación del usuario (Figura 14)

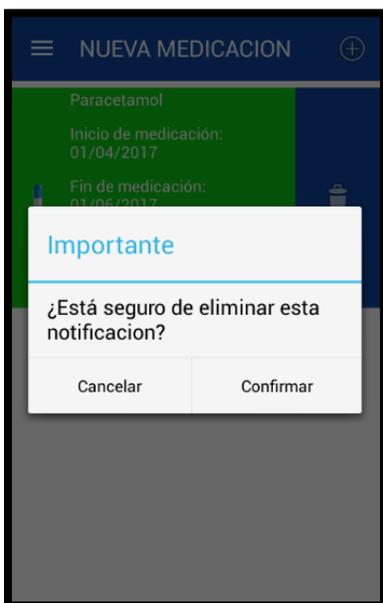


Figura 14. Diálogo de confirmación

6.3.5 Posponer alarma

Una alarma se activa cada cierto tiempo (dependiendo la frecuencia que se indicó) hasta que el usuario decida eliminar el recordatorio.

Para posponer una alarma hasta el siguiente horario establecido, se presiona la imagen del reloj (Figura 15).



Figura 15. Ejecución del recordatorio

6.4 Hardware y software necesario

6.4.1 Tecnología para el desarrollo de la aplicación

El hardware donde se instaló el entorno de desarrollo necesario para programar el proyecto es una computadora con las siguientes características:

- Procesador AMD FX 8350 4.0 GHz
- 8GB de memoria RAM DDR3
- Tarjeta de video 2GB 256-Bit DDR5
- Arquitectura x64

El hardware donde se probó la aplicación Android es un smartphone Samsung Galaxy Trend Plus con las siguientes características:

- Procesador Dual Core 1.2 GHz
- 768MB de memoria RAM
- Resolución 480 x 800(WVGA)
- Sistema operativo Android Jelly Bean (4.2)

6.4.2 Tecnología para la instalación y puesta en marcha de la aplicación

El software necesario para el desarrollo del proyecto es el siguiente:

- Sistema operativo Microsoft Windows 8.1 Pro x64
- JDK (Java Development Kit) [1]
- IDE Android Studio versión 2.3 [2]
- Manejador de base de datos SQLite

El software necesario para probar la aplicación es el siguiente:

- Sistema operativo Android Jelly Bean 4.2.2 o superior.

7 Resultados

En este proyecto se diseñaron e implementaron los siguientes módulos:

- Módulo para registrar usuario.
- Módulo para modificar datos de perfil.
- Módulo para añadir medicamentos.
- Módulo para eliminar medicamentos.
- Módulo para posponer alarma

La aplicación ReminderMed es el resultado de la integración de estos módulos.

8 Análisis y discusión de resultados

Las pruebas realizadas a los módulos mencionados anteriormente, cumplen con sus funciones correctamente.

Para probar el correcto funcionamiento de la aplicación se realizaron por lo menos 15 pruebas a cada uno de los módulos, obteniendo los resultados esperados, por lo tanto, el usuario puede usar la aplicación para registrarse, modificar sus datos, añadir y eliminar medicamentos, y posponer correctamente las alarmas.

9 Conclusiones

Como resultado del desarrollo de la aplicación presentada, es posible concluir lo siguiente:

La ventaja de la aplicación Android ReminderMed es su facilidad de uso, aprendizaje y la rapidez con que se accede a la información por parte del usuario.

El entorno de desarrollo y las herramientas de software que se usaron para desarrollar el proyecto están a disposición de todo usuario de forma gratuita.

La aplicación Android representa una valiosa ayuda para los pacientes que se encuentran en un periodo de tratamiento médico y especialmente para aquellos quienes están en tratamiento de por vida.

10 Perspectivas del proyecto

En un futuro, la aplicación pueda ser mejorada mediante el desarrollo de nuevos módulos que serán fáciles de integrar debido a su diseño modular o la optimización de los ya existentes. También se puede mejorar la interfaz, por ejemplo, queda pendiente para un futuro desarrollador el soporte para dispositivos con pantallas con resoluciones superiores a los 480 x 800 píxeles.

Referencias bibliográficas

- [1] I. Rivera Torres, “Sistema gestor de nutrición”, proyecto terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2012.
- [2] G. Hernández Vázquez, “Empleo de comandos de voz para el registro de citas médicas mediante una aplicación para dispositivos móviles con sistema operativo Android”, proyecto de integración, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2014.
- [3] F. J. García Maldonado, “Prototipo para un sistema de gestión de síntomas de un paciente”, proyecto de integración, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2014.
- [4] S. Sancho, “pastillero para pebble” [online], 7 de mayo de 2014, disponible en: <https://play.google.com/store/apps/details?id=com.sersanchus.pillbox&hl=es> 419
- [5] M. Martínez del Horno, “pastillero” [online], 4 de julio de 2013, disponible en: <https://play.google.com/store/apps/details?id=miguel.tesionv4&hl=es> 419
- [6] Zao, J.K.; Mei-Ying Wang; Peihuan Tsai; Liu, J.W.S., "Smart phone based medicine in-take scheduler, reminder and monitor," e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on, vol., no., pp.162,168, 1-3 July 2010.
- [7] Colegio de Farmacéuticos de Costa Rica. “Colegio de farmacéuticos pide a la población hacer un uso adecuado de los antibióticos”, [online], agosto de 2009 disponible en: http://colfar.com/descargas/boletines/BP_COLFAR_RESISTENCIA_ANTIBIOTICOS.doc

API de la aplicación ReminderMed

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.