UNIVERSIDAD TRES CULTURAS

INGENIERÍA EN SISTEMAS COMPUTACIONALES

APLICACIÓN VR FIREGUARD: SIMULACIÓN PARA USO DE UN EXTINTOR

MANUAL DEL PROGRAMADOR

PRESENTAN:

CASTILLO REYES ALFREDO

GUTIERREZ CHAVEZ FERNANDO MISAEL

MEDINA PAVÓN BRYAN MIGUEL

NAVA DE LA CRUZ MIGUEL ANGEL

ZAMORA MERCADO STEPHANIE

ZARATE GONZALEZ DILAN SERGIO

ASESORES:

M. en C. SELVAS DIAZ BERSAIN

M. en DES. MINOR GÓMEZ JAIME

CUIDAD DE MÉXICO, AGOSTO 2023

MANUAL DEL PROGAMADOR

Introducción

En este anexo se describe la documentación técnica de programación, incluyendo la instalación del entorno de desarrollo, la estructura de la aplicación, su compilación, la configuración de los diferentes servicios de integración utilizados o las baterías de test realizadas.

Estructura de directorios

- `/` : contiene los ficheros de configuración de Gradle, de los servicios de integración continua, el fichero README y la copia de la licencia.
- `/app/` : módulo correspondiente a la aplicación.
- `/app/src/` : código fuente de la aplicación.
- `/app/src/main/` : contiene todas las clases comunes a todos los *flavours*.
- `/app/src/main/res/` : recursos de la aplicación (*layouts*, menús, imágenes, cadenas de texto, etc.).
- `/docs/` : documentación del proyecto.

Manual de programador

El siguiente manual tiene como objetivo servir de referencia a futuros programadores que trabajen en la aplicación. En él se explica cómo montar el entorno de desarrollo, obtener el código fuente del proyecto, compilarlo, ejecutarlo, testearlo y exportarlo.

Entorno de desarrollo

Para trabajar con el proyecto se necesita tener instalados los siguientes programas y dependencias:

- Java JDK 7.
- Android Studio.
- Git.

A continuación, se indica como instalar y configurar correctamente cada uno de ellos.

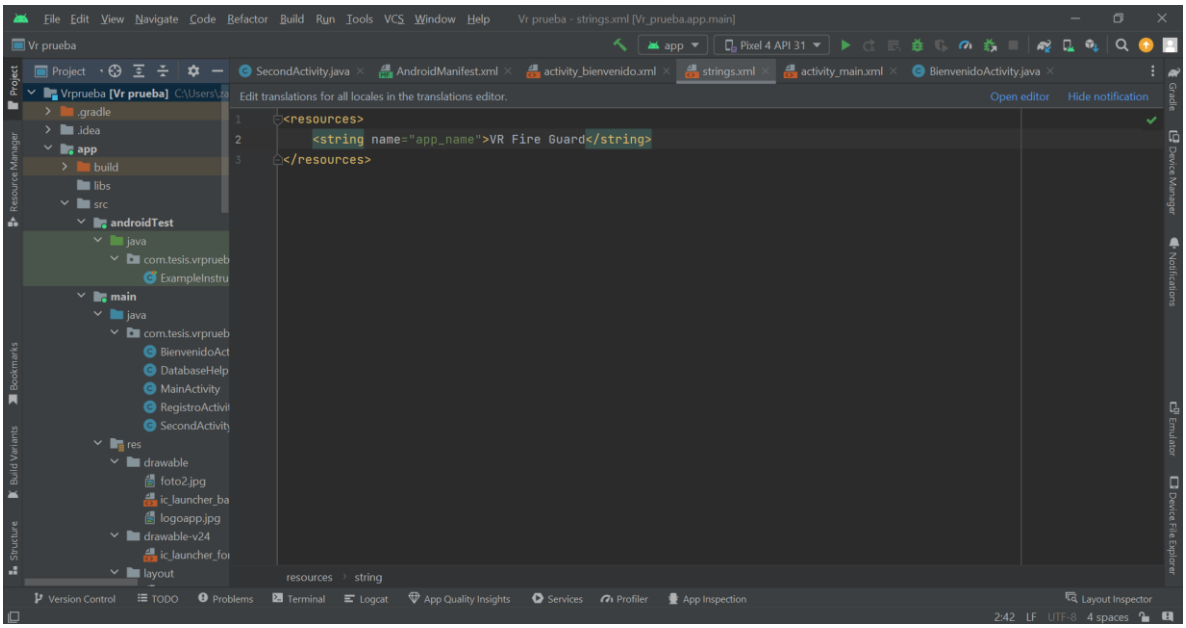
Java JDK 7

El lenguaje de programación más popular para realizar aplicaciones Android es Java. A día

de hoy, Android no soporta la versión 8 de Java, por lo que tenemos que trabajar con la versión 7.
Podemos obtener esta versión desde[http://www.oracle.com/technetwork/es/java/javase/downloads/jdk7-downloads-1880260.html] Se debe elegir correctamente el sistema operativo y la arquitectura del ordenador y posteriormente seguir el asistente de instalación.

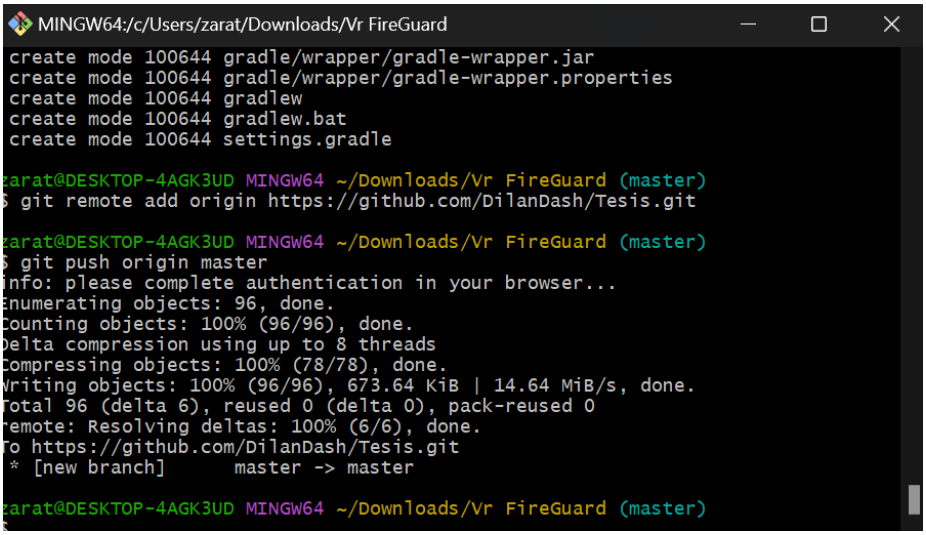
Android Studio

Android Studio es el IDE oficial para el desarrollo de aplicaciones Android. Está basado en IntelliJ IDEA de JetBrains. Proporciona soporte para Gradle, emulador, editor de layouts, refactorizaciones específicas de Android, herramientas Lint para detectar problemas de rendimiento, uso, compatibilidad de versión, etc.
Se puede obtener desde [https://developer.android.com/studio/index.html?hl=es]. Junto con Android Studio se instala también el Android SDK y Android Virtual Device (AVD).



GIT

Para hacer uso del repositorio se necesita tener instalado el gestor de versiones Git. Este programa nos permitirá clonar el repositorio, movernos por sus diferentes ramas, etiquetas, etc. Se puede obtener desde [https://git-scm.com/downloads]. Una vez instalado, trabajaremos con Git Bash.



Obtención del código fuente

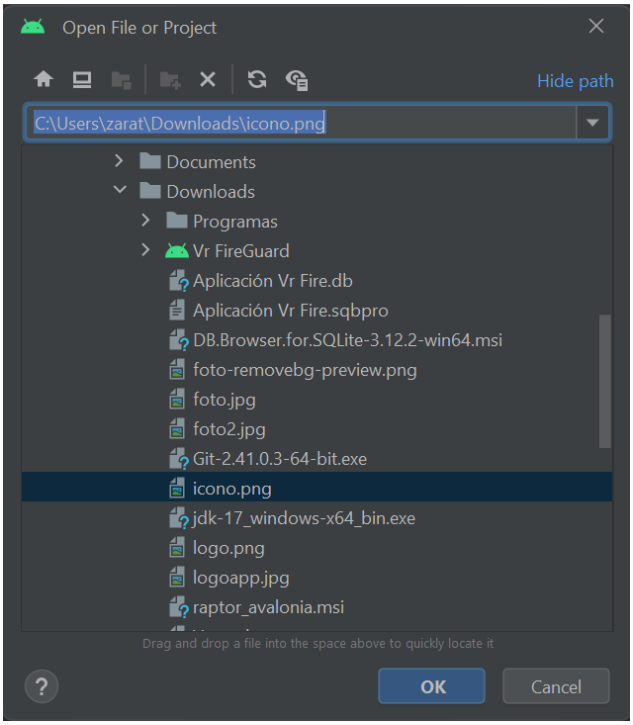
Para el desarrollo de la aplicación se ha utilizado un repositorio Git hospedado en GitHub. Para obtener una copia de este hay que proceder de la siguiente manera:

1. Abrir la terminal Git Bash.
2. Desplazarse al directorio donde se desee copiar el repositorio (utilizando el comando `cd`).
3. Introducir el siguiente comando: `git clone https://github.com/DilanDash/Tesis.git`.
4. Se iniciará la descarga del repositorio, cuando finalice se dispondrá de una copia completa de este.

Importar proyecto en Android Studio

Una vez obtenido el código fuente de la aplicación, tenemos que importarlo como proyecto de Android Studio. Para ello, hay que seguir los siguientes pasos:

1. Abrir Android Studio.
2. Menú `File > Open…`.
3. Buscamos el directorio donde hemos clonado el repositorio.
4. Dentro del repositorio, seleccionamos el archivo `build.gradle`.
5. Android Studio detectará que es un proyecto Android y lo importará automáticamente.
6. Si alguna característica de las que hace uso la aplicación no se encuentra instalada, Android Studio mostrará un mensaje de error con un enlace para instalar la característica en cuestión.



Añadir nuevas características a la aplicación

Tras importar el proyecto en Android Studio, ya estamos en disposición de realizar modificaciones de la aplicación.

Para añadir una nueva característica siguiendo la arquitectura MVP, la convención de paquete por característica y las metodologías TDD y GitFlow, se deben seguir los siguientes pasos generales.

1. Crear una nueva rama (*feature branch*) desde la rama *develop*: `git checkout -b export-data develop`.
2. Crear un nuevo paquete con el nombre de la característica que se desea añadir (ej. `exportdata`).
3. Crear una interfaz (ej. `ExportDataContract.java`) que contenga a su vez dos interfaces. En una se deben definir las responsabilidades del *presenter* y en la otra las de la vista.
   Hacer *commit*: `git add -A` `git commit -m "Add export data contract #x"`.
4. Crear una clase para el *presenter* (ej. `ExportDataPresenter.java`) que implemente su correspondiente interfaz anterior (no añadir ninguna lógica todavía). Hacer *commit*.
5. Crear una clase para la vista (ej. `ExportDataFragment`) que descienda de `Fragment` e implemente su correspondiente interfaz anterior (no añadir ninguna lógica todavía). Hacer *commit*.

6. Crear una clase que descienda de `AppCompatActivity` (ej. `ExportDataActivity.java`) y que enlace el modelo, el *presenter* y la vista. Hacer *commit*.
7. Crear un test sobre el *presenter* de acuerdo a los requisitos. Hacer *commit*.
8. Ejecutar el test y comprobar que no pasa.
9. Implementar las clases anteriores hasta conseguir que pasen el test. Hacer *commit*.
10. Refactorizar el código para mejorar su calidad. Hacer *commit*.
11. Añadir un *intent* desde donde se quiera acceder a esa característica. Hacer *commit*.
12. Una vez que se ha implementado correctamente la característica, se debe incorporar a la rama *develop* y sincronizar con GitHub: `git checkout develop` `git merge --no-ff export-data` `git branch -d myfeature` `git push origin develop`.

Actualizar dependencias

Una tarea de mantenimiento común es la actualización de las dependencias de la aplicación. Es importante tenerlas actualizadas para evitar problemas de seguridad o funcionalidad que pudiesen tener en versiones anteriores.

El proyecto utiliza Gradle como sistemas de construcción automática del software. Una de sus funcionalidades es la gestión de dependencias. Esta permite al desarrollador definir las dependencias de su aplicación, sus versiones y los repositorios donde se hospedan y Gradle se encarga de descargarlas e importarlas al proyecto automáticamente.

Las dependencias se definen en el fichero `build.gradle` del módulo de la aplicación (`go-bees/app/build.gradle`):

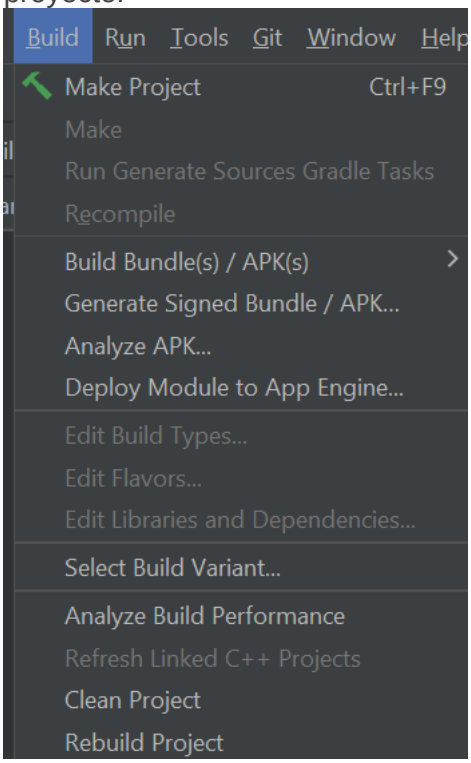
```
dependencies {

    implementation 'androidx.appcompat:appcompat:1.4.1'
    implementation 'com.google.android.material:material:1.5.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.3'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'

}
```

Compilar código fuente

La compilación del proyecto se realiza mediante la tarea `build` de Gradle. Podemos ejecutarla por línea de comandos (`./gradlew build`) o mediante la interfaz de Android Studio.
Todos los ficheros generados durante la compilación se guardan en la carpeta `build` del proyecto.



Ejecutar aplicación

Para ejecutar la aplicación en un dispositivo real, se debe conectar este al equipo de desarrollo mediante un cable USB. El equipo debe tener los *drivers* del dispositivo instalado, sino no lo reconocerá.

Una vez conectado el dispositivo:

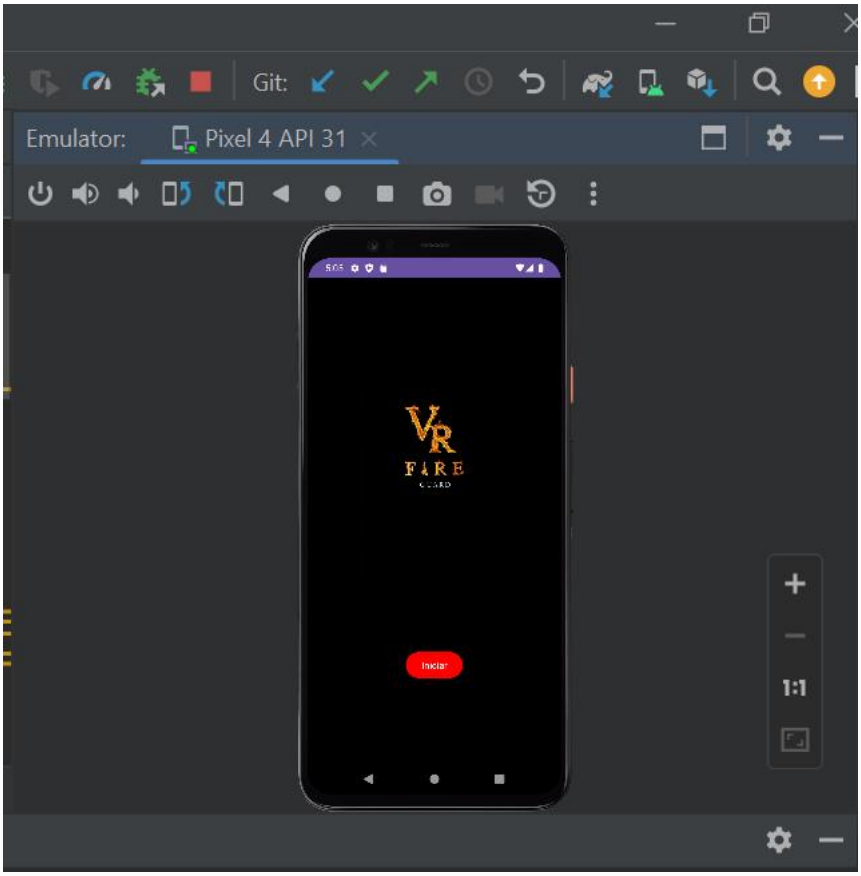
1. Presionar el botón *Run*.
2. Si el equipo reconoce el dispositivo se mostrará su nombre debajo de "*Connected Devices*".
3. Seleccionar el dispositivo y pulsa *Ok*.
4. Se transferirá el ejecutable de la aplicación y se instalará.
5. Una vez instalada, se podrá utilizar la aplicación desde el dispositivo.

Emulador

Un emulador (denominados *Android Virtual Device* - AVD) es una aplicación que simula el funcionamiento de un dispositivo real Android. La creación y gestión de los emuladores se hace a través de *AVD Manager*.

Para ejecutar la aplicación en un emulador:

1. Presionar el botón de Run.
2. Si ya se posee algún emulador instalado, se mostrará en la lista de *Android Virtual Devices*.
3. Si no, presionar el botón "*Create New Virtual Device*".
4. Seleccionar las características que se deseen para el emulador y pulsa finalizar.
5. Seleccionar el emulador creado y pulsar *Ok*.
6. Se iniciará el emulador y se instalará la aplicación en él.
7. Una vez instalada, se podrá utilizar la aplicación desde el emulador.



Exportar aplicación

Para exportar la aplicación como un fichero `.apk`:

1. Menú *Build* > *Generate APK*.
2. Se generará un archivo `apk` y se guardará en `build/output/apk`.

Pruebas del sistema