

## Ejercicios Tema 2

Responde a las siguientes cuestiones. Sube un documento Word, OpenOffice, Google Doc, etc., una vez finalizados TODOS los ejercicios (al acabar del tema).

### Ejercicio 2.1.

Busca y explica en qué consiste el "nivel mínimo de API".

El nivel de API es un valor entero que identifica de manera única la revisión de la API del framework que ofrece una versión de la plataforma de Android.

La plataforma de Android proporciona una API de framework que las aplicaciones pueden usar para interactuar con el sistema de Android subyacente. La API del framework consta de lo siguiente:

- Un conjunto básico de paquetes y clases
- Un conjunto de atributos y elementos XML para declarar un archivo de manifiesto
- Un conjunto de atributos y elementos XML para declarar recursos y acceder a estos
- Un conjunto de intents
- Un conjunto de permisos que las aplicaciones pueden solicitar, al igual que cumplimientos de permisos incluidos en el sistema

Cada versión sucesiva de la plataforma de Android puede incluir actualizaciones a la API del framework de aplicaciones de Android que ofrece.

## Ejercicio 2.2.

Busca en la página de Android Developer qué utilidad puede tener la inclusión del siguiente código en el Manifest:

```
<uses-permission android:name="permiso "  
android:maxSdkVersion = "entero"/>
```

Define un permiso con el nombre "permiso", que será necesario hasta la versión del sdk definida en el valor "entero".

## Ejercicio 2.3.

Acude a la página indicada a continuación e investiga en ella cómo debe hacerse la provisión de recursos y la asignación de los mismos:

<https://developer.android.com/guide/topics/resources/providing-resources?hl=es-419>

Los recursos son los archivos adicionales y el contenido estático que usa tu código, como mapas de bits, definiciones de diseño, strings de interfaz de usuario, instrucciones de animación, etcétera.

Siempre debes externalizar los recursos para apps, como imágenes y strings de tu código, para que puedas mantenerlos de forma independiente. También debes proporcionar recursos alternativos para configuraciones de dispositivos específicos, agrupándolos en directorios de recursos con un nombre especial. En el tiempo de ejecución, Android utiliza el recurso adecuado según la configuración actual. Por ejemplo, puedes proporcionar un diseño de interfaz de

usuario (IU) diferente según el tamaño de la pantalla o strings diferentes según la configuración de idioma.

Una vez que externalizas los recursos para tu app, puedes acceder a ellos mediante los IDs de recursos que se generan en la clase R de tu proyecto. En este documento, se muestra cómo puedes agrupar los recursos en tu proyecto de Android, proporcionar recursos alternativos para configuraciones de dispositivos específicos y acceder a ellos desde el código de tu app u otros archivos en formato XML posteriormente.

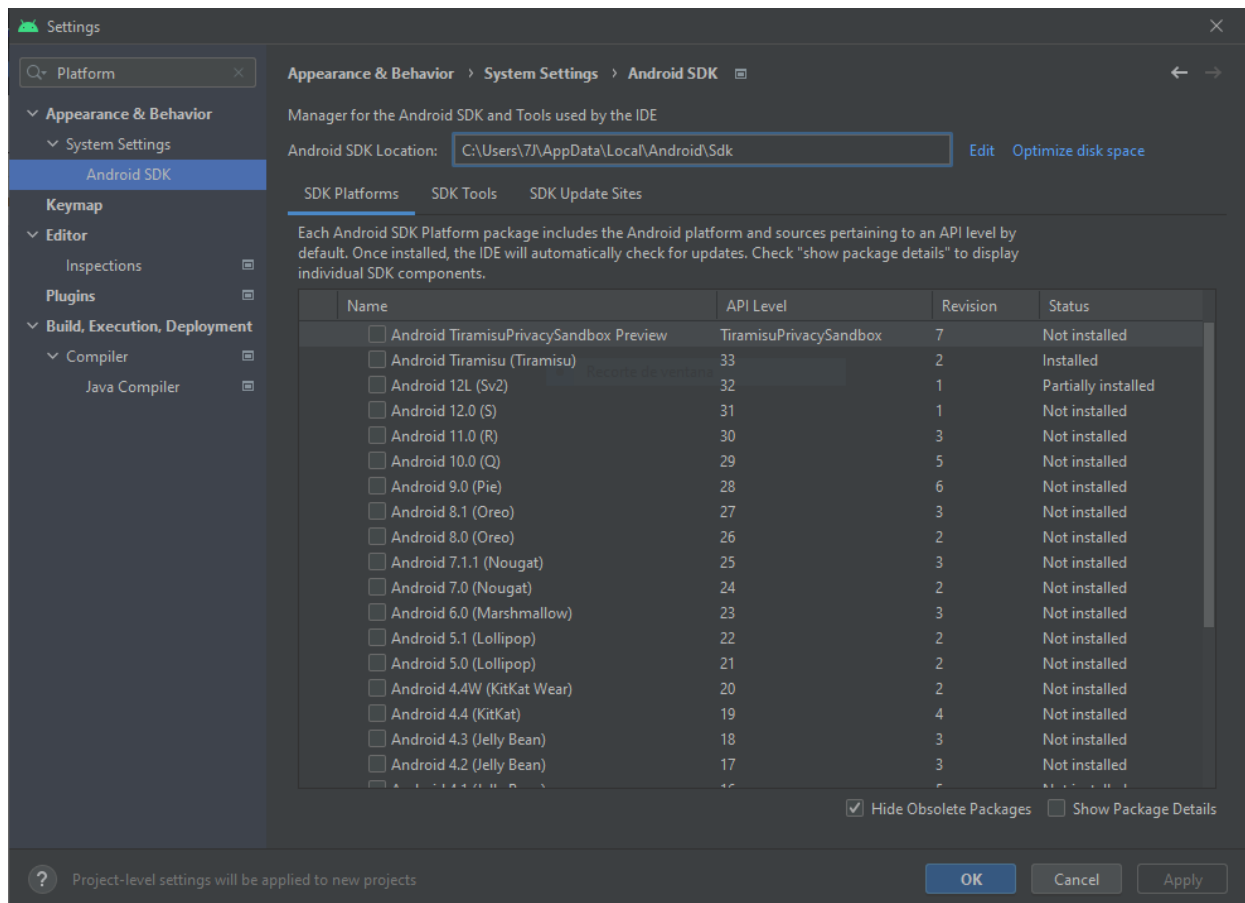
#### **Ejercicio 2.4.**

Entra en los settings de Android Studio y comprueba las Platforms de Android que tienes instaladas o parcialmente instaladas, y los SDK Tools instalados.

Investiga qué son las dependencias en Android y la inyección de dependencias, e incorpora la explicación al documento.

Inicia Android Studio y entra en el archivo build.gradle (Module: app), anota las dependencias que implementará tu entorno en las aplicaciones que desarrolles.

Captura pantallas e incorpóralas al ejercicio.



Listado de los SDK instalados.

```
dependencies {

    implementation 'androidx.appcompat:appcompat:1.3.0'
    implementation 'com.google.android.material:material:1.4.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'

}
```

Dependencias de la App, definidas en build.gradle.