ITESO, Universidad Jesuita de Guadalajara

Premio ADA BYRON A LA MUJER TECNÓLOGA

Taller de tecnología móvil

Android

Kotlin

Temario

Módulo 3

- Permisos
- Librerías
- Consumiendo APIs
- Implementación de mapas
- Publicación de aplicación en la tienda Google PlayStore

Permisos

Gestión de permisos

Cada app para Android se ejecuta en una zona de pruebas con acceso limitado. Si la app necesita usar recursos o información ajenos a su propia zona de pruebas, puedes declarar un ***permiso*** y configurar una solicitud de permiso que proporcione este acceso. Estos pasos forman parte del flujo de trabajo para usar permisos.

Los ***permisos*** de la aplicación ayudan a admitir la privacidad del usuario, ya que protegen el acceso a lo siguiente:

- **Datos restringidos:** como el estado del sistema y la información de contacto del usuario.
- **Acciones restringidas:** como conectarte a un dispositivo vinculado y grabar audio.

Ejemplo de permisos

Grupo de permisos	Permisos
Calendario	READ_CALENDAR WRITE_CALENDAR
Cámara	CAMERA
Contactos	READ_CONTACTS WRITE_CONTACTS GET_ACCOUNTS
Localización	ACCESS_FINE_LOCATION ACCESS_COARSE_LOCATION
Micrófono	RECORD_AUDIO
Teléfono	READ_PHONE_STATE CALL_PHONE READ_CALL_LOG WRITE_CALL_LOG ADD_VOICEMAIL USE_SIP PROCESS_OUTGOING_CALLS
Sensores	BODY_SENSORS
SMS	SEND_SMS RECEIVE_SMS READ_SMS RECEIVE_WAP_PUSH RECEIVE_MMS

Librerías

Inyección de dependencias

La ***inyección de dependencias (DI)*** es una técnica muy utilizada en programación y adecuada para el desarrollo de Android. Implementar la ***inyección de dependencias*** te proporciona las siguientes ventajas:

- Reutilización de código
- Facilidad de refactorización
- Facilidad de prueba

```
implementation 'com.example:examplelibrary:1.0.0'
```

Ejemplo de dependencias

- **Picasso**: Lo usaremos para cargar las imágenes de internet.
- **Retrofit**: Como ya hemos mencionado anteriormente la usaremos para la petición REST.
- **GSon**: Convertirá el *Json* en un modelo de datos fácil para poder trabajar con él.
- **CardView**: Es un componente de Android, que nos permite crear una especie de tarjetas que visualmente quedan muy bien, será el contenedor de nuestras imágenes.

```
implementation 'com.android.support:recyclerview-v7:25.0.0'
implementation 'com.squareup.picasso:picasso:2.5.2'
```

API

API Rest

Un ***API REST*** es un servicio que nos provee de las funciones necesarias para poder obtener información de un cliente externo (base de datos) dentro de nuestra propia aplicación. Tenemos distintos tipos de peticiones, los más usados son:

- **Get**: Son las más sencillas, solo nos devuelven información. Si necesitamos pasarle un parámetro a la petición será a través de la url. El problema de esto es que es poco seguro para pasar información delicada. Ej: *https://ejemplo.com/informacion/1*
- **Post**: Nos devuelven información similar a *Get* pero los parámetros no se pasan por url, por lo que es más seguro para mandar información.
- **Put**: Se suele usar para crear la entidad. Ejemplo si pensamos en un servicio como el acceso a una base de datos este crearía el usuario.
- **Delete**: Nos permite borrar los registros de la base de datos.

JSON

Json es un formato de texto simple. Se trata de uno de los estándares para el traspaso de información entre plataformas, tiene una forma muy legible que nos permite entender su contenido sin problema. Todo formato ***Json*** empieza y termina con llaves y tiene una clave-valor.

```
{
   “clave1”: “valor1”,
   “clave2”: “valor2”,
   “clave3”:  [
      {
       “clave1”: “valor1”,
        …
      },
      …
   ],
   …
}
```

```
{
  "employees": {
    "employee": [
      {
        "id": "1",
        "firstName": "Tom",
        "lastName": "Cruise",
        "photo": "https://jsonformatter.org/img/tom-cruise.jpg"
      },
      {
        "id": "2",
        "firstName": "Maria",
        "lastName": "Sharapova",
        "photo": "https://jsonformatter.org/img/Maria-Sharapova.jpg"
      },
      {
        "id": "3",
        "firstName": "Robert",
        "lastName": "Downey Jr.",
        "photo": "https://jsonformatter.org/img/Robert-Downey-Jr.jpg"
      }
    ]
  }
}
```

Manejo de hilos con Corrutinas

Cada uno de los procesos que se ejecutan en nuestra app se hacen en hilos, es decir, un conjunto de procesos que tienen su consumo de memoria y realizan X operación. Es importante saber esto porque toda la parte visual de Android, es decir, todos los componentes, interacciones y demás se ejecutan en el hilo principal, por lo que debemos intentar realizar todas las operaciones largas o potentes fuera de dicho hilo para no bloquear la interfaz del usuario.

Si hacemos una llamada a internet será no instantáneo, ya que tenemos que acceder a la API y dependiendo de nuestra red tardará algún tiempo en responder y es por ello que debemos realizar esta operación en otro hilo asíncrono, es decir, nosotros haremos la petición a nuestra API y esa lógica la haremos en un proceso fuera del hilo principal y cuando se haya terminado nos avisará. Y para hacer todo esto usaremos las ***corrutinas*.**

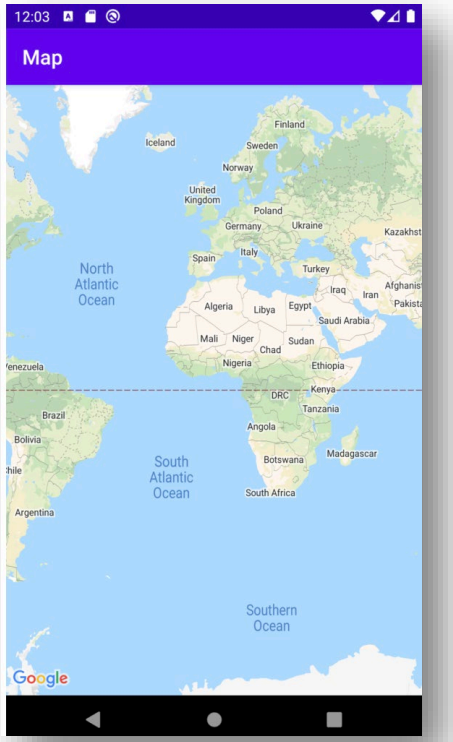
Ejercicio

Ver la práctica 1 del cuaderno de prácticas

Implementación de mapas

Google maps

El SDK de Maps para Android se distribuye como parte del SDK de Servicios de Google Play, el cual puedes agregar a través de SDK Manager. A diferencia de otras librerías que hemos usado, para poder trabajar con ***Google Maps*** necesitamos un ***api key***. Esto no es más que un código para verificar que el usuario que crea un proyecto es el creador de la aplicación.



Publicación de aplicación en la tienda Google PlayStore

Google Play Store

ITESO, Universidad Jesuita de Guadalajara

Premio
ADA BYRON
A LA MUJER TECNÓLOGA

ITESO, Universidad Jesuita de Guadalajara

Premio
ADA BYRON
A LA MUJER TECNÓLOGA