



EDUCACIÓN
CONTINUADA

Diplomado en Desarrollo de Software

Parte V

Programación Móvil Avanzada

Ruthford Jay / rjay@uninorte.edu.co



EDUCACIÓN
CONTINUADA

Programación Móvil Avanzada

Sensores

Objetivos de la Sesión

Al finalizar esta sesión estarás en capacidad de:

- Manejar los permisos del usuario
- Acceder al sensor de GPS

Permisos

Tanto en Android como en IOS antes de poder acceder a cierto hardware del dispositivo tenemos que solicitar al usuario que nos conceda privilegios para poder obtener datos de ellos.

Esto ocurre, por ejemplo, si deseamos acceder a la localización o queremos acceder a la agenda de contactos o para poder acceder a la galería multimedia, etc...

Son permisos que el usuario nos debe de conceder y que en cualquier momento nos puede revocar, por tanto siempre que deseemos utilizar estos servicios tendremos que averiguar si disponemos de los permisos necesarios en ese momento.

Permisos

Por supuesto, cada sistema operativo operativo trata esta concesión de permisos de una forma diferente y desde Flutter, por suerte, se dispone de un paquete que nos gestiona la petición de permisos para ambas plataformas de una forma única y homogénea, hablamos de Flutter Permission Handler.

https://pub.dev/packages/permission_handler

```
$ flutter pub add permission_handler
```

Permisos

En nuestro ejemplo vamos a solicitar tres permisos, estos serán: el acceso a la cámara, el acceso a los contactos y el acceso a la localización.

Para Android, los permisos se solicitan añadiéndoles en el archivo AndroidManifest.xml que está dentro del directorio de ficheros del proyecto Android.

...\\{app_name}\\android\\app\\src\\main\\AndroidManifest.xml

```
<!-- Permisos para localización -->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<!-- Permisos para camara -->
<uses-permission android:name="android.permission.CAMERA"/>
<!-- Permisos para contactos -->
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.WRITE_CONTACTS"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
```

¿Cómo Solicitar un Permiso?

Ahora nos queda mediante código solicitar propiamente el permiso al usuario, y si este no nos lo ha concedido con anterioridad, le aparecerá una alerta indicando que necesitamos dicho permiso para que nos lo conceda. Esto lo hacemos de esta forma:

```
if (await Permission.camera.request().isGranted) {  
    // Permiso concedido  
}
```

Con esta línea de código estamos verificando que el permiso esté concedido, y si no lo está, se mostrará la alerta antes mencionada.

¿Cómo Solicitar un Permiso?

Si lo que deseamos es verificar únicamente si el permiso está concedido entonces podemos utilizar la siguiente sentencia:

```
var status = await Permission.camera.status;
```

Y el status puede ser: **granted**, **denied**, **restricted** or **permanentlyDenied**

Si el usuario nos ha denegado un permiso, no podremos utilizar el recurso, pero podemos recordarle al usuario que dicho permiso es necesario para que la aplicación funcione o para que pueda mostrar determinado dato. Incluso podemos abrir la página de settings del terminal para que usuario vuelva a darnos el permiso si por lo que sea nos lo ha revocado.

Esto lo podríamos hacer de esta forma:

```
if (await Permission.camera.isPermanentlyDenied) {  
    openAppSettings();  
}
```


Geolocalización

Para acceder a la ubicación del usuario haremos uso de la dependencia geolocator, la cual nos permitirá solicitar estos datos de manera independiente a la plataforma (iOS, Android).

<https://pub.dev/packages/geolocator>

```
$ flutter pub add geolocator
```

Geolocalización

Gracias a que anteriormente se solicitaron los permisos apropiados, en esta instancia solo es necesario solicitar la ubicación del usuario con el siguiente método asincrónico:

```
await Geolocator.getCurrentLocation();
```

Este futuro permite acceder a la ubicación del usuario de forma sencilla y rápida, la cual puede ser usada para cualquier tipo de proceso, pero teniendo en cuenta que se considera que es **información sensible**, por lo cual se deben tomar las medidas necesarias para su seguridad. Si deseas acceder a un Stream también puedes hacerlo.

Geolocalización

Para acceder a la ubicación de una forma diferente es posible hacerlo usando un Stream, una metodología anteriormente explicada.

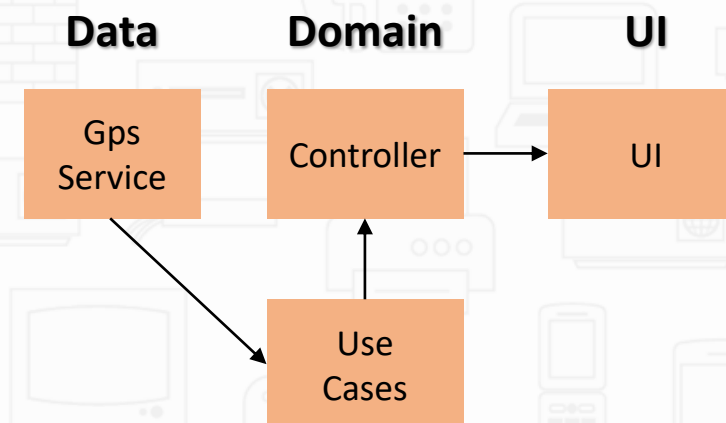
```
await Geolocator.getPositionStream();
```

El Stream permitirá escuchar por cambios en la ubicación del usuario, mediante la suscripción de listeners, así como dejar de escuchar estos cambios cuando sea necesario.

Geolocalización

Siguiendo la arquitectura que se ha trabajado desde un inicio (Clean Architecture), el flujo de datos recomendado es el siguiente:

- **Service:** Contendrá el consumo directo del servicio de ubicación incluyendo futuros y streams.
- **Use Cases:** Establece métodos para acceder a las funcionalidades del servicio desde la capa de dominio.
- **Controller:** Dentro del controlador de estado haz uso los métodos creados en el caso de uso para su uso a su vez en ui.
- **UI:** Obtener el controlador inyectado (***Get.find()***) para utilizar los métodos que creaste anteriormente.





EDUCACIÓN
CONTINUADA

Componente práctico

Aplicación demo de localización