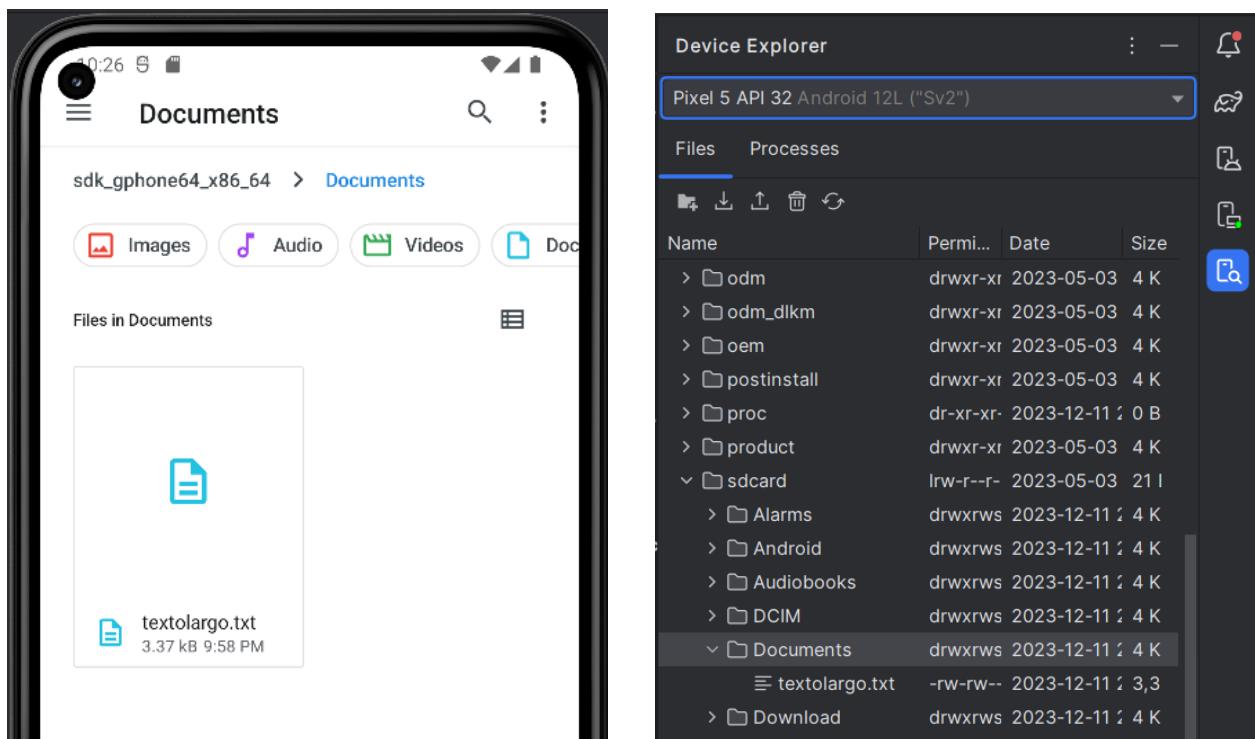


Ejercicio Lee Fichero

[Descargar estos apuntes](#)

Vamos a crear un ejercicio que permita practicar el concepto de **Intent Implicito con resultado** a la vez que usamos **corrutinas**. Lo que queremos conseguir es la carga de un archivo .txt seleccionado desde el almacenamiento del dispositivo y mostrarlo carácter a carácter en la aplicación, mientras se muestra una barra de desplazamiento con el progreso de la carga. Para ello se deberá realizar lo siguiente:

1. Al iniciar se mostrará un botón con el texto de Cargar
2. Al pulsar sobre el botón se abrirá la app administrador de archivos (Files) instalada por defecto en el sistema, para que podamos seleccionar un [archivo de texto largo](#). Previamente subido mediante el **Device Explorer** en la carpeta Documents de la memoria interna del dispositivo:



💡 Tip: Para que se inicie la aplicación se lanzará un Intent que deberemos gestionar como hemos visto en el tema, para ello crearemos el método **resgitroSelectorDeFicheroConGetContent** en el paquete **utilities.device**, la url sería `/*`

3. Una vez seleccionado el archivo, desde una función del ViewModel se pasará a la lectura de este y a su carga en una cadena, para ello utilizaremos el siguiente código:

```
@Throws(IOException::class)
private fun leeFichero(uri: Uri): String? {
    val contentResolver = getApplication<Application>().contentResolver
    contentResolver.openInputStream(uri)?.use { inputStream ->
        return BufferedReader(InputStreamReader(inputStream)).readText()
    }
    return null
}
```

⚠ Aviso: Vamos a necesitar usar la propiedad `contentResolver` definida dentro del `Context`. Por tanto, para poder acceder al mismo desde un **ViewModel** sin inyectarlo con Hilt. Tendremos que heredar nuestro **ViewModel** de **AndroidViewModel** en lugar de **ViewModel** y pasar **Application** en ambos constructores. De esta manera dentro de nuestro **ViewModel** dispondremos el método `getApplication<Application>()`

4. El siguiente paso será añadir a la pantalla de la aplicación un texto con scroll y una barra de progreso, para relacionar el movimiento de esta con la carga del texto podemos usar la longitud de este. El aspecto durante la carga sería algo como la imagen a la derecha.

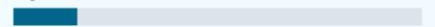
💡 Tip: Usa un `StringBuilder` como `State` para el texto que vamos cargando, añadiendo un carácter con un retardo de un milisegundo.

Si visualizas el video aclarativo, podrás ver que el scroll se realiza de abajo hacia arriba, para ello tendrás que asignar a la propiedad `reverseScrolling = true`

En este [enlace](#) encontraréis un video aclarando con el funcionamiento.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam faucibus vel orci eget hendrerit. Nulla lobortis vulputate nulla, eget sollicitudin sapien blandit in. Maecenas sed ex erat. Ut sed odio a nulla efficitur lobortis. Ut lacinia lobortis leo, vitae elementum lacus ornare ac. Mauris sit amet facilisis nulla, in pellentesque tortor. Ut neque dolor, rutrum et condimentum in, sagittis eget arcu. Nulla elit lorem, aliquet in hendrerit sed, tincidunt et felis. Mauris fringilla, dolor in consequat finibus, dolor elit

Leyendo...



15%