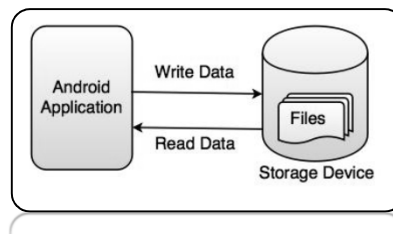


Almacenamiento interno (1)

Podemos almacenar información mediante archivos que se alojarán en la memoria interna del dispositivo. Los archivos guardados por una aplicación, por defecto no estarán accesibles para el resto de aplicaciones, y se almacenarán en la siguiente ruta:

data/data/nombre_app/files/nombre_archivo

Si desinstalamos la aplicación, estos archivos se eliminarán.



11

Almacenamiento interno (2)

■ Escritura de un archivo

- 1) Creamos un objeto *FileOutputStream* y abrimos el fichero mediante *openFileOutput* que recibirá como parámetros el nombre del archivo y el modo de apertura (privado, público en modo lectura, público en modo escritura o en adición).
- 2) Utilizamos el método *write* para escribir los datos en el archivo.
- 3) Cerramos el archivo.

Estas operaciones pueden producir dos excepciones:

- No se encuentra el archivo.
- No se puede escribir (permisos).

12

Almacenamiento interno (3)

■ Escritura de un archivo

```
FileOutputStream fos;  
try {  
    fos = openFileOutput("miArchivo.txt", Context.MODE_PRIVATE);  
    fos.write(texto.getBytes());  
    fos.close();  
} catch (FileNotFoundException e) {  
    e.printStackTrace();  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

13

Almacenamiento interno (4)

■ Lectura de un archivo

- 1) Creamos un objeto *FileInputStream* y abrimos el fichero mediante *openFileInput* que recibirá como parámetro el nombre del archivo.
- 2) Repetimos el proceso de lectura utilizando *read* y un *buffer* hasta que no quede contenido por leer en el archivo.
- 3) Cerramos el archivo.

Estas operaciones pueden producir dos excepciones:

- No se encuentra el archivo.
- No se puede leer (permisos).

14

Almacenamiento interno (5)

■ Lectura de un archivo

```
FileInputStream fis;  
try {  
    fis = openFileInput("miArchivo.txt");  
    byte[] buffer = new byte[1024];  
    StringBuilder texto = new StringBuilder();  
    while ((fis.read(buffer))!=-1) {  
        texto.append(new String(buffer));  
    }  
} catch (FileNotFoundException e) {  
    e.printStackTrace();  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

15

Almacenamiento interno (6)

■ Archivos caché

Si lo que deseamos es almacenar datos temporales que podamos utilizar durante la ejecución pero que no se almacenen de forma permanente deberemos utilizar los métodos *getCacheDir()* para obtener el directorio de caché interno de la aplicación y *createTempFile()* para almacenar los datos.

Hay que tener en cuenta que si tenemos problemas de espacio, Android puede eliminar estos archivos.

```
File archivo;  
archivo = File.createTempFile("nombreArchivo", null, this.getCacheDir());
```

16