

Almacenamiento de datos

Ph. D. M^a del Puerto Paule Ruiz

Dpto. Computer Science

University of Oviedo

Almacenamiento de Datos

- En Android hay diversas opciones para realizar la persistencia de datos:
 - Shared Preferences
 - Almacenamiento interno
 - Almacenamiento externo
 - SQLite
- La elección depende de las necesidades específicas de cada aplicación

Shared Preferences

- Las preferencias son datos de una aplicación que facilitan la experiencia de usuario
- Permiten almacenar de forma persistente pares clave-valor de tipos primitivos de datos
- Estos datos persisten aunque nuestra aplicación termine
- El API para almacenar y recuperar preferencias en Android es *SharedPreferences*

Shared Preferences

- Para obtener un objeto *SharedPreferences* utilizamos el método *getSharedPreferences (String name, int mode)*

```
SharedPreferences mSharedPreferences =  
    getSharedPreferences("MyPreferences", Context.MODE_PRIVATE);
```

- Tres modos de acceso:
 - Context.MODE_PRIVATE
 - Context.MODE_WORLD_READABLE (Deprecated API 17)
 - Context.MODE_WORLD_WRITEABLE (Deprecated API 17)

SharedPreferences

- Una vez que tenemos el objeto *SharedPreferences*, podemos obtener/insertar o modificar preferencias con los métodos get/put correspondientes al tipo básico de datos
- Obtener datos:

```
String address = mSharedPreferences.getString ("address", "Default address");
```
- Para almacenar o modificar preferencias se hace con `SharedPreferences.Editor()`
 - Accedemos a él mediante el método `mSharedPreferences.edit()`

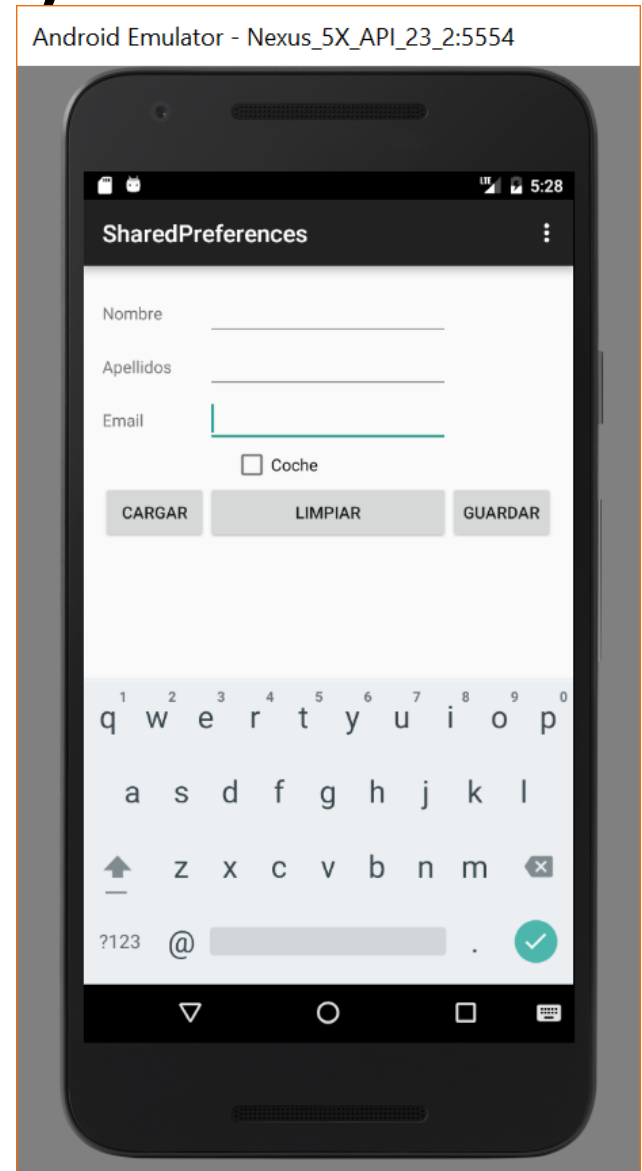
SharedPreferences

- Una vez obtenida la referencia al editor, mediante los métodos put de los tipos básicos de datos añadiremos/modificaremos preferencias
 - Es preciso llamar al método commit() de la clase Editor para que los cambios tengan efecto

```
final SharedPreferences.Editor mEditor =  
    mSharedPreferences.edit();  
mEditor.putString("address", "myAddress");  
mEditor.commit();
```

Ejercicio (I)

- Crear un formulario que permita introducir datos personales a un usuario
 - Al pulsar el botón de “Guardar” se guardarán los datos en SharedPreferences
 - Al pulsar el botón de “Cargar” se cargarán los datos en SharedPreferences



Ejercicio II

- Crear un formulario que permita introducir datos personales a un usuario
 - Al pulsar el botón de “Guardar” se guardarán los datos en SharedPreferences
 - Al pulsar el botón de “Cargar” se cargarán los datos en SharedPreferences en una segunda activity

Almacenamiento de fichero

- Necesidad de almacenar información “no estructurada” con otro tipo de formato
- Dos lugares:
 - Memoria interna del dispositivo
 - Memoria externa (tarjeta SD): Permiso en el manifiesto
- Acceso a través de:
 - `FileOutputStream openFileOutput (String fileName, int mode)`
 - `FileInputStream openFileInput (String fileName)`

SQLite

- Android nos ofrece soporte para BD con un API
- La BD que se crea desde la aplicación sólo es accesible desde ella, no desde fuera
- Se suele crear en la carpeta del proyecto

SQLite

- SQLiteOpenHelper: Nos ofrece métodos para crear y actualizar la base de datos
 - onCreate (SQLiteDataBase db)
 - onUpdate (SQLiteDataBase db, int versionAnterior, int versionActual)
- Es preciso crear una clase que extienda de SQLiteOpenHelper

SQLiteDatabase

- SQLiteDatabase proporciona los métodos básicos para manejar una base de datos
 - Insert, update, delete,
- Para obtener un objeto SQLiteDatabase lo hacemos a través de SQLiteOpenHelper
 - SQLiteDatabase getReadableDatabase()
 - SQLiteDatabase getWritableDatabase()

SQLiteDatabase (Insert)

- Métodos para operaciones básicas

```
long insertId = database.insert(String table,String column,  
ContentValues values);
```

ContentValues

Par Columna/Valor que nos permite establecer valores para las columnas a insertar en la base de datos

```
ContentValues values = new ContentValues();  
values.put(MyDBHelper.COLUMN_MESSAGE, "message");  
values.put(MyDBHelper.COLUMN_USER, "user");
```

SQLiteDatabase (Consulta)

```
Cursor cursor = database.query(MyDBHelper.TABLE_VALORATIONS,  
    allColumns, null, null, null, null, null);
```

Cursor:

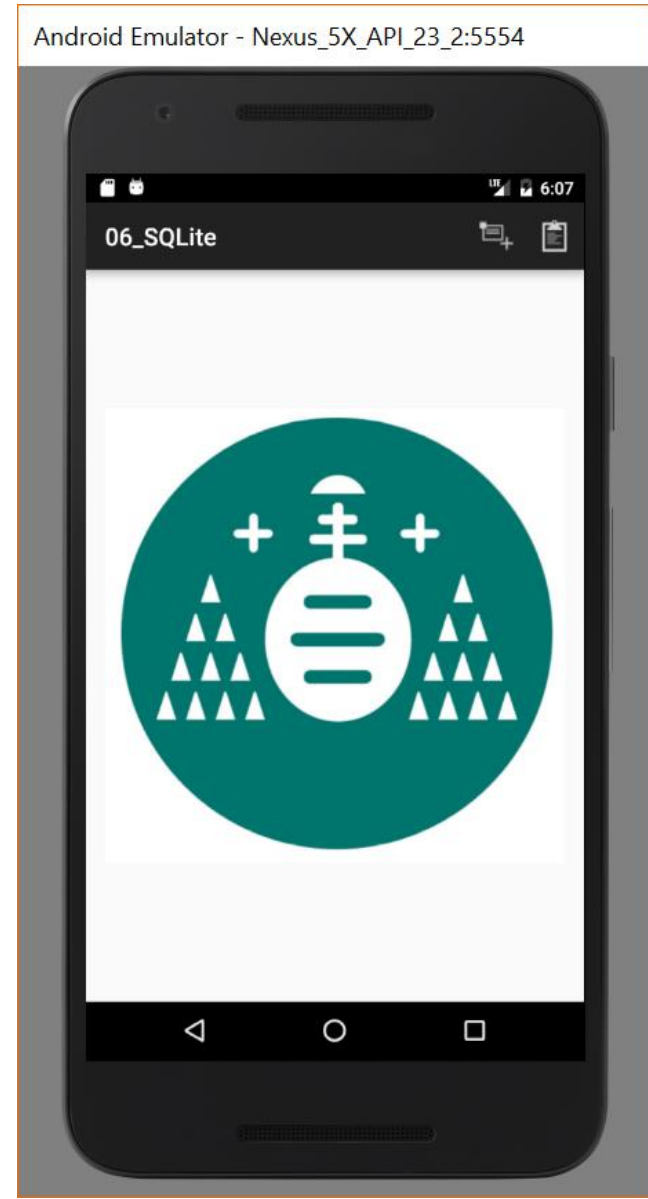
Interfaz para leer o escribir el resultado de la consulta

Proporciona **métodos** para:

1. Situar al principio: `cursor.moveToFirst()`
2. Moverse al siguiente: `cursor.moveToNext()`
3. Saber que hemos llegado al final: `cursor.isAfterLast()`

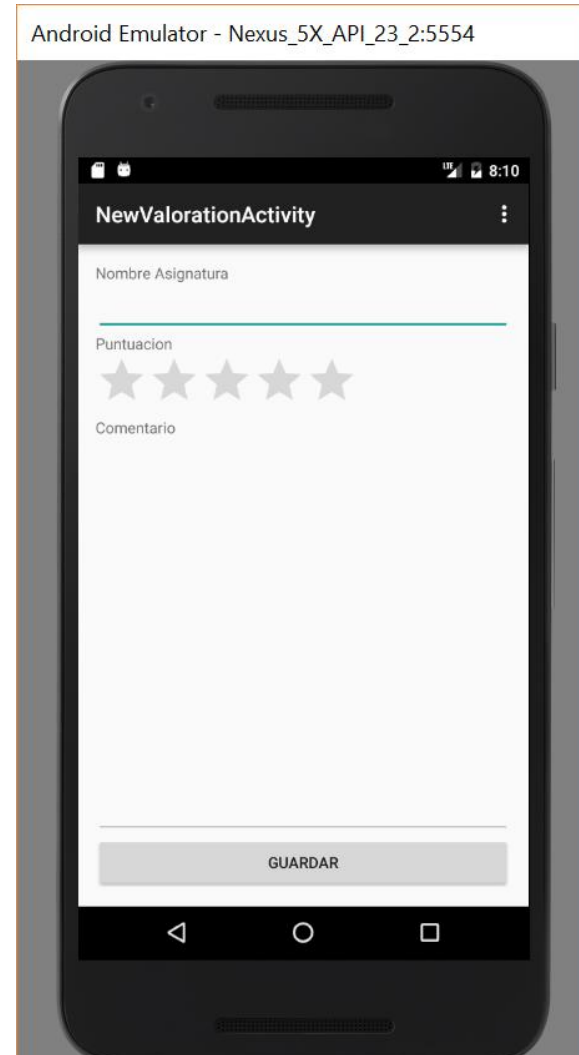
Ejercicio

- Valoraciones
 - Realizar un ejercicio que permite realizar valoraciones
 - Permite ver todas las valoraciones realizadas



Layouts

- Mostrar todas las valoraciones/Insertar una valoración



Ejercicio (FrontEnd)

1. Crear opciones de menú en MainActivity
2. Crear NewValorationActivity: Añadir una Valoración
3. Crear AllValorationActivity

Ejercicio (BackEnd-BD)

- Clase MyDBHelper (implements SQLiteOpenHelper):
 - Definición del esquema de la BD “valuation”
 - Métodos para crear y manejar la BD
- Clase Valoration:
 - Get/Set atributos
 - toString: para mostrar en el listView
- Clase ValorationDataSource:
 - DAO sobre la tabla “valuation”
 - Abrir y cerrar conexión en la BD, así como consultas sobre la tabla “valuation”

Ejercicio (cont.)

- Completar las activities NewValorationActivity y AllValorationActivity