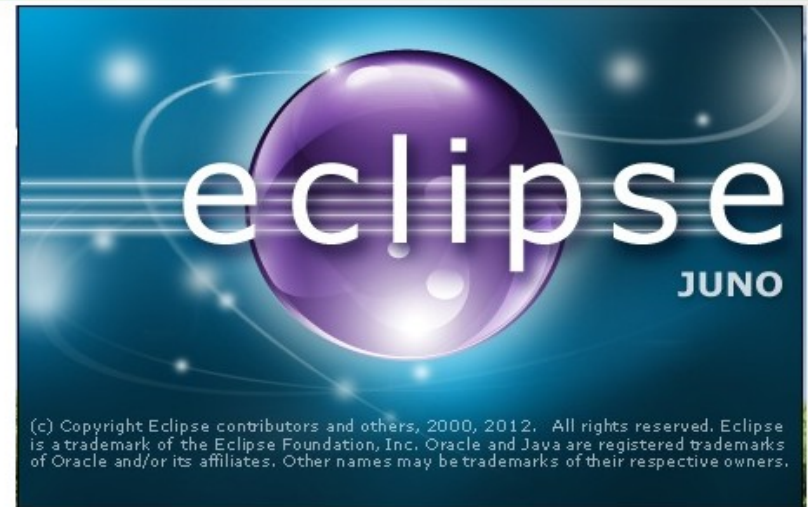




ESEI.Dojo.Android



Introducción a la programación para Android
por Baltasar García Perez-Schofield
Laboratorio de libre acceso, 9/11, 16h
Viernes 9 de Noviembre de 2012

- Persistencia de datos

- Preferencias
- Memoria interna del dispositivo.
- Memoria externa del dispositivo.
- SQL Lite.

- Preferencias
 - Se basan en la clase **SharedPreferences**.
 - Instancia con: `getPreferences(MODE_PRIVATE)`
 - Para modificar, se usa la clase **SharedPreferences.Editor**
 - Que tiene `putInt()`, `putString()`,...
 - Para leer, se usa directamente `getInt()`, `getString()`
 - Como segundo parámetro, se indica el valor a devolver si no se encuentra.

```
SharedPreferences prefs = this.getSharedPreferences( PRIVATE_MODE );
SharedPreferences.Editor editor = prefs.edit();
editor.putInt( "id1", 1 );
editor.commit();
int i = prefs.getInt( "id1", -1 );
```

- Memoria interna del dispositivo

- Son archivos normales.
- `openFileOutput(FileName, Context.MODE_PRIVATE);`
- `openFileInput(FileName);`

```
InputStreamReader inf = new InputStreamReader( f );
BufferedReader input = new BufferedReader( inf );
ArrayList<Tarea> toret = new ArrayList<Tarea>();
String line = input.readLine();
while( line != null ) {
    int pos = line.indexOf( '|' );
    Tarea t = Tarea.decodificar(
        line.substring( 0, pos ), line.substring( pos +1 ) );
    toret.add( t );

    line = input.readLine();
}
```

- Memoria externa

- Tomar el directorio de la aplicación:
 - `getExternalFilesDir(null)`
 - Existen otros directorios: `MUSIC_DIRECTORY...`
- Son archivos normales, por lo que es muy similar a la memoria interna.

```
FileOutputStream fos = null;
File file = new File( this.act.getExternalFilesDir( null ), FileName
);
fos = new FileOutputStream( file );
OutputStreamWriter of = new OutputStreamWriter( fos );
BufferedWriter output = new BufferedWriter( of );
output.write( t.getId() + '|' + t.codificar() + '\n' );
output.close();
```

- SQL Lite

- Crear una clase derivada de **SQLiteOpenHelper**

- getWritableDatabase()
 - query(), execSQL(), insert(), replace(), update(), delete()
 - Las query's devuelven objetos Cursor, que permiten inspeccionar los resultados.

```
public class ListaTareasDb extends SQLiteOpenHelper {
    public void onCreate(SQLiteDatabase db) {
        db.execSQL( "CREATE TABLE IF NOT EXISTS " + NombreTabla +
"( "
        + CampoId + " TEXT PRIMARY KEY, "
        + CampoTitulo + " TEXT, "
        + CampoDesc + " TEXT);" );
    }
}
```