



“Android – Almacenamiento de Información”

Rogelio Ferreira Escutia



Tipos de Almacenamiento



Tipos de Almacenamiento

- **File I/O**
- **Shared Preferences**
- **Content Providers**
- **Bases de Datos**



File I/O



Lugares de Almacenamiento

- **Interna**
- **Externa (SD Card)**



Tipos de apertura de archivos

- **MODE_PRIVATE**
- **MODE_APPEND**
- **MODE_WORLD_WRITABLE**
- **MODE_WORLD_READABLE**



MODE_PRIVATE

- **Es el modo por defecto con acceso privado desde nuestra aplicación.**
- **Sobreescribe el contenido del archivo.**



MODE_APPEND

- **Para añadir contenido a un archivo existente.**
- **Si no existe el archivo lo crea.**



MODE_WORLD_WRITABLE

- **Permite a otras aplicaciones escribir en el archivo.**



MODE_WORLD_READABLE

- **Permite a otras aplicaciones leer del archivo.**



SQLite





Small. Fast. Reliable.
Choose any three.

[About](#) [Documentation](#) [Download](#) [License](#) [Support](#) [Purchase](#)

Search SQLite Docs...

Go

SQLite is a software library that implements a [self-contained](#), [serverless](#), [zero-configuration](#), [transactional](#) SQL database engine. SQLite is the [most widely deployed](#) database engine in the world. The source code for SQLite is in the [public domain](#). [More...](#)

Sponsors

Continuing enhancement and maintenance of SQLite is made possible by [SQLite Consortium](#) members, including:



Bloomberg



mozilla



Current Status

- [Version 3.12.2](#) of SQLite is recommended for all new development.

Common Links

- [Features](#)
- [When to use SQLite](#)
- [Frequently Asked Questions](#)
- [Getting Started](#)
- [SQL Syntax](#)
 - [Pragmas](#)
 - [SQL functions](#)
 - [Date & time functions](#)
 - [Aggregate functions](#)
 - [JSON functions](#)
- [C/C++ Interface Spec](#)
 - [Introduction](#)
 - [List of C-language APIs](#)
- [The TCL Interface Spec](#)
- [Development Timeline](#)



Bibliotecas utilizadas

- **import android.content.Context;**
- **import android.database.sqlite.SQLiteDatabase;**
- **import android.database.sqlite.SQLiteOpenHelper;**
- **import android.provider.BaseColumns;**
- **import android.util.Log;**



Operaciones Básicas

- **Insert():** Inserts one or more rows into the database.
- **Query():** Requests rows matching the criteria you specify.
- **Update():** Replaces ones or more rows that match the criteria you specify.
- **Delete():**Deletes rows matching the criteria you specify



Clase para SQLite

- **public class DBHelper1 extends SQLiteOpenHelper { //**



Estructura de la Base

- **static final String TAG = "DbHelper";**
- **static final String DB_NAME = "timeline.db"; // static**
final int DB_VERSION = 1; //
- **static final String TABLE = "timeline"; // static final**
String C_ID = BaseColumns._ID;
- **static final String C_CREATED_AT = "created_at";**
static final String C_SOURCE = "source";
- **static final String C_TEXT = "txt";**
- **static final String C_USER = "user";**

Context context;



Inicializar Base en el constructor

- **// Constructor**

```
public DBHelper1(Context context) { //  
super(context, DB_NAME, null, DB_VERSION);  
this.context = context; }
```



Creación de la Base

- **// Called only once, first time the DB is created**
@Override
public void onCreate(SQLiteDatabase db) {
String sql = "create table " + TABLE + " (" + C_ID + " int
primary key, " + C_CREATED_AT + " int, " + C_USER + "
text, " + C_TEXT + " text)"; //
db.execSQL(sql); //
Log.d(TAG, "onCreated sql: " + sql); }



Verificar la existencia de la Base

```
▪ // Called whenever newVersion != oldVersion
@Override
public void onUpgrade(SQLiteDatabase db, int
oldVersion, int newVersion) { //
// Typically do ALTER TABLE statements, but...we're just
in development, // so:
db.execSQL("drop table if exists " + TABLE); // drops the
old database Log.d(TAG, "onUpdated");
onCreate(db); // run onCreate to get new database
```



Abrir base para abrir

- **// Open the database for writing**
db = dbHelper.getWritableDatabase(); /



Ejecución

- **@Override**

```
public void onUpgrade(SQLiteDatabase db, int  
oldVersion, int newVersion) {  
db.execSQL("drop table " + TABLE);  
this.onCreate(db); }
```



Cerrar la base

- **// Close the database `db.close();` //**





Rogelio Ferreira Escutia

***Instituto Tecnológico de Morelia
Departamento de Sistemas y Computación***

***Correo: rogelio@itmorelia.edu.mx
 rogeplus@gmail.com***

***Página Web: http://antares.itmorelia.edu.mx/~kaos/
 http://www.xumarhu.net/***

Twitter: http://twitter.com/rogeplus

Facebook: http://www.facebook.com/groups/xumarhu.net/