
ACTIVIDAD	:	Laboratorio Nro. 02
TEMA	:	Base de Datos
DOCENTE	:	Ing. Eric Gustavo Coronel Castillo

INDICE

OBJETIVO.....	2
DESCRIPCIÓN	2
CREACIÓN DEL PROYECTO	2
CLASE ENTITY	2
CLASE HELPER	4
CLASE DAO	5
INTERFAZ DE USUARIO	6
Layout	6
Activity.....	6
Método onCreate	6
Método crearBaseDeDatos	6
EJECUCIÓN DE LA APLICACIÓN	7
REVISIÓN DE LA BASE DE DATOS	7
Archivo de la base de datos	7
Consola de comandos.....	8

OBJETIVO

En este laboratorio el participante aprende a crear bases de datos con SQLite.

DESCRIPCIÓN

Debe desarrollar una aplicación que permita crear una base de datos SQLite con una tabla de contactos, agregarle algunos registros y verificar que todo funciona correctamente.

CREACIÓN DEL PROYECTO

- Crear un nuevo proyecto con las siguientes especificaciones iniciales:

Application Name	Base De Datos
Project Name	Lab02_BaseDeDatos
Package Name	pe.egcc.basededatos

- El icono según su criterio.
- Una actividad en blanco con las siguientes especificaciones:

Activity Name	MainActivity
Layout Name	activity_main
Navigation Type	None

CLASE ENTITY

La clase entidad es la siguiente:

```
package pe.egcc.basededatos;

public class Contacto {

    private int id;
    private String nombre;
    private String telefono;
    private String email;

    public Contacto() {
    }
}
```

Android Avanzado

```
public Contacto(String nombre, String telefono, String email) {
    super();
    this.nombre = nombre;
    this.telefono = telefono;
    this.email = email;
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public String getTelefono() {
    return telefono;
}

public void setTelefono(String telefono) {
    this.telefono = telefono;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}
}
```

CLASE HELPER

Se trata de una clase que representa la base de datos, debe heredar de la clase **SQLiteOpenHelper** y tiene un constructor que normalmente no es necesario sobrescribir, y dos métodos abstractos, `onCreate()` y `onUpgrade()`, que se deben personalizar con el código necesario para crear la base de datos y actualizar su estructura respectivamente.

A continuación tenemos el script:

```
package pe.egcc.basededatos;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.database.sqlite.SQLiteOpenHelper;

public class ContactosSQLiteHelper extends SQLiteOpenHelper {

    // Sentencia SQL para crear la tabla de Contactos
    private String sqlCreateTable = "CREATE TABLE contacto ( "
        + "id integer primary key autoincrement, "
        + "nombre TEXT, telefono TEXT, email TEXT )";
    private String sqlDropTable = "DROP TABLE IF EXISTS contacto";

    // Constructor
    public ContactosSQLiteHelper(Context context, String name,
        CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // Se ejecuta la sentencia SQL de creación de la tabla
        db.execSQL(sqlCreateTable);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // Se elimina la versión anterior de la tabla
        db.execSQL(sqlDropTable);
        // Se crea la nueva versión de la tabla
        db.execSQL(sqlCreateTable);
    }
}
```

```
}
```

CLASE DAO

Crear la clase DAO donde se implemente solo el método para insertar registros.

El script es el siguiente:

```
package pe.egcc.basededatos;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;

public class ContactoDao {

    // Base de datos
    private final String DB_NAME = "DBContactos";
    private final int DB_VERSION = 1;
    private ContactosSQLiteHelper dbHelper = null;
    private SQLiteDatabase db = null;

    public ContactoDao(Context context) {
        dbHelper = new ContactosSQLiteHelper(context, DB_NAME, null, DB_VERSION);
    }

    public void open() {
        try {
            // Abrimos la base de datos en modo escritura
            db = dbHelper.getWritableDatabase();
        } catch (Exception e) {
            throw new RuntimeException("Error al abrir la base de datos.");
        }
    }

    public void close() {
        try {
            dbHelper.close();
        } catch (Exception e) {
        }
    }

    public void insert(Contacto contacto) {
        try {
```

```
String sql = "INSERT INTO contacto(nombre,telefono,email) "
    + "values('" + contacto.getNombre() + "','"
    + contacto.getTelefono() + "','" + contacto.getEmail()
    + "');"
db.execSQL(sql);
} catch (Exception e) {
    // e.printStackTrace();
    throw new RuntimeException("Error en el proceso de inserción. ");
}
}
```

INTERFAZ DE USUARIO

Layout

En la interfaz de usuario solo contiene un TextView de id tvMensaje.

Activity

Método onCreate

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    crearBaseDeDatos();
}
```

Método crearBaseDeDatos

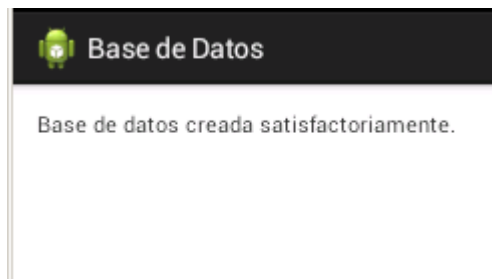
```
private void crearBaseDeDatos() {

    TextView tvMensaje = (TextView) findViewById(R.id.tvMensaje);
    ContactoDao dao = null;
    try {
        dao = new ContactoDao(this);
        dao.open();
        dao.insert(new Contacto("Gustavo Coronel", "996664457",
            "gcoronelc@gmail.com"));
    }
```

```
dao.insert(new Contacto("Ricardo Marcelo", "974567234",  
    "rmarcelo@gmail.com"));  
dao.insert(new Contacto("Hugo Valencia", "674523678",  
    "hvalencia@gmail.com"));  
dao.insert(new Contacto("Sergio Matsukawa", "878567873",  
    "smatsukawa@gmail.com"));  
dao.insert(new Contacto("Cesar Bustamante", "985678345",  
    "cbustamante@gmail.com"));  
tvMensaje.setText("Base de datos creada satisfactoriamente.");  
} catch (Exception e) {  
    tvMensaje.setText(e.getMessage());  
} finally {  
    try {  
        dao.close();  
    } catch (Exception e2) { }  
}  
}
```

EJECUCIÓN DE LA APLICACIÓN

Al ejecutar la aplicación se debe obtener el mensaje



REVISIÓN DE LA BASE DE DATOS

Archivo de la base de datos

Las bases de datos SQLite creadas por aplicaciones Android utilizando este método se almacenan en la memoria del teléfono en un fichero con el mismo nombre de la base de datos situado en una ruta que sigue el siguiente patrón:

```
/data/data/[paquete.aplicación.java]/databases/[nombre_base_datos]
```

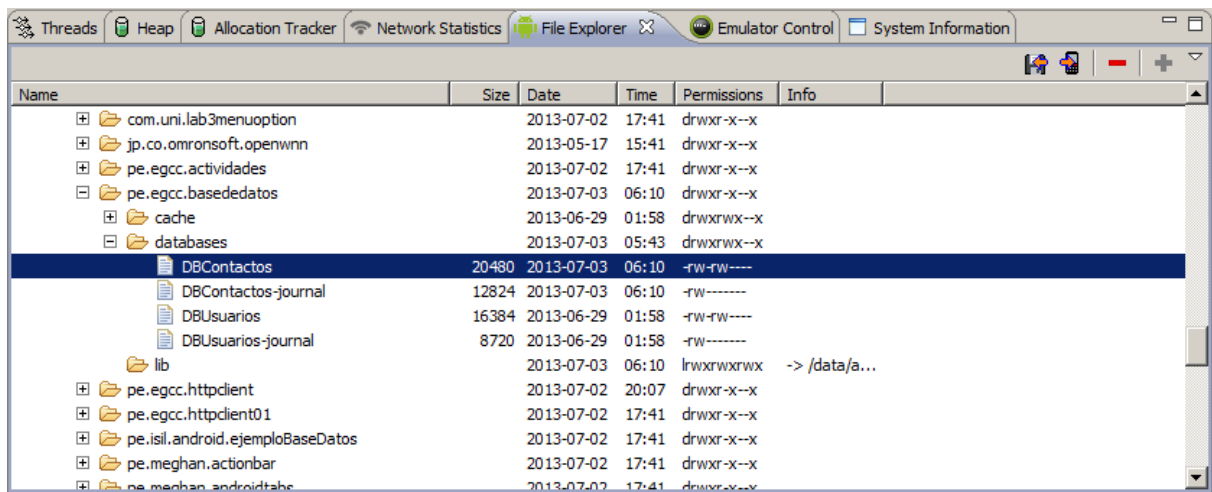
Android Avanzado

Para el caso de esta aplicación, la base de datos se encuentra en la siguiente ruta:

```
/data/data/pe.egcc.basededatos/databases/ DBContactos
```

Para comprobarlo se debe hacer lo siguiente.

Una vez ejecutada la aplicación sobre el emulador de Android y sin finalizarla, activar la perspectiva "DDMS" (Dalvik Debug Monitor Server) de Eclipse y desde la pestaña **File Explorer** se puede acceder al sistema de archivos del emulador, donde se debe localizar la ruta de la base de datos, tal como se visualiza en la siguiente imagen:



Con esto se verifica que el archivo de la base de datos se ha creado en la ruta correcta. Para comprobar que tanto las tablas creadas como los datos insertados también se han incluido correctamente en la base de datos, se debe utilizar dos posibles métodos:

- Transferir la base de datos a la PC y consultarla con cualquier administrador de bases de datos SQLite.
- Acceder directamente a la consola de comandos del emulador de Android y utilizar los comandos existentes para acceder y consultar la base de datos SQLite.

Consola de comandos

Con el emulador de Android abierto, se debe abrir una consola de MS-DOS y utilizar la utilidad **adb.exe** (Android Debug Bridge) situada en la carpeta platform-tools del SDK de Android (por ejemplo: C:\Resources\adt-bundle-windows-x86-20130219\sdk\platform-tools\).

Luego es necesario averiguar el identificar del emulador mediante el comando "**adb devices**". Este comando devuelve el identificador del emulador abierto:

```
C:\Resources\adt-bundle-windows-x86-20130219\sdk\platform-tools>adb devices
List of devices attached
emulator-5554    device
```


Android Avanzado

A continuación, después de conocer el identificador del emulador, procedemos a acceder a su shell mediante el comando "**adb -s identificador-del-emulador Shell**".

```
C:\Resources\adt-bundle-windows-x86-20130219\sdk\platform-tools>adb -s emulator-5554 shell
root@android:/ #
```

Después de establecer la conexión se pueden se puede acceder a la base de datos, el comando a utilizar es **sqlite3** pasándole la ruta del archivo de base de datos, por ejemplo "**sqlite3 /data/data/pe.egcc.basededatos/databases/DBContactos**". Si todo se ejecuta correctamente debe aparecer el prompt de SQLite "**sqlite>**", lo que nos indicará que ya se puede escribir las consultas SQL necesarias sobre la base de datos.

```
root@android:/ # sqlite3 /data/data/pe.egcc.basededatos/databases/DBContactos
sqlite3 /data/data/pe.egcc.basededatos/databases/DBContactos
SQLite version 3.7.11 2012-03-20 11:35:50
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite>
```

Como ejemplo consultaremos la tabla de contactos:

```
sqlite> select * from contacto;
select * from contacto;
1|Gustavo Coronel|996664457|gcoronelc@gmail.com
2|Ricardo Marcelo|974567234|rmarcelo@gmail.com
3|Hugo Valencia|674523678|hvalencia@gmail.com
4|Sergio Matsukawa|878567873|smatsukawa@gmail.com
5|Cesar Bustamante|985678345|cbustamante@gmail.com
```

