

Talento Digital Android 2021-06-30 (Mié) Módulo 3 - Clase 07

Libreta: GTD

Creado: 29-07-2021 10:42

Actualizado: 29-07-2021 10:44

Autor: Sebastián

Talento Digital Android 2021-06-30 (Mié) Módulo 3 - Clase 07

Día 3, semana 10

Referencias:

La Geekipedia De Ernesto

Curso Android desde cero #30 | Base de datos - Altas (SQLite) en Android

<https://www.youtube.com/watch?v=TxkdWX3UaNk&list=PLyvsggKtwbLX06iMtXnRGX5lyjiiMaT2y&index=31>

TAREA:

Resumen:

Trabajar con base de datos SQLite.

1) Crear registro.

Clase 1: Crear la base de datos. `Adminbd.java` / `AdminSQLiteOpenHelper.java`, etc

```
public class Adminbd extends SQLiteOpenHelper {
    public Adminbd(Context context, @Nullable String nombre_base_de_datos, @Nullable SQLiteDatabase.CursorFactory factory,
        super(context, nombre_base_de_datos, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(String sql_para_crear_tabla);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    }
}
```

Clase 2: Actividad correspondiente.

```
public void crear(View v){

    // 1.- Abrir la base de datos.
    Adminbd admin = new Adminbd(this, String nombre_base_de_datos, null, 1);
    SQLiteDatabase base = admin.getWritableDatabase();

    // 2.- Validar un componente haya recibido un dato.
    if(!componente.isEmpty())

    // 3.- Crear el registro a insertar.
    ContentValues registro = new ContentValues();

    // 4.- Completar valores del registro a insertar.
    crear.put(String campo, valor);

    // 5.- Insertar registro.
    base_de_datos.insert(String tabla, null, registro);

    // 6.- Cerrar base de datos.
    base_de_datos.close();

    // 7.- Borrar el texto de un componente.
    componente.setText("");

    // EXTRA: Restituir el texto de un componente a su valor inicial.
    componente.getText().clear()
}
```

2) Buscar registro.

Dentro de **MainActivity.java**.

```
public void buscar(View v){
    //1.- Abrir la base de datos.
    Adminbd admin = new Adminbd(this, String nombre_base_de_datos, null, 1);
    SQLiteDatabase base = admin.getWritableDatabase();

    //2.- Obtener el valor a buscar en la tabla.
    String valor = componente.getText().toString();

    //3.- Confirmar que el valor a buscar no esté vacío.
    if(!valor.isEmpty()){

        //4.- Ejecutar consulta para ver el nombre y precio de un producto cuyo código coincida con el entregado.
        Cursor fila = base.rawQuery
            ("select nombre, precio from producto where codigo =" + codigo, null);
        //5.- En caso de que el código de producto sea encontrado, mostrar los valores obtenidos.
        if(fila.moveToFirst()){
            ednombre.setText(fila.getString(0));
            edprecio.setText(fila.getString(1));
            base.close();
        }
        //6.- Informar si el código no es encontrado.
        } else {
            Toast.makeText(this, "No existe el producto", Toast.LENGTH_LONG).show();
            base.close();
        }
        //7.- Si el código no ha sido ingresado, solicitarlo.
    } else {
        Toast.makeText(this, "Debes ingresar un código de producto", Toast.LENGTH_LONG).show();
    }
}
```

Clase:

Extra: "Limpiar" el texto ingresado en un componente (reemplazar texto ingresado con el texto original del componente):

```
getText().clear()
```

Crear y buscar un registro en una base de datos.



1) Crear registro en base de datos.

Clase 1: Clase para crear la base de datos (no corresponde a una actividad por sí misma).

Adminbd.java

```
package com.example.usarbd;

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

import androidx.annotation.Nullable;

public class Adminbd extends SQLiteOpenHelper {
    public Adminbd(Context context, @Nullable String name, @Nullable SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table producto(codigo int primary key, nombre varchar, precio int)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    }
}
```

Clase 2: Actividad principal.

MainActivity.java

```
package com.example.usarbd;

import android.content.ContentValues;
import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
```

```

import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private EditText edcodigo, ednombre, edprecio;

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

        // Componentes y variables para ingresar datos a la base de datos.
        edcodigo = findViewById(R.id.editcodigo);
        ednombre = findViewById(R.id.editnombre);
        edprecio = findViewById(R.id.editprecio);
    }

    public void crear(View v){
        // Abrir la base de datos.
        Adminbd admin = new Adminbd(this, "Productos", null, 1);
        SQLiteDatabase base = admin.getWritableDatabase();

        String codigo = edcodigo.getText().toString();
        String nombre = ednombre.getText().toString();
        String precio = edprecio.getText().toString();

        // Validar que los datos a ingresar hayan sido ingresados.
        if(!codigo.isEmpty() && !nombre.isEmpty() && !precio.isEmpty()){

            // Crear el registro a insertar.
            ContentValues crear = new ContentValues();

            // Completar valores del registro a insertar.
            crear.put("codigo", codigo);
            crear.put("nombre", nombre);
            crear.put("precio", precio);

            // Insertar registro.
            base.insert("producto", null, crear);

            // Cerrar base de datos.
            base.close();

            // Limpiar los componentes luego de usar los datos.
            edcodigo.setText("");
            ednombre.setText("");
            edprecio.setText("");

            // Confirmación de registro exitoso.
            Toast.makeText(this, "Producto creado", Toast.LENGTH_LONG).show();
        } else{
            // Solicitar completar los campos.
            Toast.makeText(this, "Debes llenar todos los campos", Toast.LENGTH_LONG).show();
        }
    }
}

```

objeto de clase ContentValues permite crear un registro.
 método put(String campo, valor);

SQLiteDatabase base_de_datos.insert(String base_de_datos, null, ContentValues registro);

2) Buscar un registro en una base de datos.

Dentro de **MainActivity.java**.

```

public void buscar(View v){
    //1.- Abrir la base de datos.
    Adminbd admin = new Adminbd(this, "Productos", null, 1);
    SQLiteDatabase base = admin.getWritableDatabase();

    //2.- Obtener el valor a buscar en la tabla.
    String codigo = edcodigo.getText().toString();

    //3.- Confirmar que el valor a buscar no esté vacío.
}

```

```
if(!codigo.isEmpty()){

    //4.- Ejecutar consulta para ver el nombre y precio de un producto cuyo código coincida con el entregado.
    Cursor fila = base.rawQuery
        ("select nombre, precio from producto where codigo =" + codigo, null);
    //5.- En caso de que el código de producto sea encontrado, mostrar los valores obtenidos.
    if(fila.moveToFirst()){
        ednombre.setText(fila.getString(0));
        edprecio.setText(fila.getString(1));
        base.close();
    }
    //6.- Informar si el código no es encontrado.
    } else {
        Toast.makeText(this, "No existe el producto", Toast.LENGTH_LONG).show();
        base.close();
    }
}
//7.- Si el código no ha sido ingresado, solicitarlo.
} else {
    Toast.makeText(this, "Debes ingresar un código de producto", Toast.LENGTH_LONG).show();
}
}
```