Proyecto Android que permite gestionar una agenda de citas. Al iniciarse la aplicación deberá mostrarse una pantalla similar a esta:



* + - Opción "**Insertar Cita**":
      * Todos los campos son obligatorios.
      * Por cada cita se insertará en una tabla de una BD una fila con los campos Fecha y Hora, que serán de tipo texto.
      * El asunto se almacenará en un fichero de texto.
      * No podrán existir 2 citas con la misma fecha y hora.
      * El campo NumCita, que gestionará la aplicación directamente, será un campo entero autoincrementado comenzando por 1 (**\_id**).
    - Opción "**Borrar Cita**":
      * Los campos Fecha y Hora son obligatorios.
      * Deberá eliminarse de la tabla la fila correspondiente.
    - Opción "**Consultar Cita**":
      * Los campos Fecha y Hora son obligatorios.
      * Se mostrará el número de la cita y su asunto.
    - Opción "**Modificar Cita**":
      * Primero el usuario teclea una Fecha y una Hora y consultará la cita.
      * Posteriormente modificará el asunto y, pulsará la opción "Modificar Cita".
    - La aplicación emitirá mensajes emergentes que informan al usuario del resultado de cada una de sus acciones.

**SOLUCIÓN**

**MainActivity.java**

public class MainActivity extends AppCompatActivity {  
  
 BDHelper dbHelper;  
 TextView numeroCita;  
 EditText fecha, hora, asunto;  
 Toast info;  
 SQLiteDatabase db;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 info = Toast.*makeText*(MainActivity.this, "", Toast.*LENGTH\_SHORT*);  
 dbHelper = new BDHelper(this);  
  
 fecha = findViewById(R.id.*fecha*);  
 hora = findViewById(R.id.*hora*);  
 asunto = findViewById(R.id.*asunto*);  
 numeroCita = findViewById(R.id.*numeroCita*);  
  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 MenuInflater inflater = getMenuInflater();  
 inflater.inflate(R.menu.*menu*, menu);  
  
  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 Cita cita;  
 db = dbHelper.getWritableDatabase();  
 if (db != null) {  
 switch (item.getItemId()) {  
 case R.id.*insertar*:  
 try {  
 ContentValues cv = new ContentValues();  
 if (this.comprobarExistencia() == true) {  
 info.setText("El registro ya existe");  
 } else {  
 if (this.comprobarCampos() == true) {  
 info.setText("Campos obligatorios incompletos");  
 } else {  
 cv.put(BDSchema.Cita.*COL\_FECHA*, fecha.getText().toString());  
 cv.put(BDSchema.Cita.*COL\_HORA*, hora.getText().toString());  
 cv.put(BDSchema.Cita.*COL\_ASUNTO*, asunto.getText().toString());  
 db.insert(BDSchema.Cita.*TABLE\_NAME*, null, cv);  
 info.setText("Registro insertado");  
 this.limpiarFormato();  
 }  
 }  
  
 } catch (SQLException ex) {  
 info.setText("Algo salio mal");  
 }  
 info.show();  
 break;  
 case R.id.*borrar*:  
 cita = this.obtenerCita();  
 long newRowId = db.delete(BDSchema.Cita.*TABLE\_NAME*, BDSchema.Cita.*COL\_ID* + " like '" + cita.getId() + "'", null);  
 if (newRowId == -1) {  
 info.setText("No hay ninguna cita que borrar");  
 } else {  
 info.setText("Cita borrada con exito");  
 }  
 this.limpiarFormato();  
 info.show();  
 break;  
 case R.id.*modificar*:  
 if (this.comprobarCampos() == true) {  
 info.setText("Campos obligatorios incompletos");  
 cita = this.obtenerCita();  
 ContentValues cv = new ContentValues();  
 cv.put(BDSchema.Cita.*COL\_FECHA*, fecha.getText().toString());  
 cv.put(BDSchema.Cita.*COL\_HORA*, hora.getText().toString());  
 cv.put(BDSchema.Cita.*COL\_ASUNTO*, asunto.getText().toString());  
  
 db.update(BDSchema.Cita.*TABLE\_NAME*, cv, "id = '" + cita.getId() + "'", null);  
 info.setText("Cita modificada con exito");  
 this.limpiarFormato();  
 } else {  
 info.setText("Campos obligatorios incompletos");  
 }  
 info.show();  
 break;  
 case R.id.*consultar*:  
 if (this.comprobarExistencia() == true){  
 cita = this.obtenerCita();  
 numeroCita.setText(cita.getId());  
 asunto.setText(cita.getAsunto());  
 } else {  
 info.setText("No hay ninguna cita");  
 info.show();  
 }  
 }  
 }  
 return super.onOptionsItemSelected(item);  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 dbHelper.close();  
 }  
  
 /\*  
 Este metodo devuelve un objeto cita filtrado por la hora y fecha introducidos en la aplicacion  
 previamente introducido en la base de datos  
 \*/  
 private Cita obtenerCita() {  
 Cursor c;  
 Cita cita = new Cita();  
 c = db.rawQuery("SELECT \* FROM citas WHERE fecha like '" + fecha.getText().toString()  
 + "' AND hora like '" + hora.getText().toString() + "'", null);  
  
 while (c.moveToNext()) {  
 cita.setId(c.getString(0));  
 cita.setFehca(c.getString(1));  
 cita.setHora(c.getString(2));  
 cita.setAsunto(c.getString(3));  
  
 }  
 return cita;  
 }  
  
 private void limpiarFormato(){  
 hora.setText("");  
 fecha.setText("");  
 numeroCita.setText("NumCita");  
 asunto.setText("");  
 }  
 /\*  
 Devuelve true si existe la cita  
 \*/  
 private boolean comprobarExistencia(){  
 Cursor c;  
 c = db.rawQuery("SELECT \* FROM citas WHERE fecha like '" + fecha.getText().toString()  
 + "' AND hora like '" + hora.getText().toString() + "'", null);  
 if (c.moveToFirst()){  
 return true;  
 } else {  
 return false;  
 }  
 }  
 /\*  
 Devuelve false si los campos obligatorios no estan completos  
 \*/  
 private boolean comprobarCampos(){  
 if ("".equals(fecha.getText().toString()) || "".equals(hora.getText().toString())) {  
 return false;  
 } else {  
 return true;  
 }  
 }  
}

**Cita.java**

public class Cita {  
  
 private String id;  
 private String fehca;  
 private String hora;  
 private String asunto;  
  
 public Cita(String id, String fehca, String hora, String asunto) {  
 this.id = id;  
 this.fehca = fehca;  
 this.hora = hora;  
 this.asunto = asunto;  
 }  
 public Cita() {  
 }  
  
 public String getId() {  
 return id;  
 }  
  
 public void setId(String id) {  
 this.id = id;  
 }  
  
 public String getFehca() {  
 return fehca;  
 }  
  
 public void setFehca(String fehca) {  
 this.fehca = fehca;  
 }  
  
 public String getHora() {  
 return hora;  
 }  
  
 public void setHora(String hora) {  
 this.hora = hora;  
 }  
  
 public String getAsunto() {  
 return asunto;  
 }  
  
 public void setAsunto(String asunto) {  
 this.asunto = asunto;  
 }  
  
 @Override  
 public String toString() {  
 return "Cita{" +  
 "id='" + id + '\'' +  
 ", fehca='" + fehca + '\'' +  
 ", hora='" + hora + '\'' +  
 ", asunto='" + asunto + '\'' +  
 '}';  
 }  
}

**BDSchema.java**

public class BDSchema {  
  
 private BDSchema(){  
  
 }  
  
 public static class Cita implements BaseColumns{  
 public static final String *TABLE\_NAME* ="citas";  
 public static final String *COL\_ID* ="id";  
 public static final String *COL\_FECHA* ="fecha";  
 public static final String *COL\_HORA* ="hora";  
 public static final String *COL\_ASUNTO* ="asunto";  
 }  
  
 public static final String *SQL\_CREATE\_ENTRIES* =  
 "CREATE TABLE " + Cita.*TABLE\_NAME* + " (" +  
 Cita.*COL\_ID* + " INTEGER PRIMARY KEY AUTOINCREMENT, " +  
 Cita.*COL\_FECHA* + " DATE,"+  
 Cita.*COL\_HORA* + " DATETIME," +  
 Cita.*COL\_ASUNTO* + " TEXT)";  
 public static final String *SQL\_DELETE\_ENTRIES* = "DROP TABLE IF EXISTS " + Cita.*TABLE\_NAME*;  
}

**BDHelper.java**

public class BDHelper extends SQLiteOpenHelper {  
  
 public static final int *DATABASE\_VERSION* = 4;  
 public static final String *DATABASE\_NAME* = "citas.sqlite";  
  
 public BDHelper(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 db.execSQL(BDSchema.*SQL\_CREATE\_ENTRIES*);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 db.execSQL(BDSchema.*SQL\_DELETE\_ENTRIES*);  
 onCreate(db);  
 }  
}

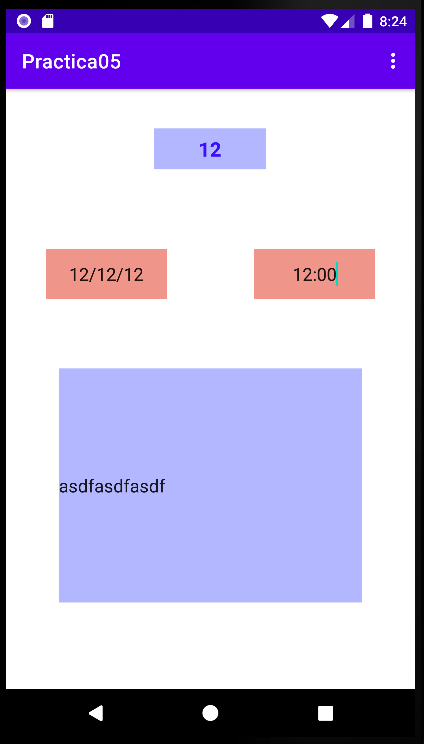
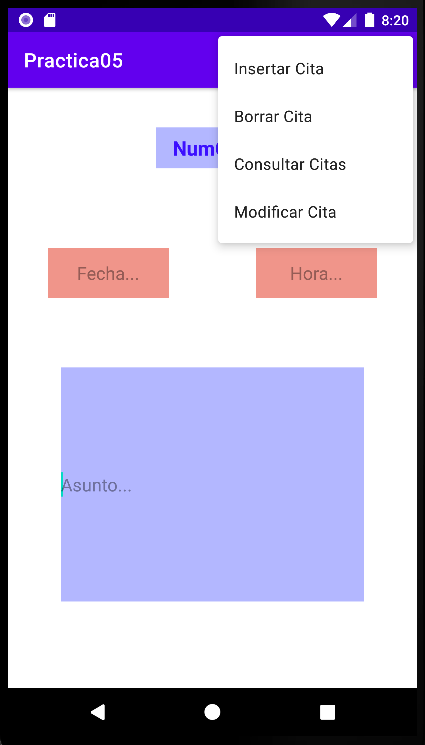
**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?><androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:layout\_width="match\_parent" android:layout\_height="match\_parent" tools:context=".MainActivity"> <TextView android:id="@+id/numeroCita" android:layout\_width="113dp" android:layout\_height="41dp" android:layout\_marginTop="40dp" android:background="@color/lightBlue" android:text="NumCita" android:textColor="@color/Blue" android:textSize="20dp" android:textStyle="bold" android:gravity="center" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.498" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" /> <EditText android:id="@+id/asunto" android:layout\_width="305dp" android:layout\_height="235dp" android:layout\_marginTop="200dp" android:ems="10" android:inputType="textPersonName" android:hint="Asunto..." android:background="@color/lightBlue" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/numeroCita" /> <EditText android:id="@+id/fecha" android:layout\_width="122dp" android:layout\_height="50dp" android:layout\_marginStart="40dp" android:layout\_marginTop="80dp" android:ems="10" android:inputType="date" android:hint="Fecha..." android:gravity="center" android:background="@color/lightOrange" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/numeroCita" /> <EditText android:id="@+id/hora" android:layout\_width="122dp" android:layout\_height="50dp" android:layout\_marginTop="80dp" android:layout\_marginEnd="40dp" android:ems="10" android:inputType="time" android:hint="Hora..." android:gravity="center" android:background="@color/lightOrange" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/numeroCita" /> <TextView android:id="@+id/info" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginTop="12dp" android:text="" android:textColor="@color/red" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.498" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/asunto" /></androidx.constraintlayout.widget.ConstraintLayout>

**menu.xml**

<?xml version="1.0" encoding="utf-8"?><menu xmlns:android="http://schemas.android.com/apk/res/android"> <item android:title="Insertar Cita" android:id="@+id/insertar"/> <item android:title="Borrar Cita" android:id="@+id/borrar"/> <item android:title="Consultar Citas" android:id="@+id/consultar"/> <item android:title="Modificar Cita" android:id="@+id/modificar"/></menu>

**Salida**

****