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

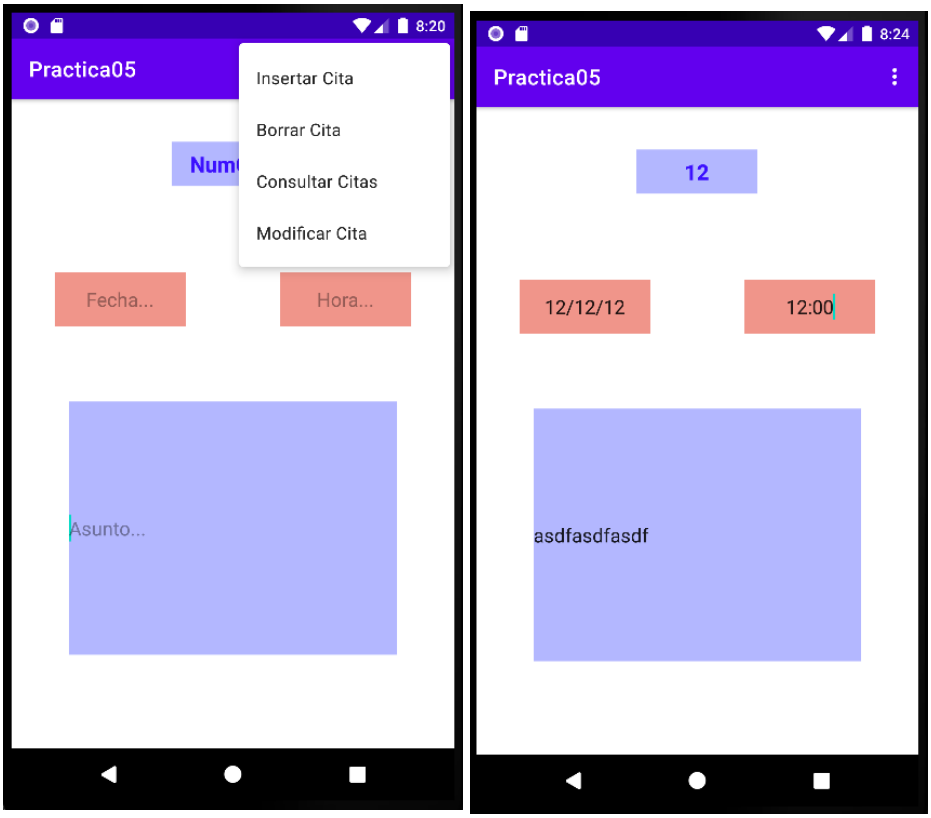
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    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


8:20
Practica05
Insertar Cita
Borrar Cita
Consultar Citas
Modificar Cita
Num
Fecha...
Hora...
Asunto...
8:24
Practica05
12
12/12/12
12:00
asdfasdfasdf