



**Universidad de El Salvador
Facultad de Ingeniería y Arquitectura
Escuela de Ingeniería en Sistemas**

Programación en Dispositivos Móviles

Guía de Laboratorio N°04 A

“Introducción al manejo de Bases de Datos con SQLite”

Version Android

Objetivo:

Que el alumno:

- 1- Implemente una base de datos a partir de su diseño Conceptual en una aplicación Android, usando el lenguaje SQLite.
- 2- Aprenda a crear una base de datos, y usar los métodos Update, Insert, Delete y Query para llevar su administración.

Descripción:

Esta práctica consistirá en realizar una aplicación que permita crear una base de datos o abrir una existente, conectarse a ella y que pueda manipularse desde la interfaz de usuario con sus respectivas Activities.

Contenido

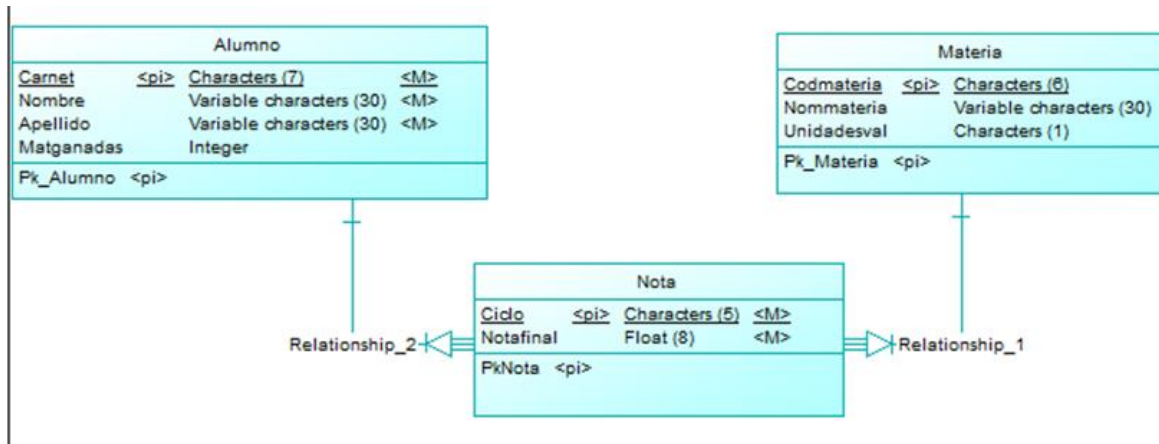
BASE DE DATOS CARNET,	1
MODELO CONCEPTUAL	1
MODELO FISICO(BASE ORACLE 11).....	1
SCRIPT DE BASE DE DATOS	1
ESTRUCTURA DE LA APLICACION	2
CREACION DE APLICACIÓN (ACTIVITIES Y OTRAS CLASES).....	3
Creación del proyecto	3
Creación del Activities	5
Creación del Clases de control de objetos.....	6
Creación del Clases controladora de Base de Datos.....	6
Estructura resultante.....	7
Creación de variables string para los layouts	8
Menú Principal	9
Submenús de tablas	10
Clases Auxiliares para Manejo de Datos.....	15
Clase Alumno.	15
Clase Materia.	16
Clase Nota.	17
CONTROLADOR DE LA BASE DE DATOS	18
Clase que controlara la base de datos.....	20
PROGRAMACION DE LOS LAYOUTS Y ACTIVITIES POR TABLA	27
Tabla ALUMNO.....	27
Método insertar (Alumno)	27
Layout activity_alumno_insertar.xml (Alumno)	27
Activity AlumnoInsertarActivity.java (Alumno).....	29
Método consultar(Alumno).....	30
Layout activity_alumno_consultar.xml (Alumno).....	30
Activity AlumnoConsultarActivity.java (Alumno).....	33
Método actualizar (Alumno)	34
Layout activity_alumno_actualizar.xml (Alumno)	34
Activity AlumnoActualizarActivity.java (Alumno)	36

Método eliminar (Alumno).....	37
Layout activity_alumno_eliminar.xml (Alumno).....	37
Activity AlumnoEliminarActivity.java (Alumno).....	38
Copiar Base de Datos inicial y ver su contenido	39
Probar opciones hechas para la tabla Alumnos desde el simulador.....	41
Ver el contenido de la tabla Alumno de nuevo	42
Tabla NOTA	42
Método insertar (Nota)	42
Layout activity_nota_insertar.xml (Nota).....	43
Activity NotaInsertarActivity.java (Nota)	44
Método consultar (Nota).....	46
Layout activity_nota_consulta.xml (Nota).....	46
Activity NotaConsultarActivity.java (Nota).....	48
Método actualizar (Nota)	49
Layout activity_nota_actualizar.xml (Nota).....	49
Activity NotaActualizarActivity.java (Nota).....	51
Método eliminar (Nota)	52
Layout activity_nota_eliminar.xml (Nota)	52
Activity EliminarNotaActivity.java (Nota)	54
Actividad final en la guía de laboratorio:.....	56
Actividad opcional en la guía de laboratorio:	56
Anexos.....	57
Anexo 1 Código de insertar materia en controlador(ControlBDCarnet).....	57
Anexo 2 Layout insertarmateria.xml	57
Anexo 3 Código de MaterialInsertarActivity,java	58
Anexo 4 Código de insertar nota en controlador (ControlBDCarnet)	59

BASE DE DATOS CARNET,

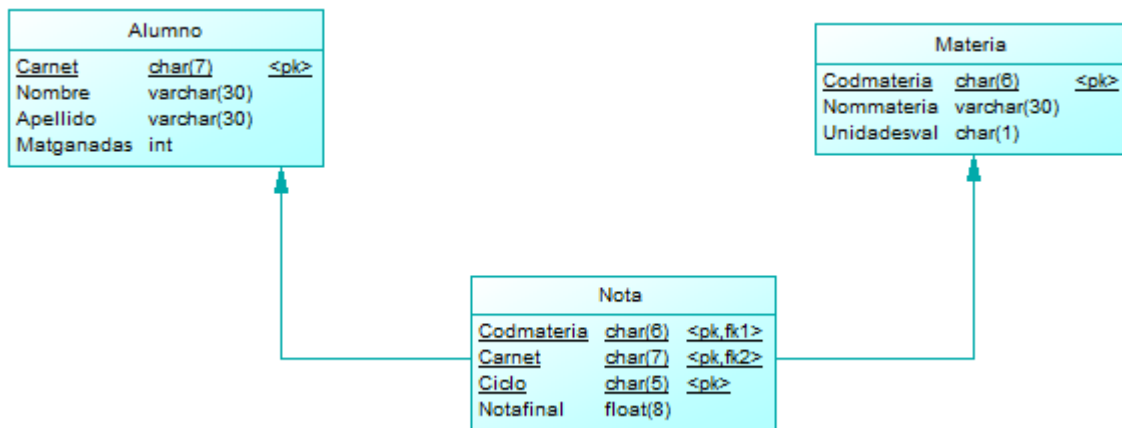
MODELO CONCEPTUAL.

Nota: revise las cardinalidades (chequear dependencia)



MODELO FISICO (BASE ORACLE 11)

Nota: Puede auxiliarse de la guía01 de Bases de Datos 2019



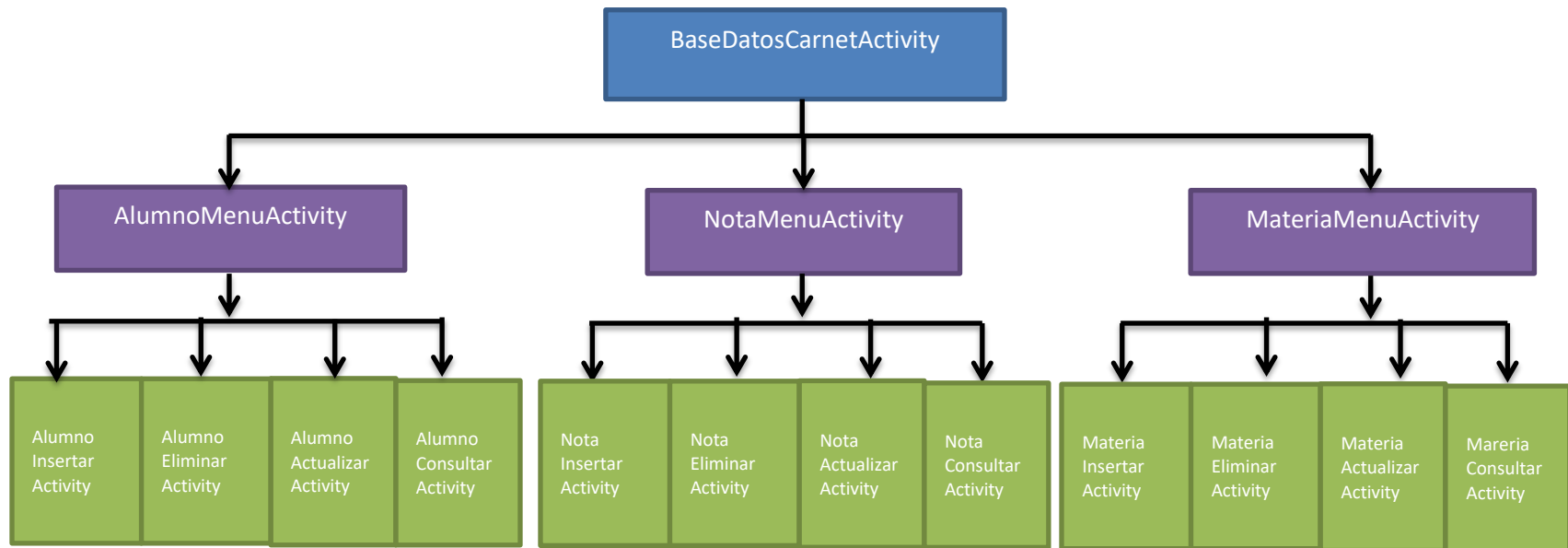
SCRIPT DE BASE DE DATOS

Nota: Buscar opción de generar base de datos (script)

ESTRUCTURA DE LA APLICACION

La aplicación que realizaremos en esta práctica, tendrá en general la siguiente estructura:

Diagrama Top Down



Ahora pasaremos a desarrollar la aplicación a medida que desarrollemos las actividades. Por tanto desarrollaremos el código en Java al mismo tiempo.

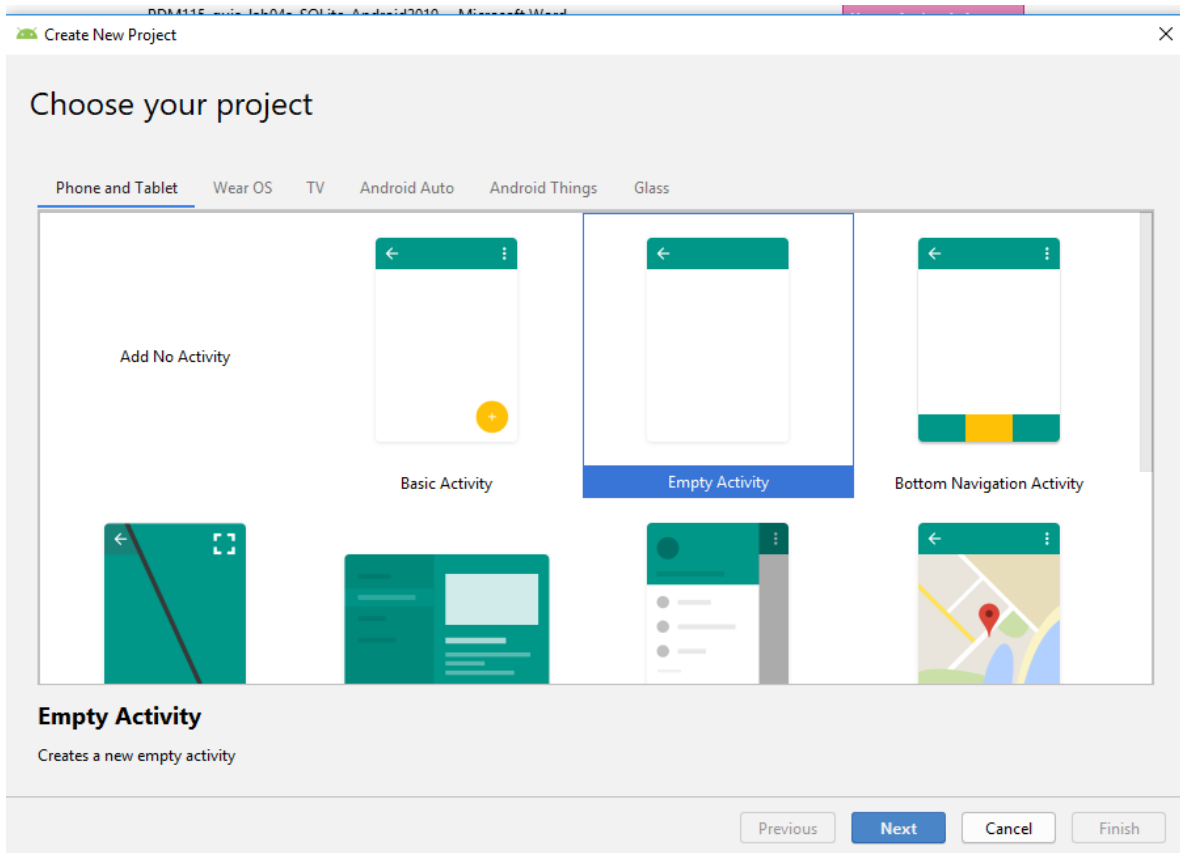
Luego, realizaremos un clase en Java que nos permita controlar cada aspecto de nuestra base de datos, antes que nada, debemos tener presente que SQLite no implementa llaves foráneas, por lo que deberemos comprobar la integridad antes de insertar, modificar o borrar datos de las tablas.

CREACION DE APLICACIÓN (ACTIVITIES Y OTRAS CLASES)

Creación del proyecto

Para empezar con nuestra aplicación, crea un nuevo proyecto con los siguientes parámetros:

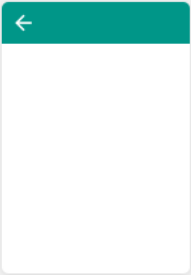
Empty Activity



Con nombre BaseDatosCarnet

Create New Project

Configure your project



Empty Activity

Creates a new empty activity

Name
BaseDatosCarnet

Package name
sv.edu.ues.fia.eisi.basedatoscarnet

Save location
C:\Users\Win10\BaseDatosCarnet

Language
Java

Minimum API level
API 16: Android 4.1 (Jelly Bean)

i Your app will run on approximately **99,6%** of devices.
[Help me choose](#)

This project will support instant apps

Previous Next Cancel **Finish**

Creación del Activities

Según el diagrama top down, crear los (Empty) Activities siguientes:

Posicione sobre el paquete **basedatoscarnet** y presione clic derecho, new, Activity, Empty Activity...

Y haga las activities que se muestran a continuación:

Para los menus

AlumnoMenuActivity.

NotaMenuActivity.

MateriaMenuActivity.

Para las opciones de Alumno

AlumnoInsertarActivity.

AlumnoEliminarActivity.

AlumnoActualizarActivity.

AlumnoConsultarActivity.

Para las opciones de Nota

NotaInsertarActivity.

NotaEliminarActivity.

NotaActualizarActivity.

NotaConsultarActivity.

Para las opciones de Materia

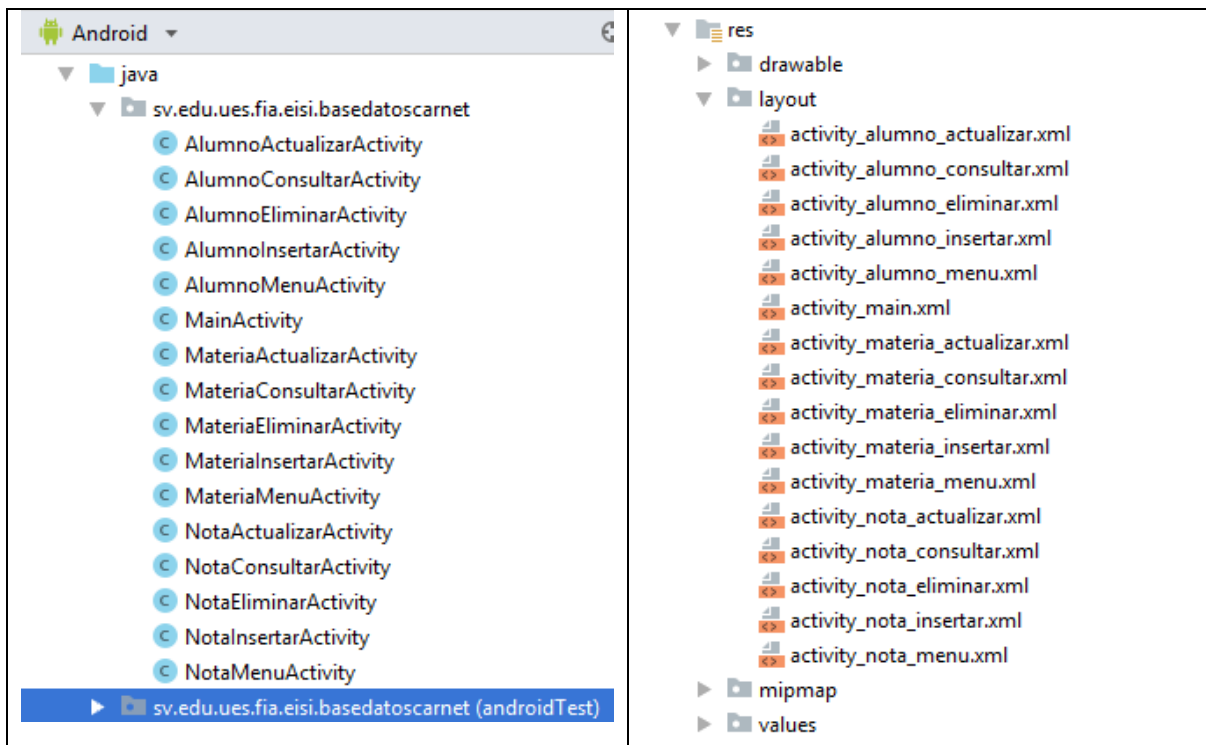
MateriaInsertarActivity.

MateriaEliminarActivity.

MateriaActualizarActivity.

MateriaConsultarActivity.

Hagalo hasta ver un árbol de proyecto como se muestra:



Creación del Clases de control de objetos

Estas clases nos servirán para manipular objetos de cada tabla mediante métodos get y set.

Crear como una clase de java (java class en el paquete anterior):

Alumno.

Nota.

Materia.

Creación del Clases controladora de Base de Datos

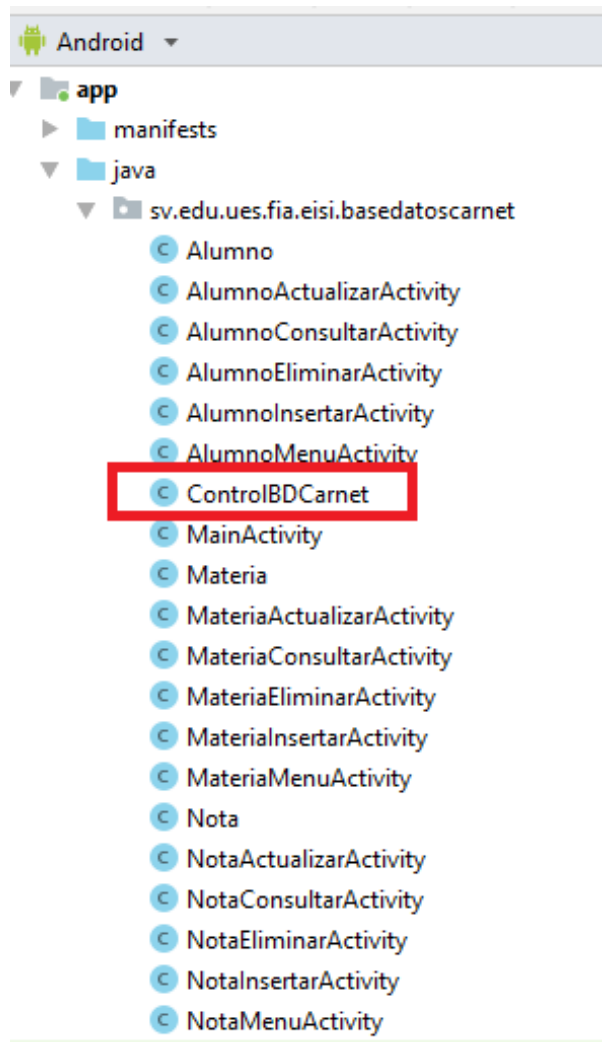
Estas clases nos servirán para manipular los eventos relacionados con la base de datos Carnet03

Crear como una clase de java :

ControlBDCarnet

Estructura resultante

Con lo anterior quedara esta estructura:



Creación de variables string para los layouts

Antes de iniciar a modificar cualquier interfaz que permitirán administrar la base de datos Carnet, agregaremos las siguientes variables al recurso *strings.xml*, de donde llamaremos en cada uno de los *layout*, los label o texto estándar que mostraran los componentes según sea necesario utilizarlos, tales como el texto que mostrara cada TextView o los diferentes Botones.

```
<resources>

    <string name="app_name">BaseDatosCarnet</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>
    <string name="title_activity_alumno_menu">AlumnoMenuActivity</string>
    <string name="title_activity_nota_menu">NotaMenuActivity</string>
    <string name="title_activity_materia_menu">MateriaMenuActivity</string>
    name="title_activity_alumno_insertar">AlumnoInsertarActivity</string>
    <string
name="title_activity_alumno_eliminar">AlumnoEliminarActivity</string>
    <string
name="title_activity_alumno_actualizar">AlumnoActualizarActivity</string>
    <string
name="title_activity_alumno_consultar">AlumnoConsultarActivity</string>
    <string name="title_activity_nota_insertar">NotaInsertarActivity</string>
    <string name="title_activity_nota_eliminar">NotaEliminarActivity</string>
    <string
name="title_activity_nota_actualizar">NotaActualizarActivity</string>
    <string name="title_activity_nota_consultar">NotaConsultarActivity</string>
    <string
name="title_activity_materia_insertar">MateriaInsertarActivity</string>
    <string
name="title_activity_materia_eliminar">MateriaEliminarActivity</string>
    <string
name="title_activity_materia_actualizar">MateriaActualizarActivity</string>
    <string
name="title_activity_materia_consultar">MateriaConsultarActivity</string>
    <string name="alumno_menu">Función a Ejecutar Alumno</string>
    <string name="nota_menu">Función a Ejecutar Nota</string>
    <string name="materia_menu">Función a Ejecutar Materia</string>
    <string name="insertar_registro">Insertar Registro</string>
    <string name="eliminar_registro">Eliminar Registro</string>
    <string name="consultar_registro">Consultar Registro</string>
    <string name="actualizar_registro">Actualizar Registro</string>
    <string name="carnet">Carnet</string>
    <string name="nombre">Nombre</string>
    <string name="apellido">Apellido</string>
    <string name="sexo">Sexo</string>
    <string name="matganadas">Materias Ganadas</string>
    <string name="insertar">Insertar</string>
    <string name="eliminar">Eliminar</string>
    <string name="actualizar">Actualizar</string>
    <string name="consultar">Consultar</string>
    <string name="ciclo">Ciclo</string>

</resources>
```

```

<string name="notafinal">Nota Final</string>
<string name="codmateria">Codigo</string>
<string name="unidadesval">Unidades Valorativas</string>
<string name="Limpiar">Limpiar</string>
<string name="codmatnota">Codigo Materia</string>
<string name="nommateria">Nombre Materia</string>
<string name="opciones">Opciones:</string>
</resources>

```

Menú Principal

Realizaremos un menú, pero no usaremos un recurso **layout** para nuestra interfaz, implementaremos un **extends** de la clase **ListActivity** que nos facilitara la creación de un **ListView** programáticamente.

Modifica el archivo: *MainActivity.java* con el siguiente código.

```

package sv.ues.fia.basedatoscarnet;

import android.app.ListActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class MainActivity extends ListActivity {

    String[] menu={"Tabla Alumno","Tabla Nota","Tabla Materia","LLenar Base
de Datos"};
    String[]
activities={"AlumnoMenuActivity","NotaMenuActivity","MateriaMenuActivity"};

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setListAdapter(new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, menu));
    }

    @Override
    protected void onItemClick(ListView l,View v,int position,long id){
        super.onItemClick(l, v, position, id);

        if(position!=3){

            String nombreValue=activities[position];

            try{
                Class<?>
clase=Class.forName("sv.ues.fia.basedatoscarnet."+nombreValue);
                Intent inte = new Intent(this,clase);
                this.startActivity(inte);
            }

```

```

        }catch(ClassNotFoundException e){
            e.printStackTrace();
        }

        }else{

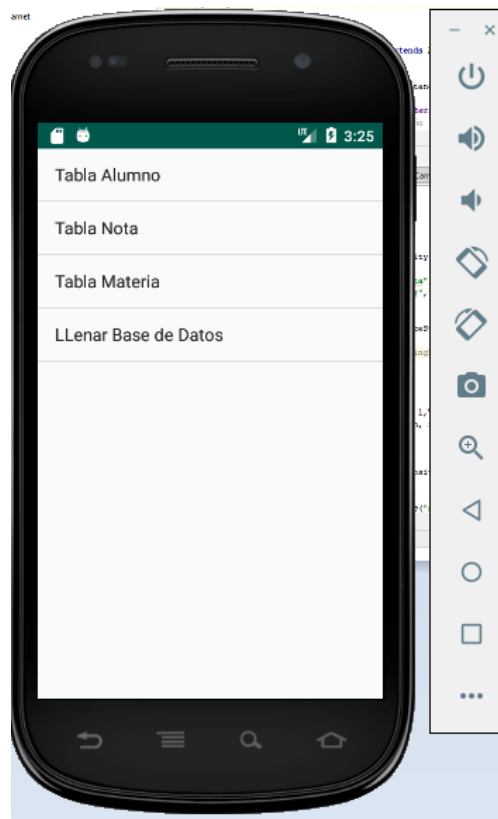
            //CODIGO PARA LLENAR BASE DE DATOS

        }
    }
}

```

OBSERVA que al seleccionar en el menú **Llenar Base de Datos**, ubicado en la **position** número 3 (ya que el índice de un arreglo inicia en cero) se deberá ejecutar cierto código que permita llenar la base de datos **carnet.s3ql**. Dejaremos esto así y regresaremos a modificar **MainActivity.java** más adelante en la práctica.

Ejecuta la aplicación, deberías ver algo como esto:



Submenús de tablas

Hasta aquí podemos seleccionar la tabla que queremos utilizar, sin embargo necesitaremos un submenú que muestre la operación a realizar sobre la tabla seleccionada (insertar, eliminar, consultar o modificar datos). Para ello crearemos un submenú para cada tabla de esta manera

Sub menú Alumno: Modificamos una nueva clase dentro del paquete **sv.ues.fia.basedatoscarnet** llamada **AlumnoMenuActivity.java**, esta deberá contener el siguiente código.


```

package sv.ues.fia.basedatoscarnet;

import android.app.ListActivity;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;

public class NotaMenuActivity extends ListActivity {

    String[] menu={"Insertar Registro","Eliminar Registro","Consultar Registro",
"Actualizar Registro"};
    String[]
activities={"NotaInsertarActivity","NotaEliminarActivity","NotaConsultarActivity",
"NotaActualizarActivity"};

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setListAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,
menu));

        ListView listView = getListView();
        listView.setBackgroundColor(Color.rgb(64, 0, 128));

        ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1, menu);
        setListAdapter(adapter);
    }

    @Override
    protected void onItemClick(ListView l,View v,int position,long id){
        super.onItemClick(l, v, position, id);

        String nombreValue=activities[position];


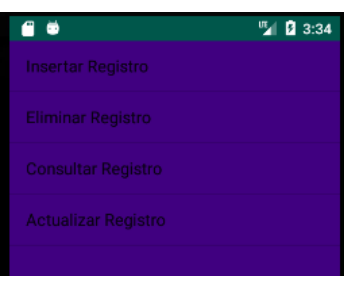

        l.getChildAt(position).setBackgroundColor(Color.rgb(255, 128, 0));

        try{
            Class<?> clase=Class.forName("sv.ues.fia.basedatoscarnet."+nombreValue);
            Intent inte = new Intent(this,clase);
            this.startActivity(inte);
        }catch(ClassNotFoundException e){
            e.printStackTrace();
        }
    }
}

```

Sub menú Materia: Modifica la clase dentro del paquete sv.ues.fia.basedatoscarnet llamada **MateriaMenuActivity.java**, esta deberá contener el siguiente código.

Ahora cada submenú se visualizará así:

AlumnoMenuActivity	NotaMenuActivity	MateriaMenuActivity
		

Nota:

Si no aparecen estos submenús... revisa en las clases la siguiente línea de código....

```
Class<?>
clase=Class.forName("sv.edu.ues.fia.eisi.basedatoscarnet."+nombreValue);
```

Y verifica que el nombre entrecomillado sea el de tu paquete, sino... **cámbialo** y vuelve a ejecutarlo.

Clases Auxiliares para Manejo de Datos

Para simplificar y entender de mejor manera la lógica de nuestra aplicación haremos uso de 3 clases que encapsulan los datos que se almacenaran en las tablas. Estas son la clase *Alumno*, *Materia* y *Nota*, nótese que corresponden a las tres tablas de nuestra base de datos.

El fin de estas clases es guardar y obtener los valores de una fila de la tabla, para hacer el uso adecuado de ellos.

Dentro de nuestro paquete crear las 3 clases con el siguiente código:

Clase Alumno.

```
package sv.ues.fia.basedatoscarnet;
public class Alumno {

    private String carnet;
    private String nombre;
    private String apellido;
    private String sexo;
    private int matganadas;

    public Alumno(){
    }

    public Alumno(String carnet, String nombre, String apellido, String sexo) {
        this.carnet = carnet;
        this.nombre = nombre;
        this.apellido = apellido;
        this.sexo = sexo;
    }

    public String getCarnet() {
        return carnet;
    }

    public void setCarnet(String carnet) {
        this.carnet = carnet;
    }

    public String getNombre() {
        return nombre;
    }

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

    public String getApellido() {
        return apellido;
    }

    public void setApellido(String apellido) {
        this.apellido = apellido;
    }
}
```

```

    }

    public String getSexo() {
        return sexo;
    }

    public void setSexo(String sexo) {
        this.sexo = sexo;
    }

    public int getMatganadas() {
        return matganadas;
    }

    public void setMatganadas(int matganadas) {
        this.matganadas = matganadas;
    }
}

```

Clase Materia.

```

package sv.ues.fia.basedatoscarnet;

public class Materia {

    private String codmateria;
    private String nommateria;
    private String unidadesval;

    public Materia(){

    }

    public Materia(String codmateria, String nommateria, String unidadesval) {
        this.codmateria = codmateria;
        this.nommateria = nommateria;
        this.unidadesval = unidadesval;
    }

    public String getCodmateria() {
        return codmateria;
    }

    public void setCodmateria(String codmateria) {
        this.codmateria = codmateria;
    }

    public String getNommateria() {
        return nommateria;
    }

    public void setNommateria(String nommateria) {
        this.nommateria = nommateria;
    }
}

```

```

    public String getUnidadesval() {
        return unidadesval;
    }

    public void setUnidadesval(String unidadesval) {
        this.unidadesval = unidadesval;
    }
}

```

Clase Nota.

```

package sv.ues.fia.basedatoscarnet;

public class Nota {

    private String carnet;
    private String codmateria;
    private String ciclo;
    private float notafinal;

    public Nota(){

    }

    public Nota(String carnet, String codmateria, String ciclo, float
notafinal) {
        this.carnet = carnet;
        this.codmateria = codmateria;
        this.ciclo = ciclo;
        this.notafinal = notafinal;
    }

    public String getCarnet() {
        return carnet;
    }

    public void setCarnet(String carnet) {
        this.carnet = carnet;
    }

    public String getCodmateria() {
        return codmateria;
    }

    public void setCodmateria(String codmateria) {
        this.codmateria = codmateria;
    }

    public String getCiclo() {
        return ciclo;
    }

    public void setCiclo(String ciclo) {

```

```

        this.ciclo = ciclo;
    }

    public float getNotafinal() {
        return notafinal;
    }

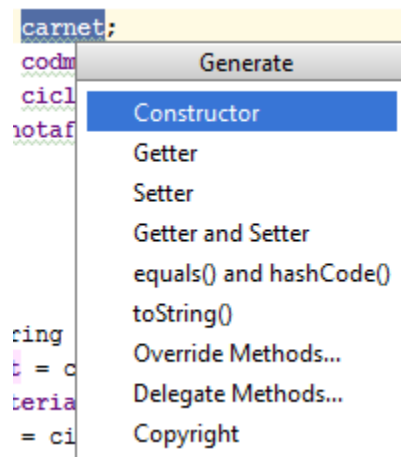
    public void setNotafinal(float notafinal) {
        this.notafinal = notafinal;
    }
}

```

Observaciones

El *IDE AndroidStudio* cuenta con la opción de insertar código que es de uso común, por ejemplo constructores y los métodos get y set, para ello inicialmente DEBEMOS ESPECIFICAR LOS MIEMBROS O ATRIBUTOS DE NUESTRA CLASE y luego ir al clic derecho, generate (o ALT+Insert)

Y se mostrara un cuadro como el siguiente:



CONTROLADOR DE LA BASE DE DATOS

Esta será la estructura de la clase que controlará nuestra base de datos en la aplicación.

```

public class NuestraClaseQueControlaLaBD {

    Aquí declarar todas las variables que usamos

    private final Context context; //←Almacenara el context de nuestra Activity
    private DatabaseHelper DBHelper; //←Nuestro Auxiliador de BD
    private SQLiteDatabase db; //←Instancia de nuestra BD

    public NuestraClaseQueControlaLaBD (Context ctx) { //←Constructor
        this.context = ctx;
    }
}

```

```

    DBHelper = new DatabaseHelper(context);
}

private static class DatabaseHelper extends SQLiteOpenHelper {
    DatabaseHelper(Context context) {
        super(context, "NOMBRE_BASEDATOS", null, No_VERSION_BASEDATOS);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        try {
            db.execSQL("Comando1SQLpara CREAM_TABLA1");
            db.execSQL("Comando2SQLpara CREAM_TABLA2 o LLenarla");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        Aquí nuestra lógica al actualizar nuestra base de datos
    }
}

// ---abrir la base de datos---
public void abrir() throws SQLException {
    db = DBHelper.getWritableDatabase(); // ←Abrir BD con permisos R/W
    return;
}

// ---cerrar la base de datos---
public void cerrar() {
    DBHelper.close();
}
}

Aquí todos los métodos de inserción de datos a la base, de consulta, borrado, actualización y otros.

```

OJO: el método `execSQL()` solo permita ejecutar un comando SQL a la vez.

La clase `SQLiteDatabase` permite crear, borrar, ejecutar comandos SQL, y realizar otras tareas comunes para el manejo de bases de datos.

Ahora bien, la clase `SQLiteOpenHelper` es una clase auxiliar para la creación de la base de datos y la gestión de las versiones por las que posiblemente cambiará la base de datos.

Al hacer una clase **EXTENDS** de `SQLiteOpenHelper` unos de los métodos que se heredan, y los cuales son los más comunes a redefinir a nuestra conveniencia serán `onCreate()` y `onUpgrade()`

IMPORTANTE: `NOMBRE_BASEDATOS` es el nombre del archivo de nuestra base de datos, por lo que este parámetro es utilizado por el constructor de la clase `SQLiteOpenHelper` de manera que, si no encuentra la base de datos con el nombre `NOMBRE_BASEDATOS` especificado, convoca al método `onCreate()`, el cual redefinimos con `@Override` para que además de crear la base de datos, cree las tablas que necesitamos.

DatabaseHelper solo nos ayuda, a definir nuestra lógica al crear la base de datos, a definir rutinas para crear las tablas si es primera vez que se crea, y a auxiliarnos para llevar el control de las versiones de la misma.

Clase que controlara la base de datos

Modifica la clase `ControlBDCarnet.java` dentro del paquete `sv.ues.fia.basedatoscarnet`, copiale este código.

Nota: que hemos aplicado la estructura que se explico al inicio de esta practica.

```
package sv.ues.fia.basedatoscarnet;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class ControlBDCarnet{

    private static final String[]camposAlumno = new String []
{"carnet","nombre","apellido","sexo", "matganadas"};
    private static final String[]camposNota = new String []
{"carnet","codmateria","ciclo","notafinal"};
    private static final String[] camposMateria = new String []
{"codmateria","nommateria","unidadesval"};

    private final Context context;
    private DatabaseHelper DBHelper;
    private SQLiteDatabase db;

    public ControlBDCarnet(Context ctx) {
        this.context = ctx;
        DBHelper = new DatabaseHelper(context);
    }
}
```

```

}

private static class DatabaseHelper extends SQLiteOpenHelper {

    private static final String BASE_DATOS = "alumno.s3db";
    private static final int VERSION = 1;

    public DatabaseHelper(Context context) {
        super(context, BASE_DATOS, null, VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        try{
            db.execSQL("CREATE TABLE alumno(carnet VARCHAR(7) NOT NULL PRIMARY
KEY,nombre VARCHAR(30),apellido VARCHAR(30),sexo VARCHAR(1),matganadas INTEGER);");
            db.execSQL("CREATE TABLE materia(codmateria VARCHAR(6) NOT NULL
PRIMARY KEY,nommateria VARCHAR(30),unidadesval VARCHAR(1));");
            db.execSQL("CREATE TABLE nota(carnet VARCHAR(7) NOT NULL ,codmateria
VARCHAR(6) NOT NULL ,ciclo VARCHAR(5) ,notafinal REAL ,PRIMARY
KEY(carnet,codmateria,ciclo));");

        }catch(SQLException e){

            e.printStackTrace();

        }
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // TODO Auto-generated method stub

    }

}

public void abrir() throws SQLException{
    db = DBHelper.getWritableDatabase();
    return;
}

public void cerrar(){
    DBHelper.close();
}

public String insertar(Alumno alumno){

    return null;
}

public String insertar(Nota nota){

```



```
        return null;
    }

    public String insertar(Materia materia){
        return null;
    }

    public String actualizar(Alumno alumno){
        return null;
    }

    public String actualizar(Materia materia){
        return null;
    }

    public String actualizar(Nota nota){
        return null;
    }

    public String eliminar(Alumno alumno){
        return null;
    }

    public String eliminar(Nota nota){
        return null;
    }

    public Alumno consultarAlumno(String carnet){
        return null;
    }

    public Materia consultarMateria(String codmateria){
        return null;
    }

    public Nota consultarNota(String carnet, String codmateria, String ciclo){
        return null;
    }

    private boolean verificarIntegridad(Object dato, int relacion) throws SQLException{
        switch(relacion){
```

```

        case 1:
        {
            //verificar que al insertar nota exista carnet del alumno y el
            codigo de materia
            Nota nota = (Nota)dato;
            String[] id1 = {nota.getCarnet()};
            String[] id2 = {nota.getCodmateria()};
            //abrir();
            Cursor cursor1 = db.query("alumno", null, "carnet = ?", id1, null,
            null, null);
            Cursor cursor2 = db.query("materia", null, "codmateria = ?", id2,
            null, null, null);
            if(cursor1.moveToFirst() && cursor2.moveToFirst()){
                //Se encontraron datos
                return true;
            }
            return false;
        }

        case 2:
        {
            //verificar que al modificar nota exista carnet del alumno, el
            codigo de materia y el ciclo
            Nota nota1 = (Nota)dato;
            String[] ids = {nota1.getCarnet(), nota1.getCodmateria(),
            nota1.getCiclo()};
            abrir();
            Cursor c = db.query("nota", null, "carnet = ? AND codmateria = ? AND
            ciclo = ?", ids, null, null, null);
            if(c.moveToFirst()){
                //Se encontraron datos
                return true;
            }
            return false;
        }

        case 3:
        {
            Alumno alumno = (Alumno)dato;
            Cursor c=db.query(true, "nota", new String[] {
                "carnet" }, "carnet='"+alumno.getCarnet()+"'",null,
            null, null, null, null);
            if(c.moveToFirst())
                return true;
            else
                return false;
        }

        case 4:
        {
            Materia materia = (Materia)dato;
            Cursor cmat=db.query(true, "nota", new String[] {
                "codmateria" },
            "codmateria='"+materia.getCodmateria()+"'",null, null, null, null, null);
            if(cmat.moveToFirst())

```

```

        return true;
    else
        return false;
    }

    case 5:
    {
        //verificar que exista alumno
        Alumno alumno2 = (Alumno)dato;
        String[] id = {alumno2.getCarnet()};
        abrir();
        Cursor c2 = db.query("alumno", null, "carnet = ?", id, null, null,
null);

        if(c2.moveToFirst()){
            //Se encontro Alumno
            return true;
        }
        return false;
    }

    case 6:
    {
        //verificar que exista Materia
        Materia materia2 = (Materia)dato;
        String[] idm = {materia2.getCodmateria()};
        abrir();
        Cursor cm = db.query("materia", null, "codmateria = ?", idm, null,
null, null);

        if(cm.moveToFirst()){
            //Se encontro Materia
            return true;
        }
        return false;
    }

    default:
        return false;
    }

}

}

public String llenarBDCarnet(){

    final String[] VAcarnet = {"0012035", "OF12044", "GG11098", "CC12021"};
    final String[] VANombre = {"Carlos", "Pedro", "Sara", "Gabriela"};
    final String[] VAApellido = {"Orantes", "Ortiz", "Gonzales", "Coto"};
    final String[] VAsexo = {"M", "M", "F", "F"};

    final String[] VMcodmateria = {"MAT115", "PRN115", "IEC115", "TSI115"};
    final String[] VMnommateria = {"Matematica I", "Programacion I", "Ingenieria
Economica", "Teoria de Sistemas"};
    final String[] VMunidadesval = {"4", "4", "4", "4"};

```

```

        final String[] VNcarnet =
{"0012035", "OF12044", "GG11098", "CC12021", "0012035", "GG11098", "OF12044"};
        final String[] VNcodmateria =
{"MAT115", "PRN115", "IEC115", "TSI115", "IC115", "MAT115", "PRN115"};
        final String[] VNciclo =
{"12016", "12016", "22016", "22016", "22016", "12016", "22016"};
        final float[] VNnotafinal = {7,5,8,7,6,10,7};
        abrir();
        db.execSQL("DELETE FROM alumno");
        db.execSQL("DELETE FROM materia");
        db.execSQL("DELETE FROM nota");

        Alumno alumno = new Alumno();
        for(int i=0;i<4;i++){
            alumno.setCarnet(VAcarnet[i]);
            alumno.setNombre(VAnombre[i]);
            alumno.setApellido(VAapellido[i]);
            alumno.setSexo(VAsexo[i]);
            alumno.setMatganadas(0);
            insertar(alumno);
        }

        Materia materia = new Materia();
        for(int i=0;i<4;i++){
            materia.setCodmateria(VMcodmateria[i]);
            materia.setNommmateria(VMnommmateria[i]);
            materia.setUnidadesval(VMunidadesval[i]);
            insertar(materia);
        }

        Nota nota = new Nota();
        for(int i=0;i<7;i++){
            nota.setCarnet(VNcarnet[i]);
            nota.setCodmateria(VNcodmateria[i]);
            nota.setCiclo(VNciclo[i]);
            nota.setNotafinal(VNnotafinal[i]);
            insertar(nota);
        }

        cerrar();
        return "Guardo Correctamente";
    }
}

```

Observación:

Todo lo que respecta al método **verificarIntegridad** se refiere a verificar la Integridad de los datos dentro de la base de datos de manera programática, ya que dentro de la base de datos automáticamente no se puede realizar por lo cual es necesario cumplir con la leyes de integridad y de esta manera los datos insertados pueden ser considerados válidos. Hasta este momento existen 12 métodos que no hemos implementado (insertar, eliminar, consultar, actualizar), estos son los que modificaremos a medida avancemos en la práctica.

Ahora si podremos implementar la opción **Llenar Base de Datos** del menú principal. Modifica la clase MainActivity.java debe ser de la siguiente manera:

```

package sv.ues.fia.basedatoscarnet;

import android.app.ListActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
public class MainActivity extends ListActivity {

    String[] menu={"Tabla Alumno","Tabla Nota","Tabla Materia","Llenar Base de Datos"};
    String[] activities={"AlumnoMenuActivity","NotaMenuActivity","MateriaMenuActivity"};
    ControlBDCarnet BDhelper;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setListAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,
menu));
        BDhelper=new ControlBDCarnet(this);
    }

    @Override
    protected void onItemClick(ListView l,View v,int position,long id){
        super.onItemClick(l, v, position, id);
        if(position!=3){

            String nombreValue=activities[position];

            try{
                Class<?> clase=Class.forName("sv.ues.fia.basedatoscarnet."+nombreValue);
                Intent inte = new Intent(this,clase);
                this.startActivity(inte);
            }catch(ClassNotFoundException e){
                e.printStackTrace();
            }

        }else{

            BDhelper.abrir();
            String tost=BDhelper.llenarBDCarnet();
            BDhelper.cerrar();
            Toast.makeText(this, tost, Toast.LENGTH_SHORT).show();

        }
    }
}

```

PROGRAMACION DE LOS LAYOUTS Y ACTIVITIES POR TABLA

Tabla ALUMNO

Método insertar (Alumno)

En esta parte veremos como insertar algún alumno de nuestra base de datos y para ello modificamos el método que se presenta a continuación de nuestra clase ControlDBCarnet.java

```

public String insertar(Alumno alumno){

    String regInsertados="Registro Insertado N°= ";
    long contador=0;

    ContentValues alum = new ContentValues();
    alum.put("carnet", alumno.getCarnet());
    alum.put("nombre", alumno.getNombre());
    alum.put("apellido", alumno.getApellido());
    alum.put("sexo", alumno.getSexo());
    alum.put("matganadas", alumno.getMatganadas());
    contador=db.insert("alumno", null, alum);

    if(contador==-1 || contador==0)
    {
        regInsertados= "Error al Insertar el registro, Registro
                        Duplicado. Verificar inserción";
    }
    else {
        regInsertados=regInsertados+contador;
    }
    return regInsertados;
}

```

Observación:

La variable contador es la que nos permite verificar si el registro se ha insertado; ya que al realizar la operación **db.insert ()** regresa el valor del registro ingresado es decir la nueva fila que ha sido ingresada a la base de datos.

Layout activity_alumno_insertar xml (Alumno)

El siguiente paso es modificar la interfaz de usuario con el nombre *activity_alumno_insertar xml* para mostrar los datos de la tabla Alumno.

```

<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"

```

```

android:paddingTop="30dip"
android:stretchColumns="1" >

<TableRow>
    <TextView android:text="@string/carnet"/>
    <EditText
        android:id="@+id/editCarnet"
        android:inputType="text"
        android:maxLength="7" />
</TableRow>

<TableRow>
    <TextView android:text="@string/nombre" />
    <EditText
        android:id="@+id/editNombre"
        android:inputType="text"
        android:maxLength="30" />
</TableRow>

<TableRow>
    <TextView android:text="@string/apellido" />
    <EditText
        android:id="@+id/editApellido"
        android:inputType="text"
        android:maxLength="30" />
</TableRow>

<TableRow>
    <TextView android:text="@string/sexo" />
    <EditText
        android:id="@+id/editSexo"
        android:inputType="text"
        android:maxLength="1" />
</TableRow>

    <TableRow>
        <TextView android:text="@string/opciones" />
    </TableRow>

<TableRow>
    <Button
        android:onClick="insertarAlumno"
        android:text="@string/insertar" />
    <Button
        android:onClick="limpiarTexto"
        android:text="@string/limpiar" />
</TableRow>
</TableLayout>

```

Activity AlumnoInsertarActivity.java (Alumno)

Luego modificaremos la clase llamada *AlumnoInsertarActivity.java* dentro de nuestro paquete, y agregamos el siguiente código:

```

package sv.ues.fia.basedatoscarnet;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class AlumnoInsertarActivity extends Activity {

    ControlBDCarnet helper;
    EditText editCarnet;
    EditText editNombre;
    EditText editApellido;
    EditText editSexo;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_alumno_insertar);
        helper = new ControlBDCarnet(this);
        editCarnet = (EditText) findViewById(R.id.editCarnet);
        editNombre = (EditText) findViewById(R.id.editNombre);
        editApellido = (EditText) findViewById(R.id.editApellido);
        editSexo = (EditText) findViewById(R.id.editSexo);
    }
    public void insertarAlumno(View v) {
        String carnet=editCarnet.getText().toString();
        String nombre=editNombre.getText().toString();
        String apellido=editApellido.getText().toString();
        String sexo=editSexo.getText().toString();
        String regInsertados;
        Alumno alumno=new Alumno();
        alumno.setCarnet(carnet);
        alumno.setNombre(nombre);
        alumno.setApellido(apellido);
        alumno.setSexo(sexo);
        alumno.setMatganadas(0);
        helper.abrir();
        regInsertados=helper.insertar(alumno);
        helper.cerrar();
        Toast.makeText(this, regInsertados, Toast.LENGTH_SHORT).show();
    }

    public void limpiarTexto(View v) {

        editCarnet.setText("");
        editNombre.setText("");
        editApellido.setText("");
        editSexo.setText("");
    }
}

```


Observaciones:

La variable **regInsertados** es la que almacena la posición de la nueva fila insertada dentro de la base de datos es decir el nuevo registro junto con el String mensaje, el cual menciona si el registro es insertado correctamente o no y posteriormente es mostrado en el Toast. Esta misma secuencia es la que se coloca tanto para insertar nota como para realizar la inserción de materia.

Método consultar(Alumno)

En esta parte veremos como consultar algún alumno de nuestra base de datos y para ello agregaremos el siguiente código en el método correspondiente en la clase ControlIDBCarnet.java

```
public Alumno consultarAlumno(String carnet){
    String[] id = {carnet};
    Cursor cursor = db.query("alumno", camposAlumno, "carnet = ?", id,
null, null, null);
    if(cursor.moveToFirst()){
        Alumno alumno = new Alumno();
        alumno.setCarnet(cursor.getString(0));
        alumno.setNombre(cursor.getString(1));
        alumno.setApellido(cursor.getString(2));
        alumno.setSexo(cursor.getString(3));
        alumno.setMatganadas(cursor.getInt(4));
        return alumno;
    }else{
        return null;
    }
}
```

Observaciones:

Para consultar algún alumno el único parámetro q necesitamos mandarle es su respectivo carnet, para ello utilizamos una variable de tipo Cursor la cual nos ayudara a extraer cada uno de los campos del objeto Alumno.

Layout activity_alumno_consultar.xml (Alumno)

Lo siguiente será modificar el archivo `activity_alumno_consultar.xml` dentro de la carpeta **Layout** y agregamos el siguiente código:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
```

```
android:layout_height="fill_parent"
android:orientation="vertical">

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/carnet"/>

<EditText
    android:id="@+id/editCarnet"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:inputType="text"
    android:maxLength="7"/>

<LinearLayout
    android:id="@+id/LinearLayout1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" >

    <Button
        android:id="@+id/botonConsultar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/consultar"
        android:onClick="consultarAlumno"/>

    <Button
        android:id="@+id/botonLimpiar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/limpiar"
        android:onClick="limpiarTexto"/>

</LinearLayout>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/nombre" />

<EditText
    android:id="@+id/editNombre"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="text"
    android:enabled = "false"
    android:focusable="false"
        android:focusableInTouchMode="false"
        android:cursorVisible="false"
        android:longClickable="false" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/apellido" />
```

```
<EditText
    android:id="@+id/editApellido"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="text"
    android:enabled="false"
    android:focusable="false"
        android:focusableInTouchMode="false"
        android:cursorVisible="false"
        android:longClickable="false"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/sexo" />

<EditText
    android:id="@+id/editSexo"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:inputType="text"
    android:enabled="false"
    android:focusable="false"
        android:focusableInTouchMode="false"
        android:cursorVisible="false"
        android:longClickable="false" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/matganadas" />

<EditText
    android:id="@+id/editMatganadas"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:inputType="text"
    android:enabled="false"
    android:focusable="false"
        android:focusableInTouchMode="false"
        android:cursorVisible="false"
        android:longClickable="false" />

</LinearLayout>
```

Activity AlumnoConsultarActivity.java (Alumno)

Ahora modificamos la clase llamada *AlumnoConsultarActivity.java* dentro de nuestro paquete y le agregamos el siguiente código:

```

package sv.ues.fia.basedatoscarnet;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class AlumnoConsultarActivity extends Activity {

    ControlBDCarnet helper;
    EditText editCarnet;
    EditText editNombre;
    EditText editApellido;
    EditText editSexo;
    EditText editMatganadas;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_alumno_consultar);
        helper = new ControlBDCarnet(this);

        editCarnet = (EditText) findViewById(R.id.editCarnet);
        editNombre = (EditText) findViewById(R.id.editNombre);
        editApellido = (EditText) findViewById(R.id.editApellido);
        editSexo = (EditText) findViewById(R.id.editSexo);
        editMatganadas = (EditText) findViewById(R.id.editMatganadas);
    }

    public void consultarAlumno(View v) {

        helper.abrir();
        Alumno alumno =
helper.consultarAlumno(editCarnet.getText().toString());

        helper.cerrar();
        if(alumno == null)
            Toast.makeText(this, "Alumno con carnet " +
editCarnet.getText().toString() +
                " no encontrado", Toast.LENGTH_LONG).show();
        else{
            editNombre.setText(alumno.getNombre());
            editApellido.setText(alumno.getApellido());
            editSexo.setText(alumno.getSexo());

editMatganadas.setText(String.valueOf(alumno.getMatganadas()));
        }
    }
}

```

```

    public void limpiarTexto(View v){
        editCarnet.setText("");
        editNombre.setText("");
        editApellido.setText("");
        editSexo.setText("");
        editMatganadas.setText("");
    }
}

```

Método actualizar (Alumno)

En esta sección veremos como modificar o actualizar datos almacenados en nuestra tabla Alumno, para ello agregamos el siguiente código en el método correspondiente a nuestra clase ControlDBCarnet.java

```

public String actualizar(Alumno alumno){

    if(verificarIntegridad(alumno, 5)){
        String[] id = {alumno.getCarnet()};
        ContentValues cv = new ContentValues();
        cv.put("nombre", alumno.getNombre());
        cv.put("apellido", alumno.getApellido());
        cv.put("sexo", alumno.getSexo());
        db.update("alumno", cv, "carnet = ?", id);
        return "Registro Actualizado Correctamente";
    }else{
        return "Registro con carnet " + alumno.getCarnet() + " no
existe";
    }
}

```

Observaciones:

Para modificar datos de la basa hacemos uso del método **update(nombre_tabla_a_modificar, valores_a_modificar, sentencia_where, parametros_sentencia_where)**, el uso del signo (?) dentro de la sentencia *where* especifica que aquí se colocara un valor de los que se establezcan en los parámetros del *where*.

Layout activity_alumno_actualizar.xml (Alumno)

El siguiente paso ahora será crear el archivo *activity_alumno_actualizar.xml* dentro de la carpeta **Layout** de nuestro proyecto, la cual contendrá el siguiente código:

```

<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:paddingTop="10dip"
    android:stretchColumns="1" >

```

```
<TableRow>

    <TextView android:text="@string/carnet" />

    <EditText
        android:id="@+id/editCarnet"
        android:inputType="text"
        android:maxLength="7" />
</TableRow>

<TableRow>

    <TextView android:text="@string/nombre" />

    <EditText
        android:id="@+id/editNombre"
        android:inputType="text"
        android:maxLength="30" />
</TableRow>

<TableRow>

    <TextView android:text="@string/apellido" />

    <EditText
        android:id="@+id/editApellido"
        android:inputType="text"
        android:maxLength="30" />
</TableRow>

<TableRow>

    <TextView android:text="@string/sexo" />

    <EditText
        android:id="@+id/editSexo"
        android:inputType="text"
        android:maxLength="1" />
</TableRow>

<TableRow>

    <Button
        android:onClick="actualizarAlumno"
        android:text="@string/actualizar" />
</TableRow>

<TableRow>

    <Button
        android:onClick="limpiarTexto"
        android:text="@string/limpiar" />
</TableRow>

</TableLayout>
```

Activity AlumnoActualizarActivity.java (Alumno)

Luego modificamos la clase llamada *AlumnoActualizarActivity.java* dentro de nuestro paquete, y agregamos el siguiente código:

```

package sv.ues.fia.basedatoscarnet;
import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class AlumnoActualizarActivity extends Activity {

    ControlBDCarnet helper;
    EditText editCarnet;
    EditText editNombre;
    EditText editApellido;
    EditText editSexo;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_alumno_actualizar);
        helper = new ControlBDCarnet(this);
        editCarnet = (EditText) findViewById(R.id.editCarnet);
        editNombre = (EditText) findViewById(R.id.editNombre);
        editApellido = (EditText) findViewById(R.id.editApellido);
        editSexo = (EditText) findViewById(R.id.editSexo);
    }

    public void actualizarAlumno(View v) {
        Alumno alumno = new Alumno();
        alumno.setCarnet(editCarnet.getText().toString());
        alumno.setNombre(editNombre.getText().toString());
        alumno.setApellido(editApellido.getText().toString());
        alumno.setSexo(editSexo.getText().toString());

        helper.abrir();
        String estado = helper.actualizar(alumno);
        helper.cerrar();

        Toast.makeText(this, estado, Toast.LENGTH_SHORT).show();
    }

    public void limpiarTexto(View v) {

        editCarnet.setText("");
        editNombre.setText("");
        editApellido.setText("");
    }
}

```

```

        editSexo.setText("");
    }
}

```

Método eliminar (Alumno)

Modificaremos ahora el método `eliminar(Alumno)` por lo que agregaremos este código donde corresponde en la clase `ControlDBCarnet.java`

```

public String eliminar(Alumno alumno){
    String regAfectados="filas afectadas= ";
    int contador=0;
    if (verificarIntegridad(alumno,3)) {
        contador+=db.delete("nota", "carnet='"+alumno.getCarnet()+"'", null);
    }
    contador+=db.delete("alumno", "carnet='"+alumno.getCarnet()+"'", null);
    regAfectados+=contador;
    return regAfectados;
}

```

Observación:

Podemos ver que en este caso, la cláusula `WHERE` la construimos completamente sin usar argumentos como en el caso de `“modificar(alumno)”`, y utilizamos un `contador` para acumular los valores de retorno del método `delete()`, el cual regresa el número de filas eliminadas en la acción. Además, ya que debemos respetar la integridad referencial, antes de eliminar de la tabla **Alumno** debemos eliminar todos los datos que hagan referencia al Alumno en cuestión de la tabla **Nota**.

Layout `activity_alumno_eliminar.xml` (Alumno)

Crea un nuevo `layout` con el nombre `activity_alumno_eliminar.xml` que usaremos como interfaz para eliminar de la tabla `Alumno`.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/carnet" />
    <EditText
        android:id="@+id/editCarnet"
        android:layout_width="fill_parent"

```



```

        android:layout_height="wrap_content"
        android:inputType="text"
        android:maxLength="7"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/eliminar"
        android:onClick="eliminarAlumno"/>
</LinearLayout>

```

Para eliminar un Alumno solo haremos uso del **Carnet**, por lo que definimos un solo EditText editable hasta 7 caracteres.

Activity AlumnoEliminarActivity.java (Alumno)

Luego modificaremos la clase llamada *AlumnoEliminarActivity.java* dentro de nuestro paquete, y agregamos el siguiente código:

```

package sv.ues.fia.basedatoscarnet;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class AlumnoEliminarActivity extends Activity {

    EditText editCarnet;
    ControlBDCarnet controlhelper;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_alumno_eliminar);
        controlhelper=new ControlBDCarnet (this);
        editCarnet=(EditText)findViewById(R.id.editCarnet);
    }

    public void eliminarAlumno(View v){
        String regEliminadas;
        Alumno alumno=new Alumno();
        alumno.setCarnet(editCarnet.getText().toString());
        controlhelper.abrir();
        regEliminadas=controlhelper.eliminar(alumno);
        controlhelper.cerrar();
        Toast.makeText(this, regEliminadas, Toast.LENGTH_SHORT).show();
    }
}

```

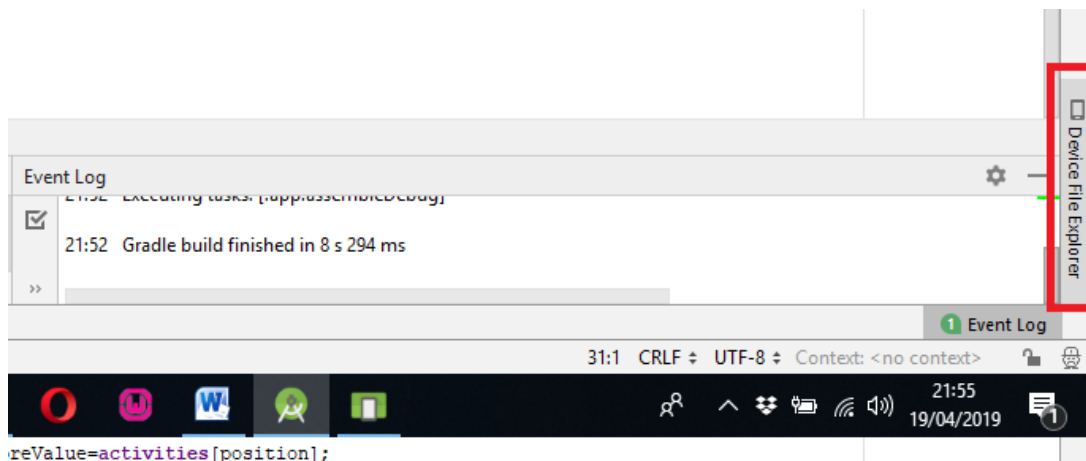
Con lo anterior tienes ya el mantenimiento completo de la tabla de alumno.,

Para que tu proyecto tenga datos de prueba (pre llenado) deberas sustituir los métodos de insertar para las tablas: Materia ([Anexo 1](#)) y Nota ([Anexo 4](#)) y luego:

Guarda el proyecto y ejecútalo, en el menú principal, corre la opción **Llenar Base de Datos**. Para ver datos de muestra.

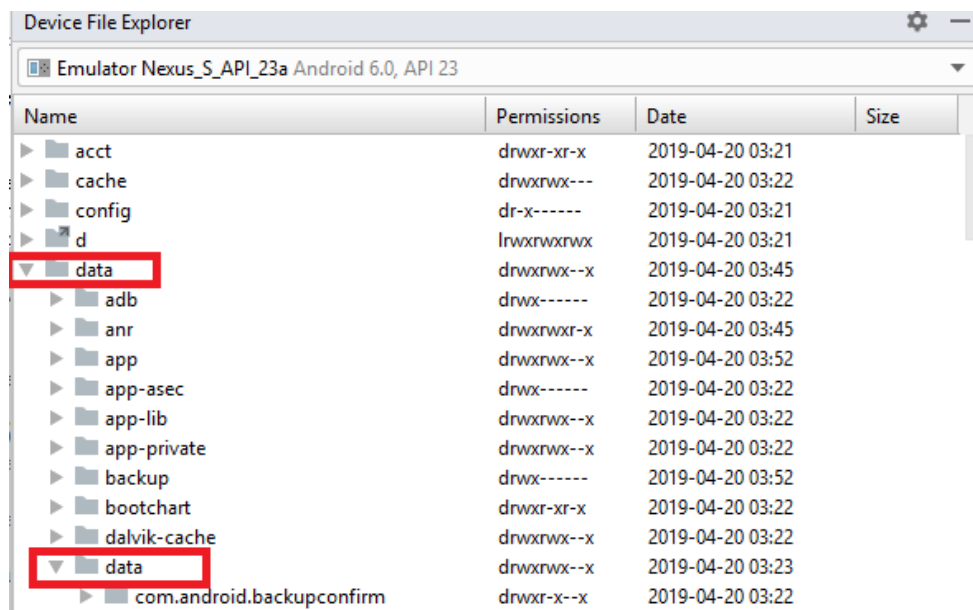
Copiar Base de Datos inicial y ver su contenido

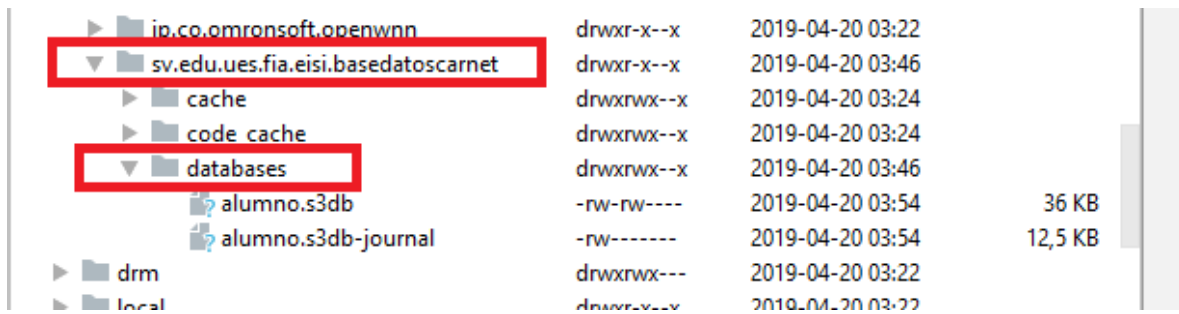
Sin cancelar la ejecución de su proyecto, entre al Device explorer



Dentro de el, encontrara la perspectiva DDMS.

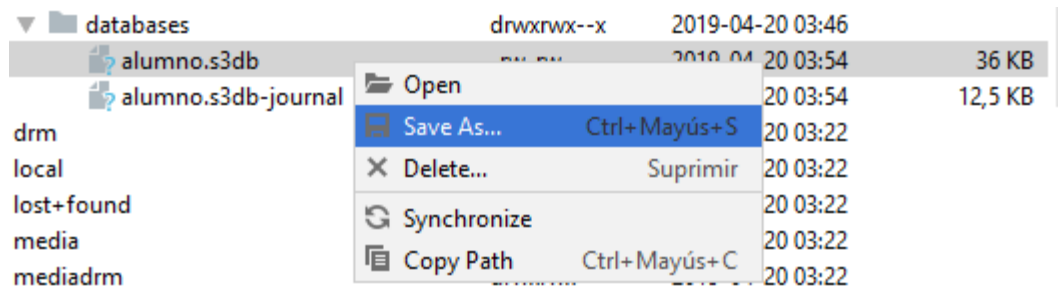
Expanda las carpetas data, data.



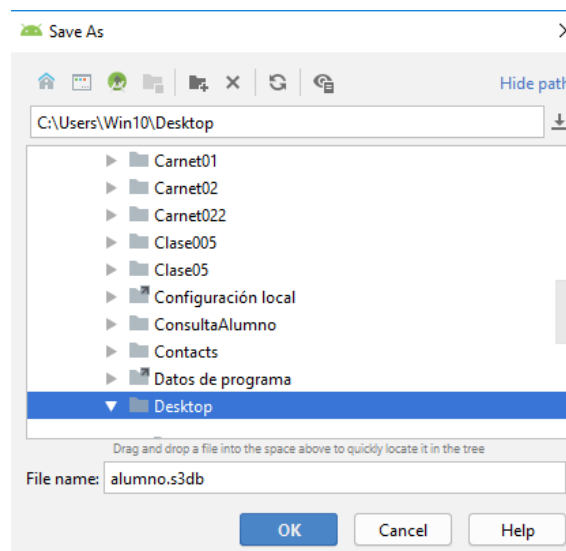


Buscar el paquete del proyecto y expandirlo hasta entrar en la carpeta databases, luego presionar clic en alumno3.s3db

Teniendo seleccionado el archivo de bdd, presionar clic derecho, save as...



Luego ubica el archivo en una carpeta como "Desktop" o en otra carpeta que tengas acceso



Usa el SQLite administrator para ver el estado actual de la base de datos.

SQLite Administrator - alumno.s3db

Base De Datos Tabla Indice Ver Trigger Consulta Datos Ayuda

Consulta SQL Resultado Editar Datos

carnet	nombre	apellido	sexo	matganadas
OO12035	Carlos	Orantes	M	0
OF12044	Pedro	Ortiz	M	0
GG11098	Sara	Gonzales	F	0
CC12021	Gabriela	Coto	F	0

En este estado verifica que todas las tablas tengan registros, Sino, indica por que no hay datos en las tablas?_____

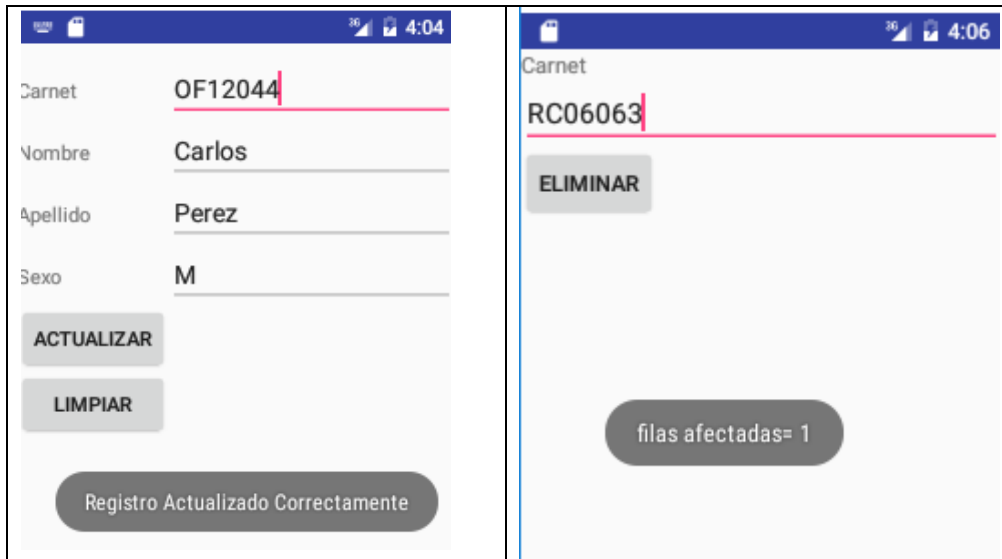
Probar opciones hechas para la tabla Alumnos desde el simulador

Luego haz una prueba completa de las opciones de **Alumno**, Entra a **insertar**, pon tus datos y presiona **insertar**, hazlo de nuevo con el registro que se muestra en la primer imagen, luego, **consulta** tus datos, después **Actualiza** uno de tus datos a excepción del carnet (llave), vuelve a **consultar** tus datos, luego **elimina** el registro y vuelve a **consultarlo**.

The image shows two screenshots of a mobile application interface for managing student records.

Left Screenshot (3:58): Shows the 'INSERTAR' screen. The fields are: Carnet (RC06063), Nombre (Amanda Lucia), Apellido (Romero Castro), and Sexo (F). There are buttons for 'INSERTAR' and 'LIMPIAR'. A message at the bottom says 'Registro Insertado N°= 5'.

Right Screenshot (4:02): Shows the 'CONSULTAR' screen. The fields are: Carnet (OO12035), Nombre (Carlos), Apellido (Orantes), and Sexo (M). There are buttons for 'CONSULTAR' and 'LIMPIAR'. The 'Materias Ganadas' field shows 0.



Ver el contenido de la tabla Alumno de nuevo

Renombra la copia del archivo alumno.s3db y ponle alumnoV1.s3db . Luego, así como se hizo anteriormente, haz una nueva copia de la base de alumnoV2.s3db (desde DDMS) , revisa que los datos coincidan con los que vas modificando.

SQLite Administrator - alumnov2

Base De Datos Tabla Indice Ver Trigger Consulta Datos Ayuda

Consulta SQL Resultado Editar Datos

carnet	nombre	apellido	sexo	matganadas
OO12035	Carlos	Orantes	M	0
OF12044	Carlos	Perez	M	0
GG11098	Sara	Gonzales	F	0
CC12021	Gabriela	Coto	F	0
CARNETX	su nombre	su apellido	X	0

Al verificar estos datos, continúe con la creación de los módulos de la Tabla NOTA, para continuar.

Tabla NOTA

Método insertar (Nota)

En esta parte veremos cómo insertar alguna nota de nuestra base de datos y para ello agregaremos el siguiente código en el método correspondiente en la clase ControlIDBCarnet.java

```
public String insertar(Nota nota){
    String regInsertados="Registro Insertado N°= ";
    long contador=0;

    if(verificarIntegridad(nota,1))
```

```

        {
            ContentValues notas = new ContentValues();
            notas.put("carnet", nota.getCarnet());
            notas.put("codmateria", nota.getCodmateria());
            notas.put("ciclo", nota.getCiclo());
            notas.put("notafinal", nota.getNotafinal());
            contador=db.insert("nota", null, notas);
        }

        if(contador==-1 || contador==0)
        {
            regInsertados= "Error al Insertar el registro, Registro
Duplicado. Verificar inserción";
        }
        else {
            regInsertados=regInsertados+contador;
        }

        return regInsertados;
    }
}

```

Observación:

Es necesario realizar la validación del registro de alumno y del código de materia de manera que se verifique que exista un carnet y el código de materia a insertar dentro de la tabla alumno y materia. En este caso la validación queda planteada pero es necesario que el estudiante realice **el método insertar de materia y el formulario(Activity – Layout de Insertar Materia)**, para que dicho código implementado no le de error o no lo deje insertar ningún tipo de nota al no existir la materia a la cual hace referencia. Por lo demás sigue la misma secuencia descrita en InsertarAlumno.

Layout activity_nota_insertar.xml (Nota)

El siguiente paso es modificar la interfaz de usuario con el nombre `activity_nota_insertar.xml` para insertar a la tabla Nota.

```

<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:paddingTop="10dip"
    android:stretchColumns="1" >

    <TableRow>
        <TextView android:text="@string/carnet"/>
        <EditText
            android:id="@+id/editCarnet"
            android:inputType="text"
            android:maxLength="7" />
    </TableRow>

```

```

</TableRow>

<TableRow>
    <TextView android:text="@string/codmateria" />
    <EditText
        android:id="@+id/editCodmateria"
        android:inputType="text"
        android:maxLength="6" />
</TableRow>

<TableRow>
    <TextView android:text="@string/ciclo" />
    <EditText
        android:id="@+id/editCiclo"
        android:inputType="number"
        android:maxLength="5" />
</TableRow>

<TableRow>
    <TextView android:text="@string/notafinal" />
    <EditText
        android:id="@+id/editNotafinal"
        android:inputType="text"
        android:maxLength="4" />
</TableRow>

<TableRow>
    <TextView android:text="@string/opciones" />
</TableRow>
<TableRow>

    <Button
        android:onClick="insertarNota"
        android:text="@string/insertar" />

    <Button
        android:onClick="limpiarTexto"
        android:text="@string/limpiar" />
</TableRow>
</TableLayout>

```

Activity NotaInsertarActivity.java (Nota)

Luego modificaremos la clase llamada *NotaInsertarActivity.java* dentro de nuestro paquete, y agregamos el siguiente código:

```

package sv.ues.fia.basedatoscarnet;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;

```

```

import android.widget.Toast;

public class NotaInsertarActivity extends Activity {

    ControlBDCarnet helper;
    EditText editCarnet;
    EditText editCodmateria;
    EditText editCiclo;
    EditText editNotafinal;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_nota_insertar);

        helper = new ControlBDCarnet(this);
        editCarnet = (EditText) findViewById(R.id.editCarnet);
        editCodmateria = (EditText) findViewById(R.id.editCodmateria);
        editCiclo = (EditText) findViewById(R.id.editCiclo);
        editNotafinal = (EditText) findViewById(R.id.editNotafinal);
    }

    public void insertarNota(View v) {

        String regInsertados;
        String carnet=editCarnet.getText().toString();
        String codmateria=editCodmateria.getText().toString();
        String ciclo=editCiclo.getText().toString();
        Float notafinal=Float.valueOf(editNotafinal.getText().toString());
        Nota nota= new Nota();
        nota.setCarnet(carnet);
        nota.setCodmateria(codmateria);
        nota.setCiclo(ciclo);
        nota.setNotafinal(notafinal);
        helper.abrir();
        regInsertados=helper.insertar(nota);
        helper.cerrar();
        Toast.makeText(this, regInsertados, Toast.LENGTH_SHORT).show();
    }

    public void limpiarTexto(View v) {

        editCarnet.setText("");
        editCodmateria.setText("");
        editCiclo.setText("");
        editNotafinal.setText("");
    }
}

```


Método consultar (Nota)

Agregamos el siguiente código al método *consultarNota* dentro de la clase *ControlBDCarnet.java*:

```
public Nota consultarNota(String carnet, String codmateria, String ciclo){
    String[] id = {carnet, codmateria, ciclo};
    Cursor cursor = db.query("nota", camposNota, "carnet = ? AND
codmateria = ? AND ciclo = ?", id, null, null, null);
    if(cursor.moveToFirst()){
        Nota nota = new Nota();
        nota.setCarnet(cursor.getString(0));
        nota.setCodmateria(cursor.getString(1));
        nota.setCiclo(cursor.getString(2));
        nota.setNotafinal(cursor.getFloat(3));
        return nota;
    }else{
        return null;
    }
}
```

Observaciones:

Para realizar nuestra consulta a este método le mandamos como parámetros el carnet del alumno, el código de la materia q deseamos conocer su nota y el ciclo respectivo. Igualmente utilizamos una variable de tipo Cursor para extraer cada uno de los datos del objeto Nota.

Layout activity_nota_consulta.xml (Nota)

Ahora dentro de la carpeta Layout modificamos el archivo *activity_nota_consulta.xml* y le agregamos el siguiente código:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/carnet"/>

    <EditText
        android:id="@+id/editCarnet"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:inputType="text"
```

```
        android:maxLength="7"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/codmateria"/>

        <EditText
            android:id="@+id/editCodmateria"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:inputType="text"
            android:maxLength="7"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/ciclo" />

        <EditText
            android:id="@+id/editCiclo"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:inputType="number"
            android:maxLength="5"/>

        <LinearLayout
            android:id="@+id/LinearLayout1"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content">

            <Button
                android:id="@+id/botonConsultar"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/consultar"
                android:onClick="consultarNota"/>

            <Button
                android:id="@+id/botonLimpiar"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/limpiar"
                android:onClick="LimpiarTexto"/>

        </LinearLayout>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/notafinal" />

        <EditText
            android:id="@+id/editNotafinal"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
```

```

        android:inputType="text"
        android:enabled = "false"
        android:focusable="false"
            android:focusableInTouchMode="false"
            android:cursorVisible="false"
            android:longClickable="false" />
</LinearLayout>

```

Activity *NotaConsultarActivity.java* (Nota)

Luego de esto modificaremos nuestra clase *NotaConsultarActivity.java* y agregamos lo siguiente:

```

package sv.ues.fia.basedatoscarnet;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class NotaConsultarActivity extends Activity {

    ControlBDCarnet helper;
    EditText editCarnet;
    EditText editCodmateria;
    EditText editCiclo;
    EditText editNotafinal;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_nota_consultar);
        helper = new ControlBDCarnet(this);

        editCarnet = (EditText) findViewById(R.id.editCarnet);
        editCodmateria = (EditText) findViewById(R.id.editCodmateria);
        editCiclo = (EditText) findViewById(R.id.editCiclo);
        editNotafinal = (EditText) findViewById(R.id.editNotafinal);
    }

    public void consultarNota(View v) {

        helper.abrir();

        Nota nota = helper.consultarNota(editCarnet.getText().toString(),
editCodmateria.getText().toString(), editCiclo.getText().toString());

        helper.cerrar();

        if(nota == null)
            Toast.makeText(this, "Nota no registrada",
Toast.LENGTH_LONG).show();

```

```

        else{
            editNotafinal.setText(String.valueOf(nota.getNotafinal()));
        }
    }

    public void limpiarTexto(View v) {
        editCarnet.setText("");
        editCodmateria.setText("");
        editCiclo.setText("");
        editNotafinal.setText("");
    }
}

```

Método actualizar (Nota)

Agregamos el siguiente código al método *actualizarNota* dentro de la clase *ControlBDAlumno.java*:

```

public String actualizar(Nota nota){

    if(verificarIntegridad(nota, 2)){
        String[] id = {nota.getCodmateria(), nota.getCarnet(),
nota.getCiclo()};
        ContentValues cv = new ContentValues();
        cv.put("notafinal", nota.getNotafinal());
        db.update("nota", cv, "codmateria = ? AND carnet = ? AND
ciclo = ?", id);
        return "Registro Actualizado Correctamente";
    }else{
        return "Registro no Existe";
    }

}
}

```

Observaciones:

De igual manera que en *actualizarAlumno* se utiliza el método *update* para modificar los valores existentes dentro de la base. En este caso verificamos que los valores de *codmateria*, *carnet* y *ciclo* ya se encuentren registrados y asignarle un nuevo valor a *notafinal*.

Layout activity_nota_actualizar.xml (Nota)

Ahora modificamos a nuestra carpeta **Layout** el archivo *activity_nota_actualizar.xml* con el siguiente código.

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:paddingTop="10dip"
    android:stretchColumns="1" >

    <TableRow>

        <TextView android:text="@string/carnet" />

        <EditText
            android:id="@+id/editCarnet"
            android:inputType="text"
            android:maxLength="7" />
    </TableRow>

    <TableRow>

        <TextView android:text="@string/codmatnota" />

        <EditText
            android:id="@+id/editCodigo"
            android:inputType="text"
            android:maxLength="6" />
    </TableRow>

    <TableRow>

        <TextView android:text="@string/ciclo" />

        <EditText
            android:id="@+id/editCiclo"
            android:inputType="text"
            android:maxLength="5" />
    </TableRow>

    <TableRow>

        <TextView android:text="@string/notafinal" />

        <EditText
            android:id="@+id/editNota"
            android:inputType="numberDecimal" />
    </TableRow>

    <TableRow>

        <Button
            android:onClick="actualizarNota"
            android:text="@string/actualizar" />
    </TableRow>

    <TableRow>

        <Button
```

```

        android:onClick="LimpiarTexto"
        android:text="@string/Limpiar" />
    </TableRow>
</TableLayout>

```

Activity NotaActualizarActivity.java (Nota)

Por ultimo modificamos la clase *NotaActualizarActivity.java* como se muestra a continuación:

```

package sv.ues.fia.basedatoscarnet;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class NotaActualizarActivity extends Activity {

    ControlBDCarnet helper;
    EditText editCarnet;
    EditText editCodigo;
    EditText editCiclo;
    EditText editNota;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_nota_actualizar);
        helper = new ControlBDCarnet(this);
        editCarnet = (EditText) findViewById(R.id.editCarnet);
        editCodigo = (EditText) findViewById(R.id.editCodigo);
        editCiclo = (EditText) findViewById(R.id.editCiclo);
        editNota = (EditText) findViewById(R.id.editNota);
    }

    public void actualizarNota(View v) {
        Nota nota = new Nota();
        nota.setCarnet(editCarnet.getText().toString());
        nota.setCodmateria(editCodigo.getText().toString());
        nota.setCiclo(editCiclo.getText().toString());
        nota.setNotafinal(Float.valueOf(editNota.getText().toString()));

        helper.abrir();
        String estado = helper.actualizar(nota);
        helper.cerrar();

        Toast.makeText(this, estado, Toast.LENGTH_SHORT).show();
    }
}

```

```

public void limpiarTexto(View v) {
    editCarnet.setText("");
    editCodigo.setText("");
    editCiclo.setText("");
    editNota.setText("");
}
}

```

Método eliminar (Nota)

Modificaremos ahora el método eliminar (Nota) por lo que agregaremos este código donde corresponde en la clase *ControlDBCarnet.java*

```

public String eliminar(Nota nota){
    String regAfectados="filas afectadas= ";
    int contador=0;
    String where="carnet='"+nota.getCarnet()+"'";
    where=where+" AND codmateria='"+nota.getCodmateria()+"'";
    where=where+" AND ciclo='"+nota.getCiclo()+"'";
    contador+=db.delete("nota", where, null);
    regAfectados+=contador;
    return regAfectados;
}

```

Observación:

Para eliminar una nota de una materia de un alumno específico, construimos la cláusula WHERE indicando el carnet, codmateria y ciclo que hacen referencia a ella.

Layout *activity_nota_eliminar.xml* (Nota)

Modificar el *layout* con el nombre *activity_nota_eliminar.xml* que usaremos como interfaz para eliminar registros de la tabla Nota.

```

<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="match_parent"
    android:stretchColumns="1"
    android:paddingTop="4dip"
    >

    <TableRow >
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/carnet"/>
    </TableRow>
</TableLayout>

```

```

<EditText
    android:id="@+id/editCarnet"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="text"
    android:maxLength="7"/>
</TableRow>

<TableRow>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/codmatnota"/>
    <EditText
        android:id="@+id/editCodmateria"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:inputType="text"
        android:maxLength="6"/>
</TableRow>

<TableRow>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/ciclo"/>
    <EditText
        android:id="@+id/editCiclo"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:inputType="number"
        android:maxLength="1"/>
</TableRow>

<TableRow>
<Button
    android:layout_width="wrap_content"
    android:text="@string/eliminar"
    android:onClick="eliminarNota"/>
<Button
    android:layout_width="wrap_content"
    android:text="@string/limpiar"
    android:onClick="limpiar"/>
</TableRow>
</TableLayout>

```

Podemos ver en el XML que usaremos tres EditText donde se le pedirá al usuario ingresar el carnet, el código de materia y el ciclo para el cual se quiere eliminar la nota.

Activity EliminarNotaActivity.java (Nota)

Luego modificaremos la nueva clase llamada *EliminarNotaActivity.java* dentro de nuestro paquete, y agregamos el siguiente código:

```

package sv.ues.fia.basedatoscarnet;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class NotaEliminarActivity extends Activity {

    EditText editCarnet, editCodmateria, editCiclo;
    ControlBDCarnet controlhelper;

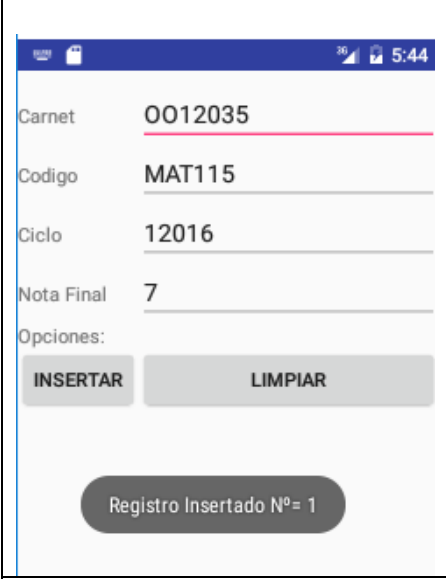

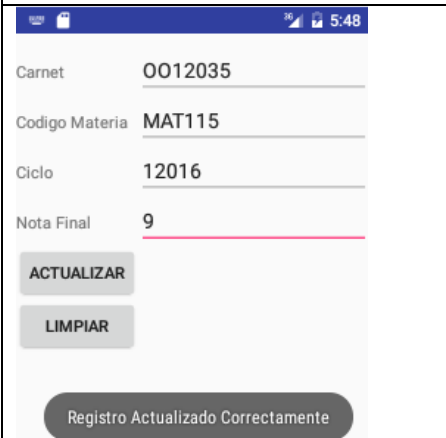

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_nota_eliminar);
        controlhelper=new ControlBDCarnet(this);
        editCarnet=(EditText)findViewById(R.id.editCarnet);
        editCodmateria=(EditText)findViewById(R.id.editCodmateria);
        editCiclo=(EditText)findViewById(R.id.editCiclo);
    }

    public void eliminarNota(View v){
        String regEliminadas;
        Nota nota=new Nota();
        nota.setCarnet(editCarnet.getText().toString());
        nota.setCodmateria(editCodmateria.getText().toString());
        nota.setCiclo(editCiclo.getText().toString());
        controlhelper.abrir();
        regEliminadas=controlhelper.eliminar(nota);
        controlhelper.cerrar();
        Toast.makeText(this, regEliminadas, Toast.LENGTH_SHORT).show();
    }

    public void limpiar(View v){
        editCarnet.setText("");
        editCodmateria.setText("");
        editCiclo.setText("");
    }
}

```

Guarde el proyecto y puebe las opciones CRUD de la tabla **Nota**. Se debería ver algo como esto al ingresar a las cuatro operaciones Insertar, Consultar, Modificar y Eliminar respectivamente,

Puede probarlo, introduciendo primero el alumno(con el carnet en la tabla Alumno), luego introduzca la nota ficticia de 7 de la asignatura MAT115, CICLO 12016, consúltela, actualicela a 9 de nota, vuelva a consúltela y por último elimínela

Actividad final en la guía de laboratorio:

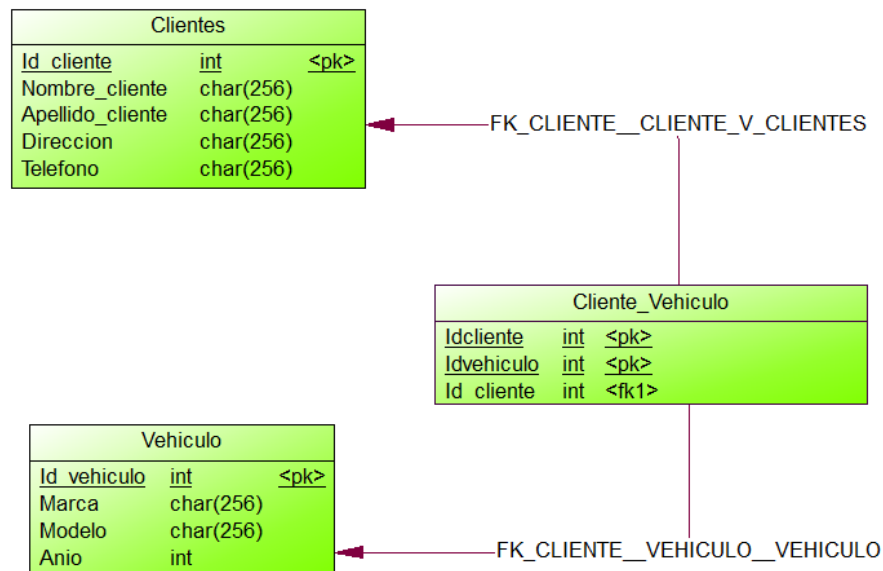
Completar la parte de la base de datos enfocada en la tabla “Materia” y realizar los métodos de inserción, eliminación, consulta y modificación. Ocupar la interfaz de usuario respectiva y agregarle a la aplicación todas las vistas que hagan referencia a dicha tabla. Cuidar de realizar la integridad de la base de datos usando siempre el método “verificarIntegridad”. Y agregar todos los métodos necesarios en la clase “ControlBDAlumno”. Realizar todas las modificaciones que sean pertinentes.

Al finalizar envía el proyecto a la opción desubida de proyecto guía 4 en el aula virtual, este debe contener solamente el proyecto BaseDatosCarnet concluido(con los activities de la tabla materia) y comprimido.

Actividad opcional en la guía de laboratorio:

En otro proyecto, realiza la implementación de la siguiente base de datos, con la misma estructura de controlador, manteniendo la integridad de la información

Modelo Físico



Anexos

Anexo 1 Código de insertar materia en controlador(ControlBDCarnet)

```

public String insertar(Materia materia){
    String regInsertados="Registro Insertado N°= ";
    long contador=0;
    ContentValues mate = new ContentValues();
    mate.put("codmateria", materia.getCodmateria());
    mate.put("nommateria", materia.getNommateria());
    mate.put("unidadesval", materia.getUnidadesval());
    contador=db.insert("materia", null, mate);
    if(contador== -1 || contador==0)
    {
        regInsertados= "Error al Insertar el registro, Registro Duplicado.
Verificar inserción";
    }
    else {
        regInsertados=regInsertados+contador;
    }
    return regInsertados;
}

```

Anexo 2 Layout insertarmateria.xml

```

<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:paddingTop="30dip"
    android:stretchColumns="1" >

    <TableRow>
        <TextView android:text="@string/codmateria"/>
        <EditText
            android:id="@+id/editCodmateria"
            android:inputType="text"
            android:maxLength="7" />
    </TableRow>

    <TableRow>
        <TextView android:text="@string/nommateria" />
        <EditText
            android:id="@+id/editNommateria"
            android:inputType="text"
            android:maxLength="30" />
    </TableRow>

    <TableRow>
        <TextView android:text="@string/unidadesval" />
        <EditText
            android:id="@+id/editUnidadesval"
            android:inputType="text"
            android:maxLength="1" />
    </TableRow>

```

```

<TableRow>
    <TextView android:text="@string/opciones" />
</TableRow>

<TableRow>
    <Button
        android:onClick="insertarMateria"
        android:text="@string/insertar" />
    <Button
        android:onClick="limpiarTexto"
        android:text="@string/limpiar" />
</TableRow>

</TableLayout>

```

Anexo 3 Código de MateriaInsertarActivity.java

```

package sv.ues.fia.basedatoscarnet;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class MateriaInsertarActivity extends Activity {

    ControlBDCarnet helper;
    EditText editCodmateria;
    EditText editNommateria;
    EditText editUnidadesval;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_materia_insertar);
        helper = new ControlBDCarnet(this);
        editCodmateria = (EditText) findViewById(R.id.editCodmateria);
        editNommateria = (EditText) findViewById(R.id.editNommateria);
        editUnidadesval = (EditText) findViewById(R.id.editUnidadesval);
    }

    public void insertarAlumno(View v) {
        String codmateria=editCodmateria.getText().toString();
        String nommateria=editNommateria.getText().toString();
        String unidadesval=editUnidadesval.getText().toString();
        String regInsertados;
        Materia materia=new Materia();
        materia.setCodmateria(codmateria);
        materia.setNommateria(nommateria);
        materia.setUnidadesval(unidadesval);
        helper.abrir();
        regInsertados=helper.insertar(materia);
        helper.cerrar();
        Toast.makeText(this, regInsertados, Toast.LENGTH_SHORT).show();
    }

    public void limpiarTexto(View v) {

        editCodmateria.setText("");
        editNommateria.setText("");
    }
}

```

```

        editUnidadesval.setText("");
    }
}

```

Datos a Insertar en tabla Materia

The screenshot shows a mobile application interface with a white background and a blue header. The header contains a signal strength icon, a battery icon, and the time '5:27'. The form consists of four input fields stacked vertically. The first field is labeled 'Codigo' and contains the text 'MAT115'. The second field is labeled 'Nombre Materia' and contains 'Matematicas 1'. The third field is labeled 'Unidades Valorativas' and contains '4'. Below the input fields is a section labeled 'Opciones:' containing two buttons: 'INSERTAR' and 'LIMPIAR'.

Anexo 4Codigo de insertar nota en controlador (ControlBDCarnet)

```

public String insertar(Nota nota){
    String regInsertados="Registro Insertado N°= ";
    long contador=0;

    if(verificarIntegridad(nota,1))
    {
        ContentValues notas = new ContentValues();
        notas.put("carnet", nota.getCarnet());
        notas.put("codmateria", nota.getCodmateria());
        notas.put("ciclo", nota.getCiclo());
        notas.put("notafinal", nota.getNotafinal());
        contador=db.insert("nota", null, notas);
    }

    if(contador==-1 || contador==0)
    {
        regInsertados= "Error al Insertar el registro, Registro
Duplicado. Verificar inserción";
    }
    else {
        regInsertados=regInsertados+contador;
    }

    return regInsertados;
}

```