

**UNIVERSIDAD
JEAN PIAGET**

Ingeniería En Sistemas Computacionales

Asignatura:
Análisis y Diseño Orientado a Objetos

Descripción del proyecto sincronizar base con la app

Presenta:
Elian Shair Armendariz Puch

Catedrático:
ING. Adrián García García.URL

Veracruz, Ver.

2025

INDICE

I.	Introducción	2
II.	Estructura General del Proyecto	3
2.1	Componentes principales:.....	3
2.2	Tecnologías Utilizadas	3
2.3	Descripción de las Clases.....	3
2.4	Descripción Detallada del Método sincro(View v)	5
III.	Conclusión.....	8

I. Introducción

Este proyecto Android está diseñado para la **gestión de información de alumnos** a través de una aplicación móvil. La aplicación permite registrar, consultar, eliminar y sincronizar alumnos, utilizando tanto una **base de datos local (SQLite)** como una **base de datos remota (MySQL vía PHP)**.

La arquitectura de la aplicación mezcla elementos de almacenamiento local con sincronización en línea, permitiendo que los usuarios puedan trabajar con datos sin conexión y luego actualizar la información desde un servidor remoto cuando sea necesario. A través de la clase Volley, la app hace peticiones HTTP para obtener un **array JSON** que contiene los datos de los alumnos, los cuales son procesados e insertados localmente.

II. Estructura General del Proyecto

2.1 Componentes principales:

- **SQLite:** Se utiliza como base de datos local en el dispositivo.
- **MySQL + PHP:** En el lado del servidor, permite almacenar y devolver información estructurada en formato JSON.
- **Volley:** Biblioteca de Android utilizada para gestionar solicitudes HTTP de forma asíncrona.
- **GridView:** Elemento visual que muestra la lista de alumnos en pantalla.
- **Clases Java:** Estructuran y organizan la lógica del sistema.

2.2 Tecnologías Utilizadas

- **Android:**
 - SQLite para almacenamiento local
 - Volley para solicitudes HTTP
 - JSON para intercambio de datos
- **Servidor:**
 - PHP para exponer los datos en formato JSON
 - MySQL como base de datos remota

2.3 Descripción de las Clases

Clase MainActivity

Ubicación: com.example.ejemplo4.MainActivity

Funcionalidad:

Esta clase representa la pantalla principal de la aplicación. Se encarga de:

- Consultar los alumnos almacenados en la base de datos local.
- Mostrar los datos en una GridView.
- Redireccionar a la pantalla de inserción de datos (ActivityAlta) mediante un botón.

Atributos utilizados:

- GridView grid: Control visual que presenta los datos al usuario.
- ArrayList<String> list: Almacena los datos leídos de SQLite.
- ArrayAdapter<String> adapter: Se encarga de adaptar los datos de la lista al formato visual.

Métodos:

- onCreate(Bundle savedInstanceState):
Configura la interfaz principal, accede a la base de datos local y carga los alumnos en la vista.
- consulta():
Es una función de refuerzo que consulta los datos nuevamente desde SQLite. Puede llamarse desde otras actividades si se desea refrescar el contenido.

Clase ActivityAlta

Ubicación: com.example.ejemplo4.ActivityAlta

Funcionalidad:

Esta clase representa la pantalla secundaria, que permite:

- Agregar alumnos nuevos.
- Consultar alumnos existentes.
- Eliminar alumnos.
- Sincronizar datos con un servidor web remoto.

Atributos utilizados:

- EditText dato1: Campo de texto para el código del alumno.
- EditText dato2: Campo de texto para el nombre del alumno.

Métodos:

- onCreate(Bundle savedInstanceState):
Inicializa la interfaz gráfica y asigna comportamiento al botón que regresa a MainActivity.

- consulta(View v):
Consulta un alumno por su código en la base de datos local. Si el alumno existe, se muestran sus datos; si no, se notifica al usuario.
- alta(View v):
Inserta un nuevo registro en la base local usando ContentValues.
- eliminacion(View v):
Elimina un alumno específico por código, e informa al usuario si fue exitoso o no.
- sincro(View v):
Sincroniza todos los datos locales con los del servidor remoto (descripción detallada más abajo).


2.4 Descripción Detallada del Método sincro(View v)

Objetivo

El método sincro() tiene como finalidad **sincronizar los datos locales de alumnos con la base de datos remota** almacenada en un servidor web. Esto se logra mediante una conexión HTTP que descarga todos los registros de la base de datos MySQL en formato JSON y los guarda localmente en SQLite.

Lógica paso a paso

Se define la URL del servidor remoto:



```
1 String URL = "http://elianarmendariz.atwebpages.com/alumnos.php";
2
```

Este archivo PHP se encarga de consultar los registros de la base de datos MySQL y devolver un JSON.

Se crea una cola de peticiones HTTP con la librería Volley:

```
1 RequestQueue queue = Volley.newRequestQueue(this);  
2
```

Se construye un JsonRequest:

Esta solicitud obtiene un array JSON del servidor:

```
1 JsonRequest jsonArrayRequest = new JsonRequest(  
2     Request.Method.GET, URL, null,  
3     response -> { ... },  
4     error -> { ... }  
5 );
```

Cuando se recibe la respuesta (response):

Se instancia el helper de SQLite:

```
1 AlumnosDbHelper dbHelper = new AlumnosDbHelper(this, "administracion", null, 1);  
2 SQLiteDatabase db = dbHelper.getWritableDatabase();
```

Se eliminan todos los registros existentes:

```
1 db.delete("alumnos", null, null);
```

Se recorre el JSON recibido y se insertan los registros:

```

1  for (int i = 0; i < response.length(); i++) {
2      JSONObject obj = response.getJSONObject(i);
3      int codigo = obj.getInt("codigo");
4      String nombre = obj.getString("nombre");
5
6      ContentValues values = new ContentValues();
7      values.put("codigo", codigo);
8      values.put("nombre", nombre);
9      db.insert("alumnos", null, values);
10 }

```

Se muestra una notificación al usuario y se regresa a la pantalla principal:

```

1  Toast.makeText(this, "Sincronización completada", Toast.LENGTH_SHORT).show();
2  Intent intent = new Intent(ActivityAlta.this, MainActivity.class);
3  startActivity(intent);
4  finish();

```

Si ocurre un error de red o de conexión:

Se informa al usuario mediante un mensaje de error y se imprime en el log.

Lado del Servidor: Script PHP

Archivo: alumnos.php

Este archivo accede a una base de datos MySQL y genera un JSON con todos los registros de la tabla alumnos.

Funcionamiento del PHP

Establece la conexión con la base de datos:

```

1  $mysqli = new mysqli("fdb1028.awardspace.net", "usuario", "contraseña", "base");

```

Ejecuta una consulta SQL:


```
1 $sql = "SELECT codigo, nombre FROM alumnos";
```

Recorre el resultado y genera un arreglo asociativo

```
1 while ($row = $result->fetch_assoc()) {  
2     $alumnos[] = $row;  
3 }
```

Devuelve los datos codificados como JSON

```
1 echo json_encode($alumnos);
```

III. Conclusión

Este proyecto es una implementación práctica de una aplicación de gestión de datos que integra:

- Almacenamiento local con SQLite.
- Acceso y visualización de datos mediante GridView.
- Inserción, eliminación y búsqueda de datos.
- Sincronización con una base de datos remota utilizando Volley y JSON.

El método sincro() representa una funcionalidad crítica que conecta el mundo local con el remoto, permitiendo que la aplicación trabaje offline y posteriormente se actualice. El uso de tecnologías como **Volley, JSON, PHP y MySQL** demuestra un enfoque profesional y escalable para la sincronización de datos en dispositivos móviles Android.

IV. DIAGRAMA DE CLASES DEL PROYECTO

