

TP. FINAL APLICACIONES MOVILES Y CLOUD COMPUTING

-DOMENICO ALEJANDRO-

El proyecto es un CRUD simple que permite agendar alumnos con datos básicos y una foto opcional, permite crear y eliminar la base de datos y trabajar con los registros para crear, actualizar y eliminar.

Breve descripción de las clases mas importantes:

Clase principal es : MainActivity

Es la actividad principal de la aplicación que muestra la pantalla inicial con los botones para realizar diferentes acciones (crear, eliminar la base de datos, agregar, editar o listar alumnos).

Método: onCreate

```
override fun onCreate(savedInstanceState: Bundle?)
```

Qué hace:

Configura el diseño inicial de la actividad.

Establece ajustes visuales para que el contenido se ajuste a los bordes de la pantalla.

Llama a las funciones para crear la base de datos y las tablas necesarias al iniciar.

Configura los eventos de clic para los botones.

Método: createDatabase

```
private fun createDatabase()
```

Qué hace:

Verifica si existe una carpeta para almacenar la base de datos en el dispositivo (dentro de Documents/Database). Si no existe, crea la carpeta. Genera el archivo MyDb para la base de datos o abre el existente. Maneja errores si no se puede crear la base de datos.

Método: createTables

```
private fun createTables()
```

Qué hace:

Define las estructuras de las tablas en la base de datos:

student: Guarda información sobre alumnos (ID, nombre, dirección, turno, edad, y una foto).

class: Opcionalmente puede usarse para guardar información sobre clases o turnos.

Verifica si las tablas ya existen antes de crearlas.

Maneja errores si no es posible ejecutar las consultas SQL.

Método: setupListeners

```
private fun setupListeners()
```

Qué hace:

Configura las acciones para los botones principales.

Acción del botón "Crear Base de Datos"

```
binding.btnCreateDatabase.setOnClickListener { ... }
```

Qué hace:

Crea la carpeta Database si no existe. Crea el archivo MyDb y las tablas necesarias si no existen.

Muestra un mensaje de confirmación si la base de datos se creó o un mensaje indicando que ya existe.

Acción del botón "Eliminar Base de Datos"

```
binding.btnDeleteRecord.setOnClickListener { ... }
```

Qué hace:

Verifica si el archivo MyDb existe. Si existe, lo elimina. Muestra un mensaje de confirmación si la base de datos se eliminó o un mensaje indicando que no existe.

Método: addData

fun addData(view: View)

Qué hace:

Inicia la actividad Student en modo "Agregar".

Pasa el mensaje "add" a la actividad para indicar que se está agregando un nuevo alumno.

Método: editData

fun editData(view: View)

Qué hace:

Verifica si el archivo de la base de datos MyDb existe. Si existe, inicia la actividad ListActivity para listar, editar o eliminar alumnos. Si no existe, muestra un mensaje indicando que primero se debe crear la base de datos.

Flujo General del Código

Inicio:

Configura la interfaz y verifica la existencia de la base de datos y las tablas necesarias.

Botón "Crear Base de Datos":

Crea la base de datos y las tablas si no existen.

Botón "Eliminar Base de Datos":

Elimina la base de datos MyDb si existe.

Botón "Agregar Datos":

Lleva al usuario a la pantalla donde puede agregar un nuevo alumno.

Botón "Editar/Listar":

Lleva al usuario a la pantalla para listar, editar o eliminar alumnos si la base de datos existe.

Clase: Student

La clase Student representa una actividad que permite agregar, editar o eliminar alumnos en la base de datos SQLite. La funcionalidad incluye cargar imágenes para los alumnos, manejar las validaciones de los campos y realizar las operaciones correspondientes en la base de datos.

Métodos y Funcionalidades

onCreate

override fun onCreate(savedInstanceState: Bundle?)

Qué hace:

- Configura el diseño visual y los bordes de la pantalla. Verifica si la base de datos existe; si no, muestra un mensaje y cierra la actividad.
- Se conecta a la base de datos.
- Determina si la actividad está en modo "Agregar" o "Editar" según el mensaje recibido ("add" o "edit").
- Si está en modo "Editar", llama a showData para cargar la información del alumno en los campos.

uploadStudentImage

fun uploadStudentImage(view: View)

Qué hace:

- Permite al usuario seleccionar una imagen desde el almacenamiento del dispositivo.
- Convierte la imagen seleccionada en un Bitmap y luego en un ByteArray para almacenarla en la base de datos.
- Actualiza el ImageView con la imagen seleccionada.

insertData

private fun insertData()

Qué hace:

- Valida que los campos obligatorios (nombre, dirección, turno, edad) no estén vacíos.
- Inserta los datos del nuevo alumno en la tabla student.
- Limpia los campos después de la inserción y coloca el foco en el campo de Nombre.
- Muestra un mensaje de confirmación.

updateData

private fun updateData()

Qué hace:

- Valida que los campos obligatorios no estén vacíos.
- Actualiza los datos del alumno seleccionado en la tabla student.
- Regresa al menú principal después de realizar la actualización.
- Muestra un mensaje de confirmación.

delData

fun delData(view: View)

Qué hace:

- Elimina el registro del alumno seleccionado en la tabla student basado en su studentId.
- Regresa al menú principal después de la eliminación.
- Muestra un mensaje de confirmación.

goToMainMenu

private fun goToMainMenu()

Qué hace:

- Regresa a la pantalla principal (clase MainActivity).
- Limpia las actividades previas de la pila para evitar comportamientos inesperados.

showData

private fun showData()

Qué hace:

- Carga la información del alumno desde la base de datos basado en su studentId.
- Rellena los campos de texto y muestra la imagen correspondiente (si existe).
- Maneja posibles errores de lectura.

clr

private fun clr()

Qué hace:

- Limpia los valores de todos los campos de entrada y elimina la imagen del ImageView.
- Coloca el foco en el campo de Nombre para facilitar nuevas inserciones.

btnClick

fun btnClick(view: View)

Qué hace:

- Determina si se debe agregar un nuevo alumno (insertData) o actualizar uno existente (updateData) según el mensaje recibido ("add" o "edit").

Flujo de Uso

Agregar un Alumno:

Se selecciona la opción "Agregar".

La actividad muestra campos vacíos.

El usuario ingresa los datos y guarda.

Los campos se limpian automáticamente y el foco se coloca en Nombre.(esto es para que siga agregando mas alumnos el usuario)

Editar un Alumno:

Se selecciona un alumno existente para editar.

La actividad muestra los datos actuales del alumno.

El usuario realiza cambios y guarda.

La aplicación regresa automáticamente al menú principal.

Eliminar un Alumno:

Se selecciona un alumno existente para eliminar.

La aplicación elimina el registro y regresa al menú principal.

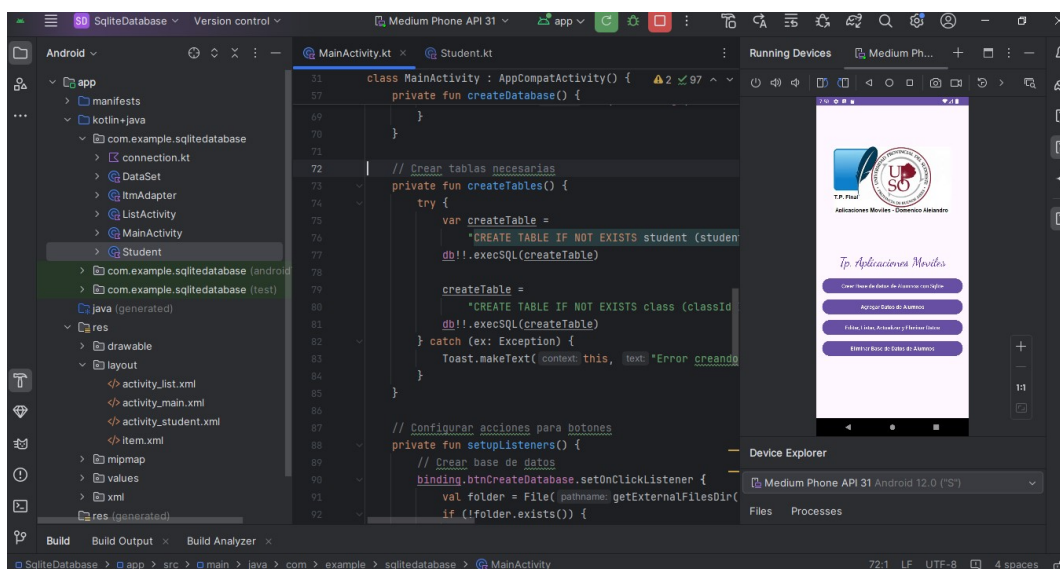
Cargar una Imagen:

Durante la edición o creación, el usuario puede seleccionar una imagen desde el almacenamiento del dispositivo.

La imagen se almacena en la base de datos y se muestra en la actividad.

Un poco del manejo de Errores

- Si la base de datos no existe, la actividad muestra un mensaje y se cierra.
- Valida que los campos obligatorios estén completos antes de realizar cualquier operación.
- Captura y muestra errores relacionados con la base de datos o la carga de imágenes.



Link del video con la explicación: <https://youtu.be/iH9iy2or9GY>