¡Por supuesto! Aquí tienes el **texto completo y extenso para tu README/documentación**. Puedes copiarlo a Word, Google Docs o donde prefieras y adaptarlo a PDF si lo necesitas.

App Registro de Estudiantes - II Corte POO2

Descripción General

Esta aplicación móvil ha sido desarrollada como parte de la Evaluación del II Corte para la asignatura **Programación Orientada a Objetos II** en la Universidad Americana (UAM). El objetivo del proyecto es aplicar los principios de la programación orientada a objetos, el patrón de arquitectura MVVM, la persistencia local de datos usando Room, y la interfaz moderna con Jetpack Compose.
La app permite registrar, validar y mostrar información básica de estudiantes, garantizando la persistencia de datos entre sesiones y la implementación de buenas prácticas de desarrollo de software.

Funcionalidades Principales

- **Registro de Estudiantes:** El usuario puede ingresar el nombre, carrera y edad de un estudiante a través de campos de texto.
- **Validaciones de Entrada:**
    - **Nombre y carrera:** Solo permite letras, tildes y espacios. Se bloquean números y símbolos especiales, evitando errores comunes de digitación.
    - **Edad:** Solo acepta números enteros en el rango de 15 a 100 años. No se permiten edades negativas ni valores vacíos.
    - Los errores se muestran con mensajes claros y en color rojo justo debajo del campo correspondiente.
- **Persistencia de Datos:** Todos los estudiantes ingresados se guardan en una base de datos local usando Room. Los datos persisten aunque se cierre y vuelva a abrir la aplicación.
- **Visualización Dinámica:** La lista de estudiantes se actualiza en tiempo real conforme se agregan nuevos registros.
- **Arquitectura Moderna (MVVM):** Separación de responsabilidades, facilitando el mantenimiento, pruebas y escalabilidad de la app.

- **Interfaz Centrada y Moderna:** Uso de Jetpack Compose para una experiencia de usuario fluida y amigable. Los campos y la lista están centrados para mejor visualización.

Estructura del Proyecto

- **data/**
  - `Estudiante.kt`: Entidad de la base de datos con los atributos requeridos.
  - `EstudianteDao.kt`: DAO con métodos para insertar y obtener estudiantes.
  - `EstudianteDatabase.kt`: Singleton de la base de datos Room.
- **repository/**
  - `EstudianteRepository.kt`: Encapsula la lógica de acceso a datos y expone métodos para el ViewModel.
- **viewmodel/**
  - `EstudianteViewModel.kt`: Gestiona el estado de la UI, invoca el repositorio y maneja las operaciones en segundo plano.
  - `ViewModelFactory.kt`: Permite la creación personalizada del ViewModel.
- **data/ui/**
  - `MainScreen.kt`: Composable principal con los campos de ingreso, validaciones y lista de estudiantes.
- **ui.theme/**
  - Definiciones de color, fuente y tema visual de la app.

Flujo de Uso

1. **Ingreso de datos:** El usuario escribe el nombre, carrera y edad del estudiante.

2. **Validación:** Al presionar “Guardar”, se valida cada campo. Si algún dato es incorrecto, se muestra el error correspondiente en rojo.

3. **Almacenamiento:** Si todos los datos son válidos, se guarda el registro en la base de datos Room.

4. **Visualización:** La lista de estudiantes registrados aparece justo debajo, mostrando nombre, carrera y edad.

5. **Persistencia:** Si se cierra y vuelve a abrir la app, la lista se mantiene gracias a Room.

Tecnologías y Herramientas Utilizadas

- **Lenguaje:** Kotlin
- **UI:** Jetpack Compose (Material 3)
- **Persistencia:** Room Database
- **Arquitectura:** MVVM (Model-View-ViewModel)
- **IDE:** Android Studio
- **Control de versiones:** Git y GitHub (con mínimo 3 commits significativos)
- **Sistema operativo objetivo:** Android

Buenas Prácticas

- **Separación clara de capas:** Se sigue estrictamente la arquitectura MVVM para mejorar el mantenimiento y la escalabilidad.
- **Validación y control de errores:** Cada campo tiene su propia validación y muestra mensajes de error detallados.
- **Uso eficiente de corrutinas y Flow:** Para operaciones asincrónicas y manejo de estados en tiempo real.
- **Código limpio y documentado:** Clases, funciones y variables con nombres descriptivos.

Colaboradores

- [José López]
- [Oscar Calero
- [Nelson Lacayo]
- [Carlos Avalos]

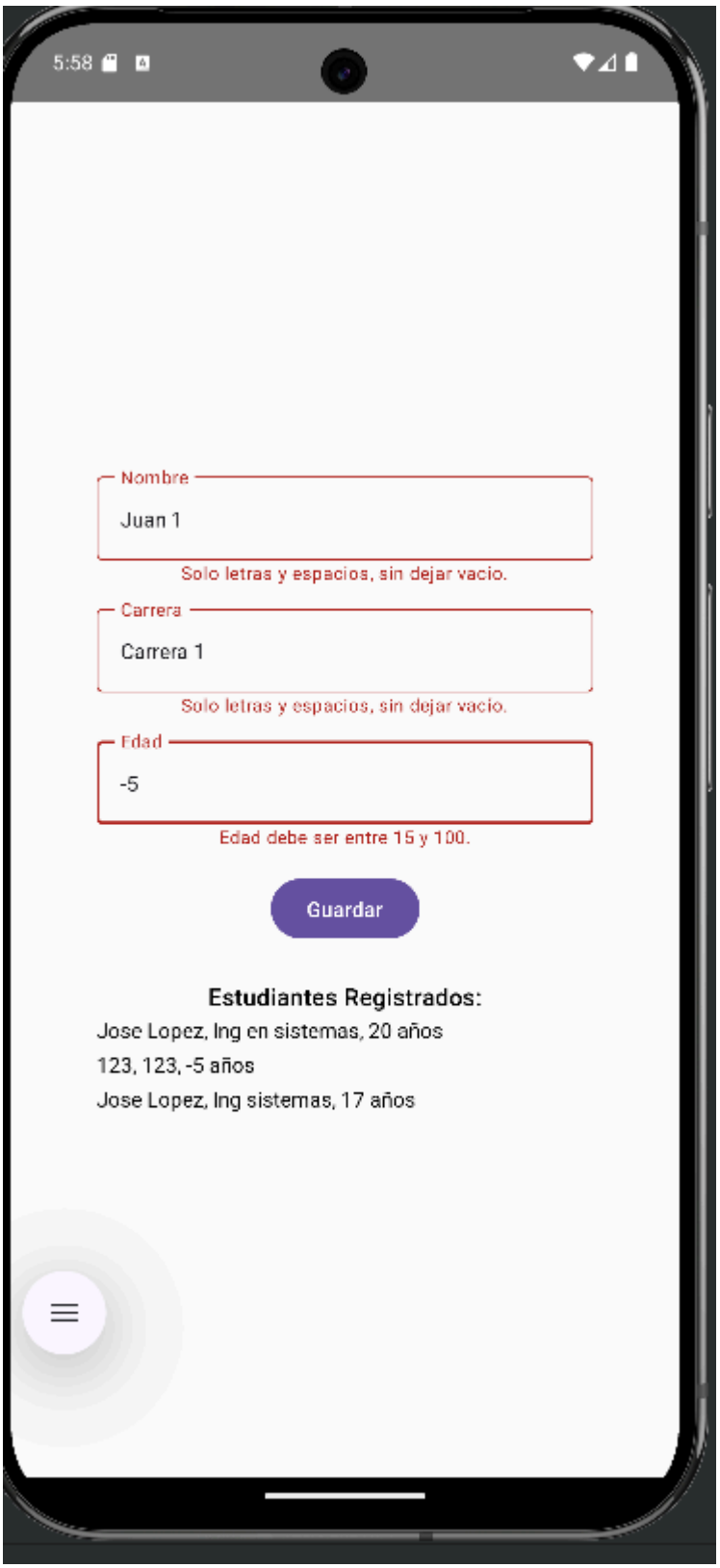
Profesor

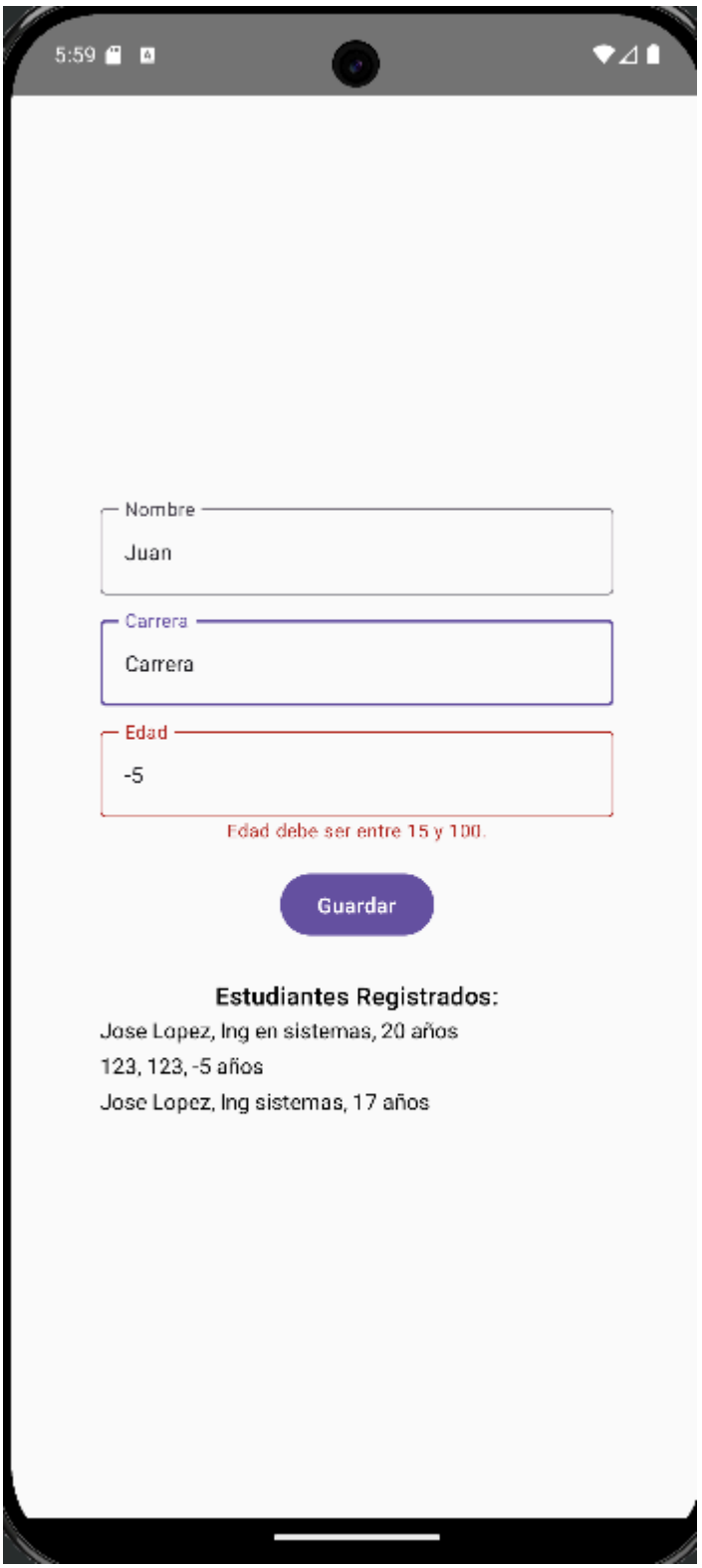
José A. Durán G.
Facultad de Ingeniería y Arquitectura
Universidad Americana (UAM)

Observaciones Finales

Esta app demuestra la integración de técnicas modernas de desarrollo móvil en Android, combinando una arquitectura robusta con una experiencia de usuario fluida y validaciones eficientes.
Es un ejemplo ideal de cómo estructurar proyectos escalables y mantenibles para el desarrollo profesional de software.

Imágenes:


5:58
Nombre
Juan 1
Solo letras y espacios, sin dejar vacio.
Carrera
Carrera 1
Solo letras y espacios, sin dejar vacio.
Edad
-5
Edad debe ser entre 15 y 100.
Guardar
Estudiantes Registrados:
Jose Lopez, Ing en sistemas, 20 años
123, 123, -5 años
Jose Lopez, Ing sistemas, 17 años


5:59
Nombre
Juan
Carrera
Carrera
Edad
-5
Edad debe ser entre 15 y 100.
Guardar
Estudiantes Registrados:
Jose Lopez, Ing en sistemas, 20 años
123, 123, -5 años
Jose Lopez, Ing sistemas, 17 años