

PROYECTO FINAL REGISTRO DE ASISTENCIA A GIMNASIO

INSTRUCCIONES GENERALES

Curso: EIF-411 DISEÑO Y PROGRAMACIÓN DE PLATAFORMAS MÓVILES, Grupo 21

Profesor: MATI Esteban González Sandoval

Fecha de entrega: 17 de junio del 2024

Modalidad de trabajo: En parejas

Entrega: Formato ZIP en el aula virtual

Presentación: Preparar una presentación para defender el proyecto

Duración de la presentación: 10 minutos

INTRODUCCIÓN

En el marco del curso Programación y Diseño de Aplicaciones Móviles, la persona estudiante llevará a cabo el desarrollo de una aplicación móvil denominada Registro de Asistencia a Gimnasio. Este proyecto final tiene como propósito integrar y aplicar los conocimientos adquiridos durante el curso, brindando una experiencia práctica en el diseño y la implementación de aplicaciones móviles modernas utilizando la tecnología de Flutter.

Objetivo de la Aplicación

La aplicación Registro de Asistencia a Gimnasio está diseñada para optimizar la gestión de cursos y el control de asistencia en un entorno de gimnasio. A través de esta herramienta, los administradores podrán gestionar de manera eficiente los cursos ofrecidos, mientras que los usuarios podrán registrar su asistencia de forma sencilla y precisa. Esta aplicación busca mejorar la organización y el seguimiento de la participación en las actividades del gimnasio, garantizando una experiencia de usuario fluida y segura.

Descripción de la Aplicación

La aplicación se estructura en diversas secciones, cada una destinada a cumplir funciones específicas:

1. Pantalla de Ingreso:

- Proporciona una interfaz para que los usuarios inicien sesión utilizando su correo electrónico y una contraseña segura.
- La contraseña debe cumplir con los criterios de seguridad, indicados en la sección correspondiente.
- La pantalla valida las credenciales y muestra mensajes de error en caso de que los datos no coincidan.

2. Pantalla Principal del Cliente:

- Al iniciar sesión, los usuarios con el rol de cliente visualizarán los cursos en los que están matriculados.
- Al seleccionar un curso, se generará y mostrará un código QR único, creado mediante un identificador UUID, que permitirá registrar la asistencia del usuario a dicho curso.

3. Pantalla Principal del Administrador:

- Al iniciar sesión, los usuarios con el rol de administrador accederán a una interfaz que les permitirá gestionar los cursos del gimnasio.
- Esta pantalla incluirá opciones para agregar, modificar y eliminar cursos.

4. Pantalla de Registro de Asistencia:

- Accesible mediante la selección de un curso en la lista de la pantalla principal.
- Muestra la lista de participantes matriculados en el curso.
- Permite el registro de asistencia mediante un switch, ofreciendo opciones de asistencia: presente, ausente y excluido.
- Implementa una funcionalidad que excluye automáticamente a la persona estudiante con tres ausencias consecutivas.

5. Sección de Administración de Cursos:

- Dirigida a los administradores, permite agregar, modificar y eliminar cursos.
- El formulario para agregar cursos incluye campos esenciales como id, nombre, fecha, hora y capacidad.
- Incorpora validaciones que impiden la eliminación de cursos que tengan estudiantes matriculados, asegurando la integridad de los datos.

6. Registro de Usuarios:

- Facilita la creación de nuevas cuentas de usuario mediante un formulario de registro.
- El formulario incluye campos para nombre completo, correo electrónico, contraseña, confirmación de contraseña y selección de rol.
- Se distinguen dos roles de usuario: administrador y cliente, cada uno con diferentes niveles de acceso y funcionalidades dentro de la aplicación.

PANTALLA DE INGRESO A LA APLICACIÓN

1. Introducción

En esta sección, la persona estudiante deberá crear la pantalla de ingreso a la aplicación. Esta pantalla permitirá a los usuarios ingresar utilizando su correo electrónico y una clave. Se implementarán validaciones tanto para el correo electrónico como para la clave, y se manejan errores en caso de que los datos no coincidan.

2. Requisitos

- **Correo electrónico:** El usuario debe ingresar un correo electrónico válido.
- **Clave:** La clave debe tener un mínimo de 8 caracteres, contener letras, números y al menos un carácter especial.

- **Botón de Ingreso:** El botón debe estar etiquetado como "Ingresar". Mientras se verifica la información, debe mostrar un indicador de "cargando" y deshabilitarse para evitar múltiples envíos.
- **Mensajes de Error:** Si la información ingresada no coincide, debe mostrarse un mensaje informando que el correo o la clave no coinciden.

3. Pasos para Implementar la Pantalla de Ingreso

1. Crear el Proyecto:

- **Nombre del proyecto:** mygym_app.
- Configura el entorno de desarrollo de Flutter e inicia un nuevo proyecto con este nombre.

2. Diseño de la Interfaz:

- **Formulario de Ingreso:** La pantalla debe contener un formulario con los siguientes campos:
 - Correo Electrónico: Campo de texto para ingresar el correo electrónico, con validaciones para asegurar que es un correo válido.
 - Clave: Campo de texto para ingresar la clave, con validaciones para asegurar que cumple con los requisitos mínimos.
 - Botón de Ingreso: Botón etiquetado como "Ingresar" que, al ser presionado, valida los campos y procesa la solicitud.

3. Validación de Datos:

- Correo Electrónico: Validar que no esté vacío y que tenga el formato de correo electrónico.
- Clave: Validar que tenga al menos 8 caracteres, los cuales incluyen letras, números y al menos un carácter especial.

4. Indicador de Carga:

- Mientras se verifica la información ingresada, muestra un indicador de "cargando" y deshabilita el botón de ingreso para evitar múltiples envíos.

5. Manejo de Errores:

- Si la información ingresada no es correcta, muestra un mensaje de error indicando que el correo o la clave no coinciden.

4. Evaluación

- Funcionalidad: La pantalla de ingreso debe funcionar correctamente, validando los datos ingresados y mostrando mensajes de error adecuados.
- Interfaz de Usuario: La interfaz debe ser intuitiva y fácil de usar.
- Código: El código debe estar bien estructurado y documentado.

PANTALLA DE LISTA DE CURSOS

1. Introducción

En esta sección, la persona estudiante deberá a crear la pantalla que se presenta luego del ingreso exitoso del usuario con rol de administrador. Esta pantalla mostrará una lista de los diferentes cursos que ofrece el gimnasio para registrar la asistencia.

2. Requisitos

- Lista de Cursos: La pantalla debe presentar una lista de cursos disponibles en el gimnasio para registrar su asistencia.
- Diseño intuitivo: La interfaz debe ser clara y fácil de navegar.
- Navegación: Debe haber una forma de navegar a la lista de clientes de cada curso desde la lista principal.

3. Pasos para Implementar la Pantalla de Lista de Cursos

1. Diseño de la Interfaz:

- Lista de Cursos: Utiliza un componente widget de lista (como ejemplo: `ListView`) para mostrar los cursos.
- Elementos de la Lista: Cada elemento de la lista debe mostrar información relevante del curso, como el nombre, y su horario.

2. Navegación:

- Detalles del Curso: Implementar la navegación desde cada elemento de la lista a una pantalla de detalles del curso. Esta pantalla debe proporcionar información más detallada sobre el curso seleccionado, mostrando la lista de clientes matriculados.

4. Evaluación

- Funcionalidad: La lista de cursos debe mostrarse correctamente, y la navegación a los detalles del curso debe funcionar sin problemas.
- Interfaz de Usuario: La interfaz debe ser intuitiva y proporcionar una experiencia de usuario agradable.
- Código: El código debe estar bien estructurado y documentado.

Pantalla de Asistencia a Curso (lista de matriculados)

1. Introducción

En esta sección, la persona estudiante deberá crear la pantalla de asistencia a un curso, la cual se presenta al seleccionar un curso en la lista de cursos. Esta pantalla mostrará la lista de personas matriculadas en el curso y permitirá registrar la asistencia para el día correspondiente mediante un widget switch .

2. Requisitos

- Lista de Personas Matriculadas: La pantalla debe mostrar una lista de las personas matriculadas en el curso seleccionado.
- Registro de Asistencia: La asistencia debe registrarse mediante un switch.
- Opciones de Asistencia: Las opciones de asistencia deben ser: presente, ausente y excluido.
- Exclusión Automática: Si un estudiante registra 3 ausencias consecutivas, debe ser excluido automáticamente del curso.
- Diseño Intuitivo: La interfaz debe ser clara y fácil de usar.

3. Pasos para Implementar la Pantalla de Asistencia a Curso

1. Diseño de la Interfaz:

- Lista de Personas Matriculadas: Utiliza un componente de lista (como `ListView`) para mostrar los nombres de las personas matriculadas.
- Registro mediante switch: Añade una funcionalidad para registrar la asistencia.

2. Registro de Asistencia:

- Registro de asistencia: Implementa la lógica para registrar una asistencia. Se debe actualizar su asistencia al estado presente. El estado por defecto es ausente o excluido según corresponda.
- Visualización del Estado: La lista debe reflejar visualmente el estado de asistencia del estudiante (por ejemplo, cambiando el color del nombre o mostrando un ícono).

3. Exclusión Automática:

- Implementa una funcionalidad que registre las ausencias consecutivas de cada estudiante.
- Si un estudiante registra 3 ausencias consecutivas, debe ser excluido automáticamente del curso. Actualiza el estado de asistencia y refleja visualmente el cambio.

4. Evaluación

- Funcionalidad: La lista de personas matriculadas debe mostrarse correctamente, y el registro de asistencia mediante QR debe funcionar sin problemas. La exclusión automática debe activarse después de 3 ausencias consecutivas.

- **Interfaz de Usuario:** La interfaz debe ser intuitiva y proporcionar una experiencia de usuario agradable.
- **Código:** El código debe estar bien estructurado y documentado.

Pantalla de Administración de Cursos

1. Introducción

En esta sección, la persona estudiante deberá crear la pantalla de administración de cursos brindados por el gimnasio. Esta pantalla permitirá a los administradores agregar, editar y eliminar cursos, con validaciones específicas.

2. Requisitos

- **Lista de Cursos:** La pantalla debe mostrar una lista de todos los cursos actuales ofrecidos por el gimnasio.
- **Agregar Curso:** Debe haber una opción para agregar un nuevo curso.
- **Editar Curso:** Cada curso en la lista debe tener una opción para editar sus detalles.
- **Eliminar Curso:** Cada curso en la lista debe tener una opción para eliminarlo, con una validación para evitar la eliminación de cursos con estudiantes matriculados.
- **Diseño Intuitivo:** La interfaz debe ser clara y fácil de usar.

3. Pasos para Implementar la Pantalla de Administración de Cursos

1. Diseño de la Interfaz:

- **Lista de Cursos:** Utiliza un componente de lista (como `ListView`) para mostrar los cursos actuales.
- **Opciones de Administración:** Añade botones para agregar, editar y eliminar cursos.

2. Agregar Curso:

- Formulario de Agregar Curso: Crea un formulario que permita ingresar los detalles del nuevo curso: id, nombre, fecha, hora y capacidad de matriculados.
- Validaciones: Implementa validaciones para asegurar que los datos ingresados sean correctos.

3. Editar Curso:

- Formulario de Editar Curso: Permite modificar los datos del curso.
- Guardar Cambios: Implementa la funcionalidad para guardar los cambios realizados.

4. Eliminar Curso:

- Confirmación de Eliminación: Al seleccionar eliminar, muestra una confirmación para evitar eliminaciones accidentales.
- Eliminar Curso: Implementa la funcionalidad para eliminar el curso seleccionado, siempre que no tenga estudiantes matriculados.

4. Evaluación

- Funcionalidad: La lista de cursos debe mostrarse correctamente, y las funcionalidades de agregar, editar y eliminar cursos deben funcionar sin problemas, incluyendo la validación para evitar la eliminación de cursos con estudiantes matriculados.
- Interfaz de Usuario: La interfaz debe ser intuitiva y proporcionar una experiencia de usuario agradable.
- Código: El código debe estar bien estructurado y documentado.

PANTALLA DE INGRESO PARA EL ROL DE CLIENTE

1. Introducción

En esta sección, la persona estudiante deberá crear la pantalla de ingreso para los usuarios que tienen el rol de cliente. Esta pantalla mostrará únicamente los cursos en los que el cliente está matriculado y, al seleccionar un curso, se generará y mostrará un código QR único.

2. Requisitos

- Lista de Cursos Matriculados: La pantalla debe mostrar una lista de los cursos en los que el cliente está matriculado.
- Código QR Único: Al seleccionar un curso, se debe generar y mostrar un código QR único para la combinación de usuario y curso.
- Identificador UUID: Utilizar un identificador UUID para generar el código QR único.
- Diseño Intuitivo: La interfaz debe ser clara y fácil de usar.

3. Pasos para Implementar la Pantalla de Ingreso para el Rol de Cliente

1. Diseño de la Interfaz:

- Lista de Cursos Matriculados: Utiliza un componente de lista (como `ListView`) para mostrar los cursos en los que el cliente está matriculado.
- Código QR: Añade una vista para mostrar el código QR al seleccionar un curso.

2. Generación de la Lista de Cursos:

- Datos del Usuario: Obtener los datos del usuario logueado y listar los cursos en los que está matriculado.
- Componente de Lista: Implementar la lista para mostrar los nombres de los cursos.

3. Generación del Código QR:

- Identificador UUID: Generar un identificador UUID para cada combinación de usuario y curso.
- Biblioteca de Código QR: Utilizar una biblioteca adecuada para generar el código QR a partir del UUID.
- Mostrar Código QR: Mostrar el código QR en la pantalla cuando el usuario selecciona un curso.

4. Evaluación

- **Funcionalidad:** La lista de cursos debe mostrarse correctamente y el código QR debe generarse y mostrarse adecuadamente al seleccionar un curso.
- **Interfaz de Usuario:** La interfaz debe ser intuitiva y proporcionar una experiencia de usuario agradable.
- **Código:** El código debe estar bien estructurado y documentado.

Tecnologías adicionales requeridas

Para completar el proyecto, se requiere el uso de las siguientes tecnologías adicionales:

1. **Backend con Strapi:** El backend de la aplicación debe estar desarrollado utilizando Strapi, un sistema de gestión de contenido de código abierto y flexible que permite la creación rápida de API.
2. **Almacenamiento del token del usuario con ISAR:** El token del usuario, utilizado para la autenticación en el backend, debe almacenarse en el dispositivo móvil utilizando una base de datos local. Se recomienda utilizar ISAR, una base de datos local rápida y segura para Flutter.
3. **Gestor de estado con Provider:** Para gestionar el estado de la aplicación de forma eficiente y escalable, se recomienda utilizar Provider, un paquete de gestión de estado simple y fácil de usar para Flutter.

Estas tecnologías adicionales son fundamentales para el funcionamiento correcto y seguro de la aplicación móvil de Registro de Asistencia a Gimnasio.