asistenciacontrol

A new Flutter project.

EXAMEN PRÁCTICO – UNIDAD III

- **Curso:** Desarrollo de Aplicaciones Móviles
- **Fecha:** [PON AQUÍ LA FECHA DE ENTREGA, EJ: 27/05/2024]
- **Estudiante:** Brayar Lopez Catunta

URL del Repositorio

El proyecto se encuentra alojado en el siguiente repositorio público de GitHub:

https://github.com/BCZLopezCatunta/SM2_ExamenUnidad3Lopez

Evidencias y Documentación

A continuación, se presentan las capturas de pantalla que evidencian el cumplimiento de los objetivos del examen.

1. Captura de la Estructura de Carpetas

La siguiente imagen muestra la estructura de carpetas requerida por el examen, destacando la creación de los directorios `.github/workflows/` y `test/` en la raíz del proyecto.

> **[INSERTA AQUÍ LA CAPTURA DE PANTALLA DE LA ESTRUCTURA DE CARPETAS]** *(Debe mostrar claramente las carpetas `.github` y `test`)*

2. Captura del Contenido del Archivo `quality-check.yml`

Este es el contenido del workflow de GitHub Actions. El flujo de trabajo está configurado para ejecutarse automáticamente y realizar análisis de calidad sobre el código fuente.

> **[INSERTA AQUÍ LA CAPTURA DE PANTALLA DEL CÓDIGO DEL ARCHIVO `quality-check.yml`]**

3. Captura de la Ejecución Exitosa del Workflow en "Actions"

La imagen a continuación evidencia la ejecución automática y exitosa (100% Passed) del workflow "Quality Check". Esta acción se disparó tras realizar un `push` a la rama `main`, y se puede observar que todos los pasos, incluyendo el análisis de código (`analyze`) y la ejecución de pruebas (`test`), se completaron correctamente con una marca de verificación verde.

> **[INSERTA AQUÍ LA CAPTURA DE PANTALLA DE LA ACCIÓN EXITOSA (CON EL CHECK VERDE ✅)]**
> *(Asegúrate de que se vea el nombre del workflow y los pasos completados)*

Explicación de lo Realizado

Para cumplir con los requisitos del examen, se ha implementado un flujo de trabajo de **Integración Continua (CI)** utilizando **GitHub Actions**. El objetivo principal es automatizar las revisiones de calidad del código para un proyecto móvil desarrollado en Flutter.

El proceso se llevó a cabo de la siguiente manera:

1. **Configuración del Repositorio:** Se creó y configuró la estructura de directorios solicitada. Se añadió la carpeta `.github/workflows/` para alojar el archivo de definición del workflow y la carpeta `test/` para contener las pruebas unitarias.

2. **Implementación de Pruebas Unitarias:** En el archivo `test/main_test.dart`, se definieron tres pruebas unitarias simples. Estas pruebas verifican lógica básica y sirven para validar que el comando `flutter test` del workflow se ejecute correctamente.

3. **Creación del Workflow (`quality-check.yml`):** Se definió un workflow que se activa automáticamente ante dos eventos clave:

    - `push`: Cada vez que se suben cambios a la rama `main`.
    - `pull_request`: Cada vez que se crea una solicitud de fusión hacia la rama `main`.

4. **Pasos del Flujo de Trabajo Automatizado:** El workflow ejecuta una serie de pasos secuenciales en un entorno virtual (`ubuntu-latest`) proporcionado por GitHub:

    - **Checkout:** Descarga el código fuente del repositorio para que los siguientes pasos puedan trabajar con él.
    - **Set up Flutter:** Instala y configura la versión correcta del SDK de Flutter, garantizando un entorno consistente. Se especificó una versión compatible con el proyecto para evitar conflictos de dependencias.
    - **Install dependencies:** Ejecuta el comando `flutter pub get` para descargar todas las librerías y paquetes de los que depende el proyecto.
    - **Analyze:** Lanza el comando `flutter analyze`, una potente herramienta de análisis estático que revisa el código en busca de errores, advertencias y violaciones de las guías de estilo de Dart. Esto asegura que el código esté limpio y siga las buenas prácticas.
    - **Run tests:** Ejecuta el comando `flutter test`, que corre automáticamente todas las pruebas definidas en la carpeta `test/`. Esto es crucial para confirmar que los cambios recientes no han roto ninguna funcionalidad existente.

La implementación de este pipeline de CI asegura que cada contribución al código base sea validada automáticamente, mejorando significativamente la calidad, la estabilidad y la mantenibilidad del proyecto, lo cual es una práctica fundamental en la cultura **DevOps**.