Examen Práctico Unidad 3 - Automatización de Calidad con GitHub Actions

---

1. Datos del Estudiante

- **Curso:** Soluciones Móviles II
- **Fecha:** 27 de Junio de 2024
- **Nombre Completo:** Ricardo Daniel Cutipa Gutierrez

---

2. URL del Repositorio Público

El código fuente y la configuración de este examen se encuentran en el siguiente repositorio de GitHub:

https://github.com/RicardoCutipa/SM2_ExamenUnidad3

---

3. Evidencias del Proceso

A continuación, se presentan las capturas de pantalla que documentan la implementación del flujo de trabajo de CI/CD.

3.1. Estructura de Carpetas

La siguiente imagen muestra la estructura de directorios requerida (`.github/workflows/`) creada en la raíz del repositorio para alojar el archivo de configuración del workflow.

[PEGA AQUÍ TU CAPTURA DE PANTALLA DE LA ESTRUCTURA DE CARPETAS]

Ejemplo de cómo se vería el enlace una vez que subas tu imagen: `![Estructura de Carpetas](https://raw.githubusercontent.com/RicardoCutipa/SM2_ExamenUnidad3/main/docs/estructura.png)`

3.2. Contenido del Archivo `quality-check.yml`

El archivo `quality-check.yml` define los pasos que GitHub Actions ejecuta automáticamente. Este workflow se activa en cada `push` o `pull_request` a la rama `main` y realiza análisis de calidad y pruebas unitarias sobre el código Flutter.

[PEGA AQUÍ TU CAPTURA DE PANTALLA DEL CÓDIGO YML]

Ejemplo de cómo se vería el enlace una vez que subas tu imagen: `![Contenido del Workflow](https://raw.githubusercontent.com/RicardoCutipa/SM2_ExamenUnidad3/main/docs/workflow.png)`

3.3. Ejecución Exitosa del Workflow en "Actions"

La captura de pantalla a continuación evidencia la ejecución exitosa (100% Passed) del workflow "Quality Check" en la pestaña "Actions" del repositorio. Se puede observar que todos los pasos, incluyendo `analyze` y `test`, se completaron sin errores.

[PEGA AQUÍ TU CAPTURA DE PANTALLA DE LA EJECUCIÓN EN ACTIONS]

Ejemplo de cómo se vería el enlace una vez que subas tu imagen: `![Ejecución Exitosa del Workflow](https://raw.githubusercontent.com/RicardoCutipa/SM2_ExamenUnidad3/main/docs/actions.png)`

---

4. Explicación de lo Realizado

El objetivo de este examen fue implementar un flujo de trabajo de Integración Continua (CI) utilizando **GitHub Actions** para automatizar el análisis de calidad de un proyecto móvil desarrollado en Flutter, integrando así prácticas de DevOps en el ciclo de desarrollo.

El proceso se puede resumir en los siguientes pasos:

1. **Configuración del Repositorio:** Se creó un repositorio público en GitHub llamado `SM2_ExamenUnidad3` y se subió el código fuente del proyecto móvil.
2. **Creación del Workflow:** Se definió la estructura de directorios `.github/workflows/` y se creó el archivo `quality-check.yml`. Este archivo contiene las instrucciones para el workflow, que se activa automáticamente ante eventos de `push` y `pull request` en la rama `main`.
3. **Definición de las Pruebas:** Se implementaron **3 pruebas unitarias** en el archivo `test/main_test.dart`. Estas pruebas se enfocaron en validar la lógica pura de las funciones de validación de formularios (email, contraseña, etc.), asegurando que son robustas y funcionan de manera aislada.
4. **Ejecución del Flujo de Trabajo:** El workflow configurado ejecuta una serie de trabajos en un entorno de `ubuntu-latest`:
    - **Checkout:** Clona el código del repositorio.
    - **Set up Flutter:** Configura la versión específica de Flutter del proyecto.
    - **Install dependencies:** Ejecuta `flutter pub get` para instalar todas las dependencias.
    - **Analyze:** Lanza `flutter analyze` para revisar el código en busca de errores, advertencias y problemas de estilo, garantizando que se cumplan las buenas prácticas de Dart.
    - **Run tests:** Ejecuta `flutter test` para correr las pruebas unitarias y verificar que la lógica principal de la aplicación no se ha roto.
5. **Verificación y Resultados:** Tras subir los cambios, se confirmó en la pestaña "Actions" de GitHub que el workflow se ejecutó correctamente y todos los pasos pasaron con éxito, lo que demuestra la calidad y estabilidad del código.

Este ejercicio permitió automatizar tareas repetitivas de control de calidad, facilitando la detección temprana de errores y asegurando que cada cambio integrado mantenga la integridad del proyecto.