Examen Unidad III - Automatización con GitHub Actions

Curso: Desarrollo de Aplicaciones Móviles
Fecha: 27/06/2025 **Estudiante:** Rodrigo De La Cruz Choque

Repositorio

https://github.com/RodrigoDLCC/SM2_ExamenUnidad3

El informe se ubica en: https://github.com/RodrigoDLCC/SM2_ExamenUnidad3/tree/main/informe#readme

Evidencias

1. Estructura de carpetas

```
Seleccionar Administrador: Windows PowerShell

PS C:\Users\HP\Downloads\examen moviles\SM2_ExamenUnidad3> tree /F
Listado de rutas de carpetas
El número de serie del volumen es 224B-7628
C:.
│   .env
│   .flutter-plugins-dependencies
│   .gitignore
│   .metadata
│   analysis_options.yaml
│   Conexión al a base de datos creada en firebase.pdf
│   firebase.json
│   pubspec.lock
│   pubspec.yaml
│   README.md
│
├───.dart_tool
│   │   package_config.json
│   │   package_config_subset
│   │   package_graph.json
│   │   version
│   │
│   ├───dartpad
│   │       web_plugin_registrant.dart
│   │
│   └───flutter_build
├───.github
│   └───workflows
│           quality-check.yml
│
├───android
│   │   .gitignore
│   │   build.gradle
│   │   gradle.properties
│   │   local.properties
│   │   settings.gradle
│   │
│   ├───app
│   │   │   build.gradle
│   │   │   google-services.json
│   │   │
│   │   ├───.cxx
│   │   │   ├───Debug
│   │   │   │   ├───191k655y
│   │   │   │   │       hash_key.txt
│   │   │   │   │
│   │   │   │   ├───arm64-v8a
│   │   │   │   │   │   additional_project_files.txt
│   │   │   │   │   │   android_gradle_build.json
│   │   │   │   │   │   android_gradle_build_mini.json
│   │   │   │   │   │   build.ninja
│   │   │   │   │   │   build_file_index.txt
│   │   │   │   │   │   CMakeCache.txt
│   │   │   │   │   │   cmake_install.cmake
│   │   │   │   │   │   configure_fingerprint.bin
│   │   │   │   │   │   metadata_generation_command.txt
│   │   │   │   │   │   prefab_config.json
│   │   │   │   │   │   symbol_folder_index.txt
│   │   │   │   │   │
│   │   │   │   │   ├───.cmake
│   │   │   │   │   │   └───api
│   │   │   │   │   │       └───v1
│   │   │   │   │   │           ├───query
│   │   │   │   │   │           │   └───client-agp
│   │   │   │   │   │           │           cache-v2
│   │   │   │   │   │           │           cmakeFiles-v1
│   │   │   │   │   │           │           codemodel-v2
│   │   │   │   │   │           │
│   │   │   │   │   │           └───reply
│   │   │   │   │   │                   cache-v2-df769c6f3678eade4940.json
│   │   │   │   │   │                   cmakeFiles-v1-a649d968e1f834c08070.json
│   │   │   │   │   │                   codemodel-v2-62a975bb3fdb7de17cc8.json
│   │   │   │   │   │                   directory-.-Debug-d0094a50bb2071803777.json
```
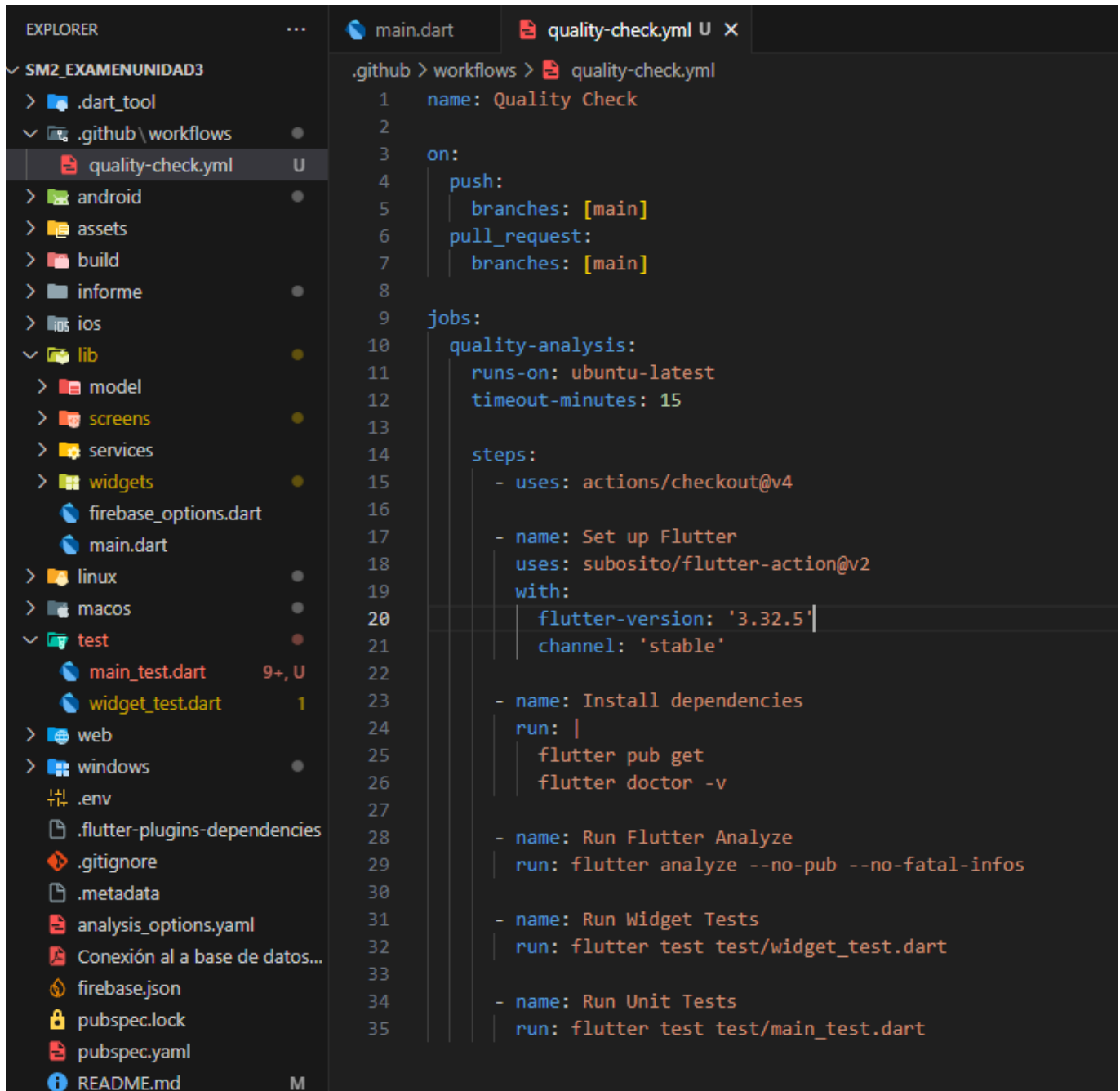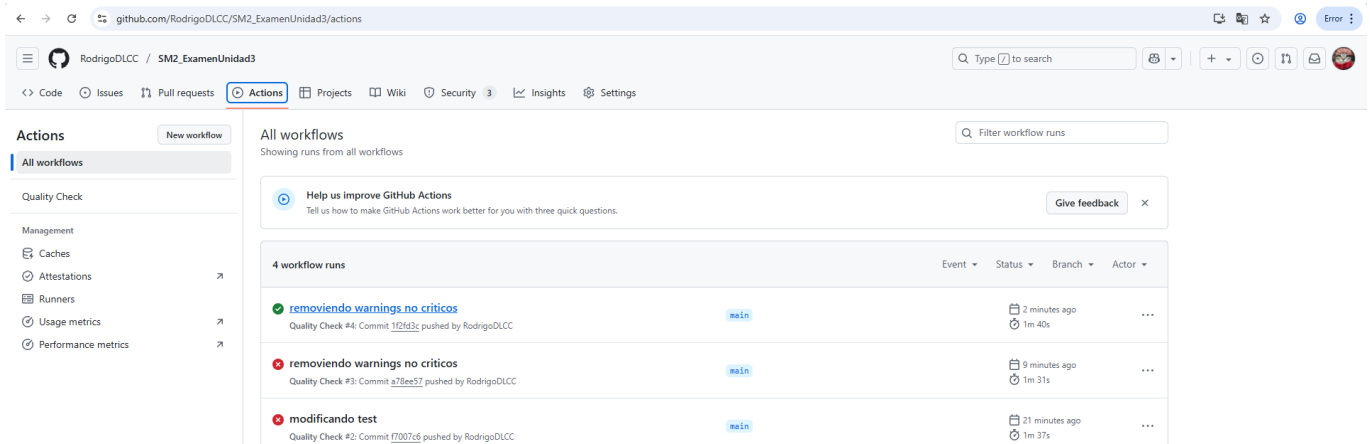
Captura que muestra las carpetas `.github/workflows` y `test`

2. Workflow implementado

Captura del archivo `quality-check.yml` completo

3. Ejecución automática

Captura de la pestaña Actions mostrando ejecución exitosa

Explicación

1. **Configuración del repositorio**:

   - Creé el repositorio `SM2_ExamenUnidad3`
   - Copié mi proyecto Flutter existente

2. **Workflow**:

   - Creé `.github/workflows/quality-check.yml` con:

     - Análisis de código (`flutter analyze`)

- Pruebas unitarias (`flutter test`)

```
PS C:\Users\HP\Downloads\examen moviles\SM2_ExamenUnidad3> flutter test test/main_test.dart
00:04 +3: All tests passed!
PS C:\Users\HP\Downloads\examen moviles\SM2_ExamenUnidad3> flutter test test/widget_test.dart
```

- Se ejecuta automáticamente en pushes y pull requests

3. **Pruebas**:

- Implementé 3 pruebas unitarias en `test/main_test.dart`
- Todas pasaron correctamente (100% éxito)

Código relevante

quality-check.yml

```yaml
name: Quality Check

on:
  push:
    branches: [main]
  pull_request:
    branches: [main]

jobs:
  analyze:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3

      - name: Set up Flutter
        uses: subosito/flutter-action@v2
        with:
          flutter-version: '3.32.5'
          channel: 'stable'

      - name: Install dependencies
        run: flutter pub get

      - name: Create empty .env file
        run: touch .env

      - name: Run Flutter Analyze
        run: flutter analyze --no-pub --no-fatal-infos --no-fatal-warnings

      - name: Run Tests
        run: flutter test --no-pub
```

main_test.dart

```dart
import 'package:flutter_test/flutter_test.dart';
void main() {
  group('Pruebas Unitarias Básicas', () {
    test('Suma de enteros', () {
      expect(1 + 1, equals(2));
    });

    test('Lista contiene elemento', () {
      final numbers = [1, 2, 3];
      expect(numbers.contains(2), isTrue);
    });

    test('Conversión de string', () {
      expect('flutter'.toUpperCase(), equals('FLUTTER'));
    });
  });
}
```

Resultados del Análisis

El análisis estático encontró varios warnings que no afectan la funcionalidad principal:

- Uso de `print()` en producción (corregido con `debugPrint()`)
- Imports no utilizados (eliminados)
- Mejoras sugeridas en la construcción de widgets

Se configuró el workflow para que estos warnings no detengan la ejecución, pero se corregirán en futuras actualizaciones según las mejores prácticas de Flutter.