

Conversación con Gemini

Exported: 15/6/2026 9:07:44

Jordi H.

Has dicho estoy intentando que github escanee mi código para hacer un reporte de calidad y probablemente vulnerabilidades. usa el CodeQL suyo. mi proyecto es un backend en python y un frontend en angular. pero me dice: No CodeQL supported languages to scan in this repository

Gemini

Ese error suele ocurrir porque el proceso de **autodiscovery** de GitHub Actions no está detectando correctamente los archivos fuente o porque la configuración por defecto del workflow no está apuntando a los lenguajes que realmente componen tu repositorio. Dado que trabajas con **Python** (Backend) y **TypeScript/JavaScript** (Angular), aquí tienes los pasos para forzar a CodeQL a reconocer y analizar ambos lenguajes de forma efectiva

1. Configuración del Workflow (`.github/workflows/codeql.yml`)

Asegúrate de que tu archivo YAML defina explícitamente la matriz de lenguajes. CodeQL necesita saber que debe buscar tanto Python como JavaScript/TypeScript (para Angular).YAM

CODE

```
name: "CodeQL"

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

jobs:
  analyze:
    name: Analyze
    runs-on: ubuntu-latest
    permissions:
      security-events: write
    actions: read
    contents: read

    strategy:
      fail-fast: false
    matrix:
      # Forzamos el escaneo de ambos mundos
      language: [ 'javascript-typescript', 'python' ]

    steps:
      - name: Checkout repository
        uses: actions/checkout@v4

      - name: Initialize CodeQL
        uses: github/codeql-action/init@v3
        with:
          languages: ${matrix.language}
          # Si tienes el código en carpetas específicas, puedes indicarlo:
          # paths-ignore: ['data/**', 'tests/**']
```

```
- name: Autobuild
uses: github/codeql-action/autobuild@v3

- name: Perform CodeQL Analysis
uses: github/codeql-action/analyze@v3
```

2. Causas comunes del "No languages to scan"

Si tras especificar los lenguajes sigue fallando, revisa estos tres puntos críticos:

A. Estructura de Carpetas

Si tu código está muy "profundo" en el árbol de directorios (ej. proyecto/src/backend/...), a veces el scanner se rinde

- Solución: Asegúrate de que los archivos .py y .ts sean accesibles desde la raíz o usa la propiedad paths en la inicialización de CodeQL

B. El caso de Angular (TypeScript)

CodeQL trata a TypeScript como parte del lenguaje javascript-typescript.

- Nota importante: Para Angular, CodeQL no necesita "compilar" el frontend como tal para el análisis de seguridad, pero sí necesita encontrar el archivo package.json o archivos .ts/.js válidos

C. Repositorios Mono-repo

Si tienes el backend y el frontend en repositorios separados pero los has unido recientemente, verifica que no haya un archivo .gitignore en la raíz que esté ignorando accidentalmente carpetas de código fuente

3. Recomendación para tu Laboratorio (Software Libre)

Como estás montando un vector tecnológico y priorizas el **software libre**, considera integrar también **Bandit** para el análisis específico de Python. Es mucho más rápido que CodeQL para chequeos rápidos de seguridad en el backend. Puedes añadir este paso a tu pipeline de GitHub Actions: YA

CODE

```
- name: Run Bandit (Python Security)
run: |
  pip install bandit
  bandit -r ./tu_carpeta_python -f sarif -o bandit-results.sarif
```

Resumen de acción inmediata:

1. Revisa que en tu archivo .yaml el lenguaje para Angular esté puesto como javascript-typescript.
 2. Verifica que el paso init de CodeQL tenga la matriz de lenguajes configurada.
 3. Si el repositorio es privado, asegúrate de que tienes habilitadas las "GitHub Advanced Security" features en los ajustes del repo (aunque en públicos es gratuito)
-