



**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**LABORATORIO N°01 – Unidad 3**

**Gestión Automatizada De Pruebas Con Github**

**Curso: Calidad y Pruebas de Software**

**Docente: Cuadros Quiroga, Patrick Jose**

**Estudiante:**

**Hurtado Ortiz, Leandro**

**(2015052384)**

**Tacna – Perú**

**2024**

Contenido

SESION DE LABORATORIO N° 01: GESTION AUTOMATIZADA DE PRUEBAS CON GITHUB .....3

OBJETIVOS .....3

REQUERIMIENTOS.....3

CONSIDERACIONES INICIALES.....3

DESARROLLO.....3

Conclusiones.....16

# SESION DE LABORATORIO N° 01: GESTION AUTOMATIZADA DE PRUEBAS CON GITHUB

## OBJETIVOS

- \* Desarrollar la automatización de la gestión de pruebas de una aplicación utilizando Github Actions.

## REQUERIMIENTOS

- \* Conocimientos:
  - Conocimientos básicos de Bash (powershell).
  - Conocimientos básicos de Contenedores (Docker).
  - Conocimientos básicos de lenguaje YAML.
- \* Hardware:
  - Virtualization activada en el BIOS..
  - CPU SLAT-capable feature.
  - Al menos 4GB de RAM.
- \* Software:
  - Windows 10 64bit: Pro, Enterprise o Education (1607 Anniversary Update, Build 14393 o Superior)
  - Docker Desktop
  - Powershell versión 7.x
  - Net 8 o superior
  - Visual Studio Code

## CONSIDERACIONES INICIALES

- \* Clonar el repositorio mediante git para tener los recursos necesarios
- \* Tener una cuenta de Github valida.

## DESARROLLO

1. Abrir un navegador de internet e ingresar a la pagina de SonarCloud (<https://www.sonarsource.com/products/sonarcloud/>), iniciar sesión con su cuenta de Github.

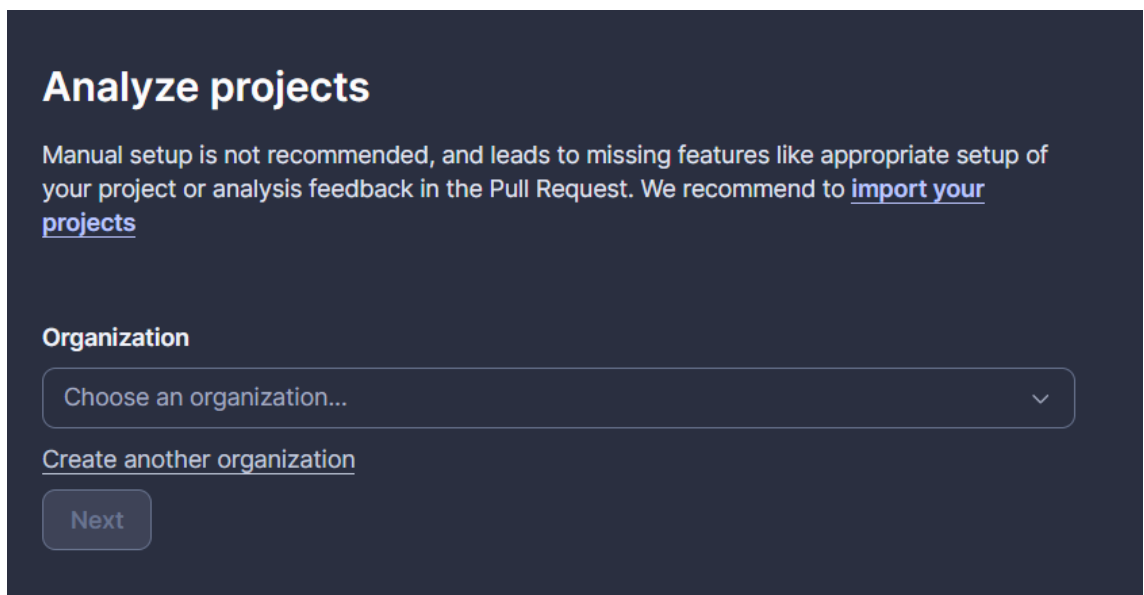
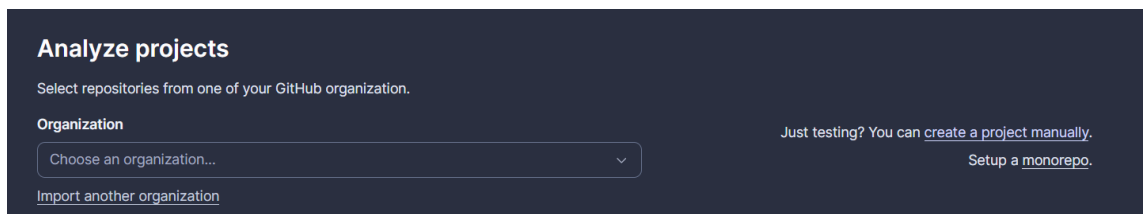
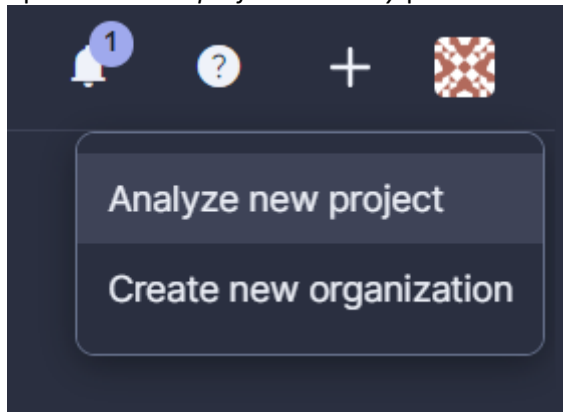
The screenshot shows the SonarCloud dashboard with a sidebar on the left containing filters for Quality Gate, Reliability, and Security. The main area displays three project analysis cards. The first card for 'Leandro Hurtado / idiomify' shows 52 Bugs, 47 Vulnerabilities, 0.0% Hotspots Reviewed, 347 Code Smells, and 19.2% Duplications. The second card for 'Leandro Hurtado / idiomify\_fix1' shows 48 Bugs, 39 Vulnerabilities, 0.0% Hotspots Reviewed, 313 Code Smells, and 18.9% Duplications. The third card for 'Leandro Hurtado / idiomify\_fix2' shows 45 Bugs, 25 Vulnerabilities, 0.0% Hotspots Reviewed, 238 Code Smells, and 17.6% Duplications. A user profile dropdown is visible in the top right corner.

2. En el navegador de internet, en la pagina de SonarCloud, generar un nuevo token con el nombre que desee, luego de generar el token, guarde el resultado en algún archivo o aplicación de notas. Debido a que se utilizará mas adelante.

The screenshot shows the 'Generate Tokens' page in the SonarCloud user interface. It includes a 'Generate Tokens' section with a text input for 'Enter Token Name' and a 'Generate Token' button. A success message states: 'New token "labfinal" has been created. Make sure you copy it now, you won't be able to see it again!'. Below this, the token 'aca0Fc770390f665c1083a67bd7f3faed9983f7c' is displayed. The 'Existing Tokens' section contains a table with the following data:

Name	Last use	Created	Revoke
Analyze "idiomify"	Never	26 April 2024	Revoke
upt2024	Never	18 April 2024	Revoke
labfinal	Never	6 July 2024	Revoke

3. En el navegador de internet, en la pagina de SonarCloud, hacer click en el icono + y luego en la opción *Analyze projects*. En la ventana de Analyze Projects, seleccionar la opción *create a project manually* para crear un proyecto de manera manual.



4. En el navegador de internet, en la pagina de SonarCloud, en la pagina de nuevo proyecto ingresar el nombre *BancaApp*, tomar nota del valor generado en el cuadro Project Key que sera utilizado mas adelante, confirmar la creación del proyecto haciendo click en el boton Next.

### Organization

Leandro Hurtado

Create another organization

### Display Name \* ?

BancaApp

Up to 255 characters

### Project Key \* ?

leandroh98\_bancaapp

Up to 400 characters. All letters, digits, dash, underscore, period or colon.

### Project visibility

☒ Public

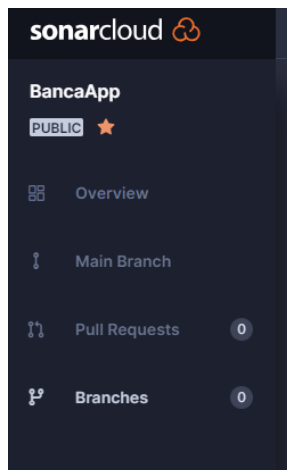
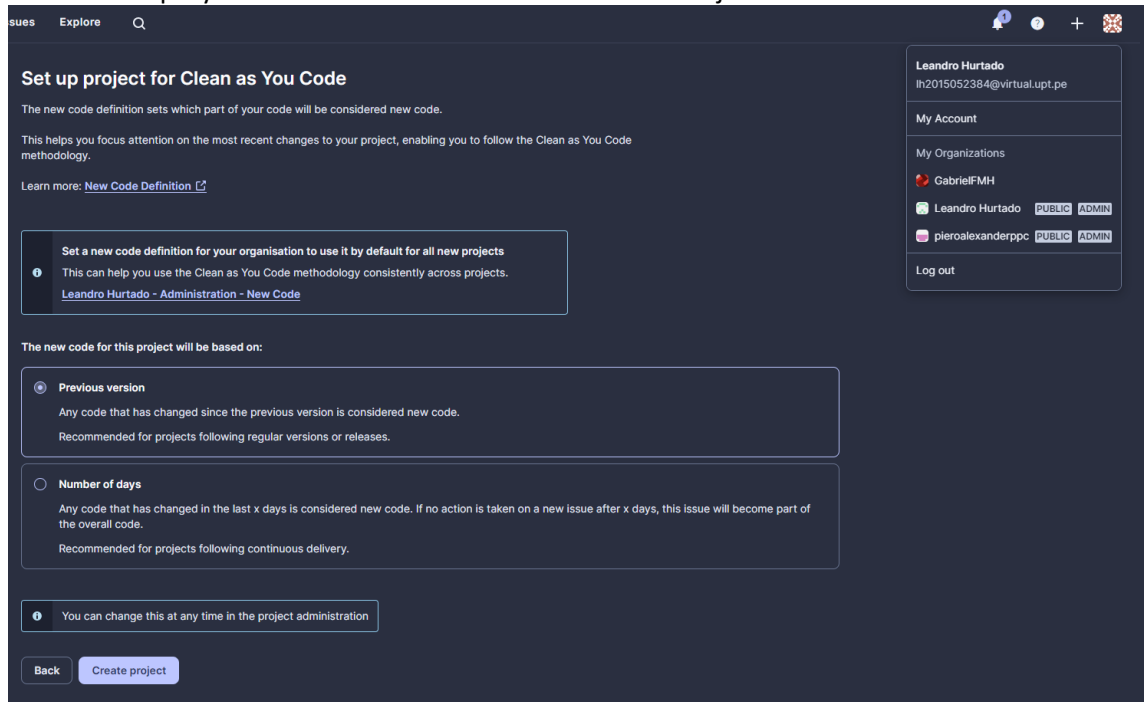
Anyone will be able to browse your source code and see the result of your analysis.

☐ Private PAID PLAN

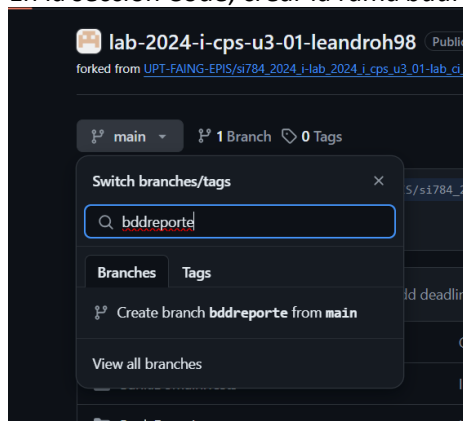
Only members of the organization will be able to browse your source code and see the result of your analysis.

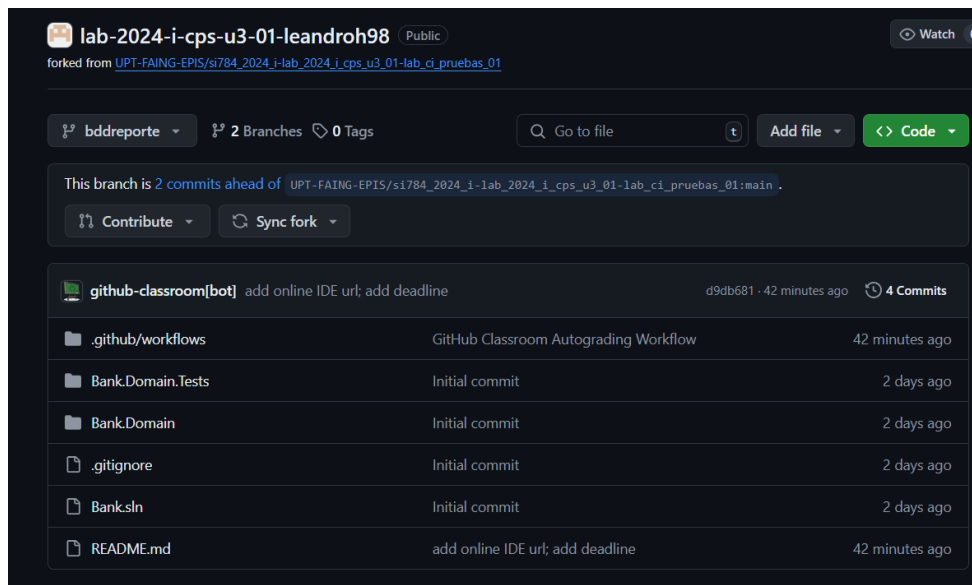
Next

5. En el navegador de internet, en la pagina de SonarCloud, en la pagina de *Set up your project or Clean as You Code*, seleccionar la opción *Previous version*, confirmar la creación del proyecto haciendo click en el boton **Create Project**.

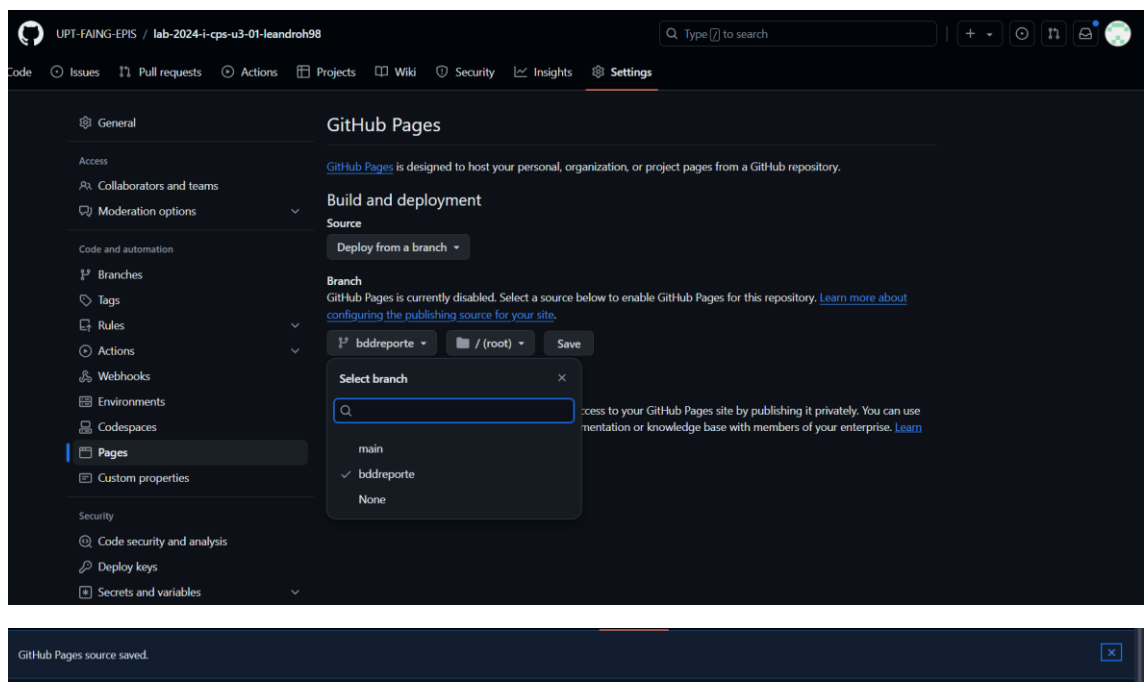


6. En el navegador de internet, ingresar a la pagina Github del repositorio de su proyecto. En la sección **Code**, crear la rama *bddreporte*

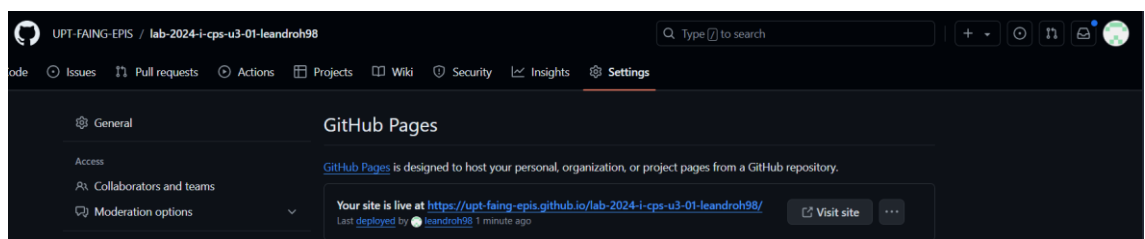




7. En el navegador de internet, en pagina Github del repositorio de su proyecto. En la sección Settings, ingresar a la opción Pages y en Branch seleccionar la rama recientemente creada, seguidamente hacer click en el botón **Save**.

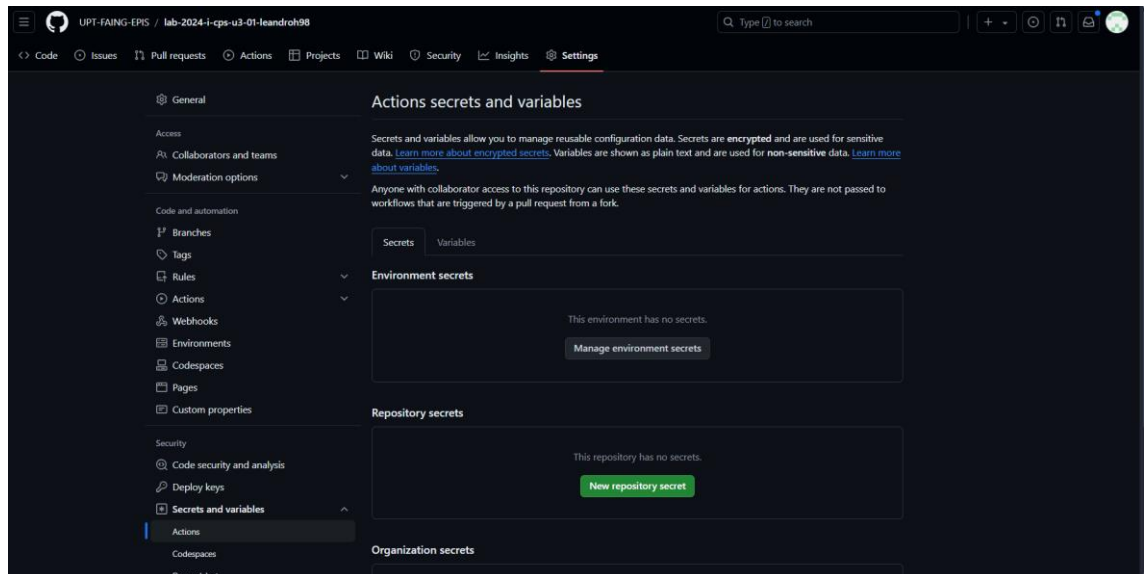


8. En el navegador de internet, en pagina Github del repositorio de su proyecto. En la sección Settings, en la opción Pages despues de unos minutos aparecerá la url publica del proyecto. Tomar nota de esa dirección que sera utilizada mas adelante.





9. En el navegador de internet, en pagina Github del repositorio de su proyecto. En la sección Settings, ingresar a la opción Secrets and variables y luego en la opción Actions, hacer click en el botón *New repository secret*.



10. En el navegador de internet, en pagina Github del repositorio de su proyecto. En la pagina de Actions secrets / New Secret, en el nombre ingresar el valor SONAR\_TOKEN y en secreto ingresar el valor del token de SonarCloud generado en el paso 2.

Actions secrets / New secret

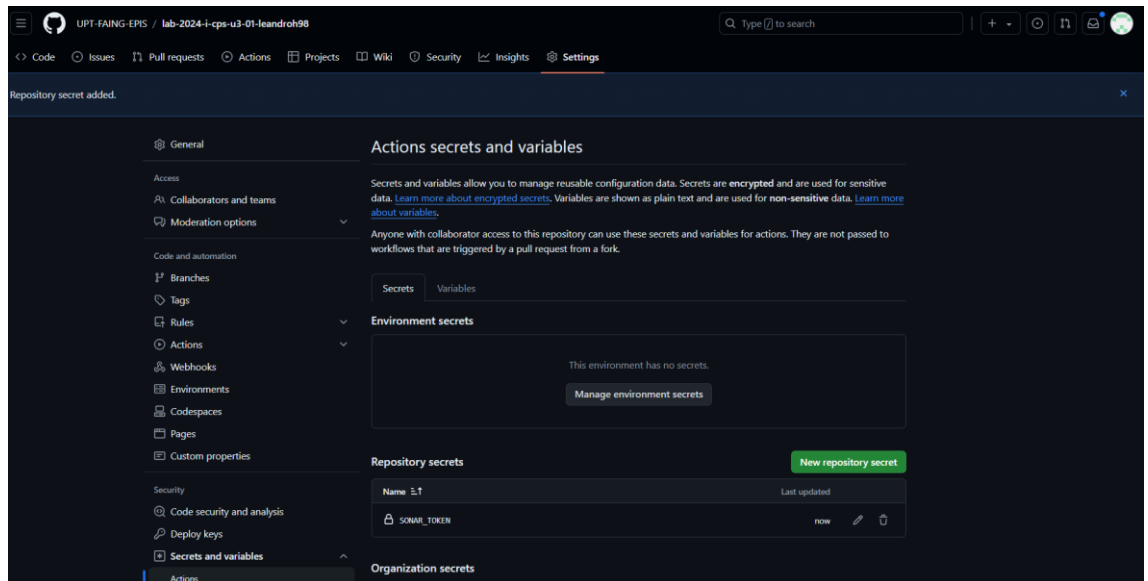
Name \*

SONAR\_TOKEN

Secret \*

aca0fc770390f665c1083a67bd7f3faed9983f7c

Add secret



11. Abrir Visual Studio Code, cargar la carpeta del repositorio del proyecto. Seguidamente crear la carpeta **.github** y dentro de esta la carpeta **workflows**. Seguidamente crear el archivo ci.yml con el siguiente contenido

```
name: Tarea Automatizada de ejecución de pruebas

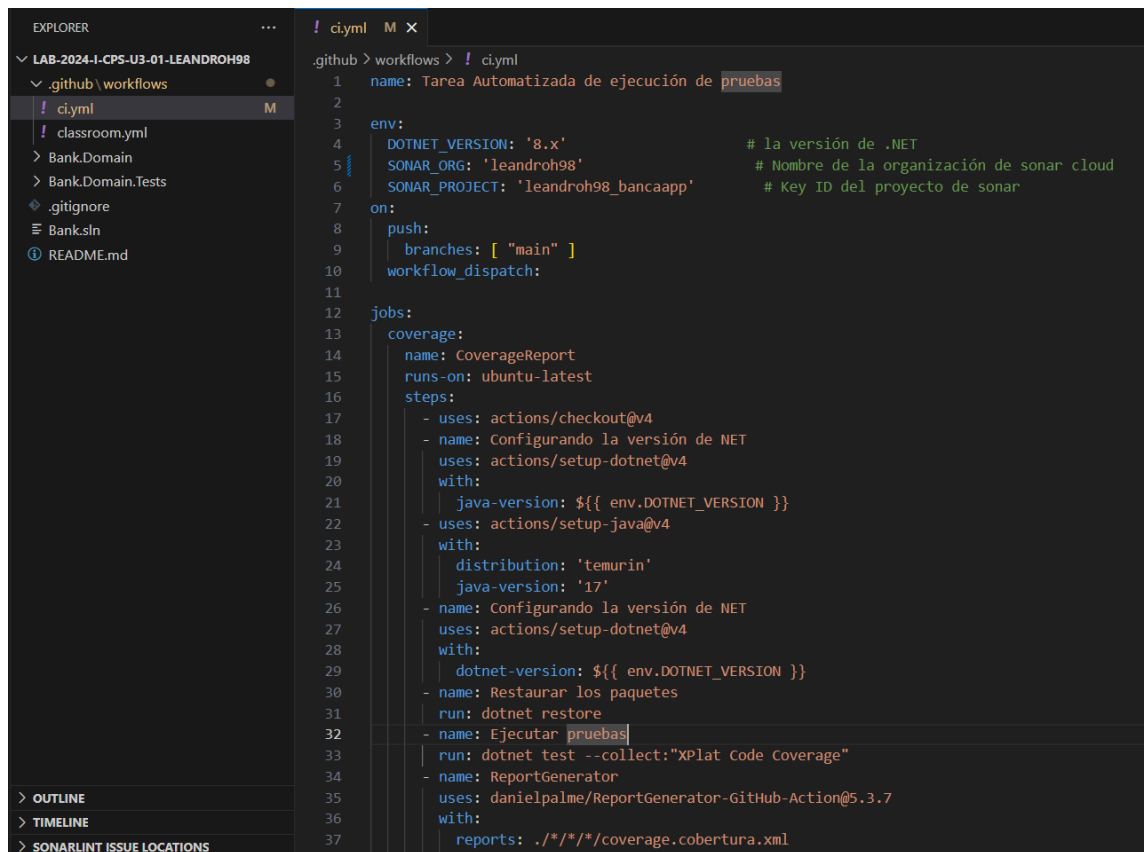
env:
  DOTNET_VERSION: '8.x' # la versión de .NET
  SONAR_ORG: 'p-cuadros' # Nombre de la organización de
sonar cloud
  SONAR_PROJECT: 'p-cuadros_bancaapp' # Key ID del proyecto de sonar
on:
  push:
    branches: [ "main" ]
  workflow_dispatch:

jobs:
  coverage:
    name: CoverageReport
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - name: Configurando la versión de NET
        uses: actions/setup-dotnet@v4
        with:
          java-version: ${ env.DOTNET_VERSION }
      - uses: actions/setup-java@v4
        with:
          distribution: 'temurin'
          java-version: '17'
      - name: Configurando la versión de NET
        uses: actions/setup-dotnet@v4
        with:
          dotnet-version: ${ env.DOTNET_VERSION }
      - name: Restaurar los paquetes
        run: dotnet restore
      - name: Ejecutar pruebas
        run: dotnet test --collect:"XPlat Code Coverage"
      - name: ReportGenerator
        uses: danielpalme/ReportGenerator-GitHub-Action@5.3.7
```

```

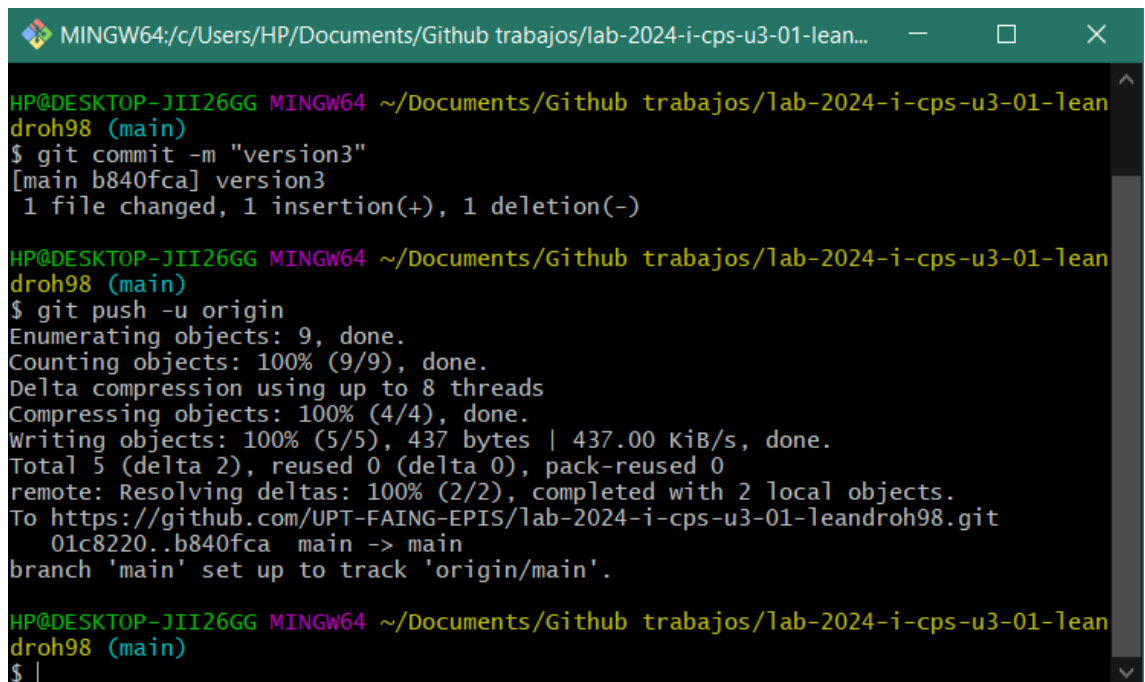
    with:
      reports: ./**/*/coverage.cobertura.xml
      targetdir: coveragereport
      reporttypes:
MarkdownSummary;MarkdownAssembliesSummary;MarkdownSummaryGithub
  - name: Upload coverage report artifact
    uses: actions/upload-artifact@v4
    with:
      name: CoverageReport
      path: coveragereport
  - name: Publish coverage in build summary #
    run: cat coveragereport/SummaryGithub.md >> $GITHUB_STEP_SUMMARY
    shell: bash
  - name: Instalar Scanner
    run: dotnet tool install -g dotnet-sonarscanner
  - name: Ejecutar escaneo
    run: |
      dotnet-sonarscanner begin /k:"${{ env.SONAR_PROJECT }}" /o:"${{
env.SONAR_ORG }}" /d:sonar.login="${{ secrets.SONAR_TOKEN }}"
/d:sonar.host.url="https://sonarcloud.io"
      dotnet build
      dotnet-sonarscanner end /d:sonar.login="${{ secrets.SONAR_TOKEN }}"
  - name: Install Living Doc
    run: dotnet tool install -g SpecFlow.Plus.LivingDoc.CLI
  - name: Generate living doc
    run: livingdoc test-assembly
./Bank.Domain.Tests/bin/Debug/net8.0/Bank.Domain.Tests.dll -t
./Bank.Domain.Tests/bin/Debug/net8.0/TestExecution.json -o
./report/index.html
  - uses: actions/upload-artifact@v3
    with:
      name: specflow
      path: report
  - name: Deploy
    uses: peaceiris/actions-gh-pages@v3
    with:
      github_token: ${ secrets.GITHUB_TOKEN }
      publish_branch: bddreporte
      publish_dir: ./report/

```



```
1 name: Tarea Automatizada de ejecución de pruebas
2
3 env:
4   DOTNET_VERSION: '8.x' # la versión de .NET
5   SONAR_ORG: 'leandroh98' # Nombre de la organización de sonar cloud
6   SONAR_PROJECT: 'leandroh98_bancaapp' # Key ID del proyecto de sonar
7
8 on:
9   push:
10     branches: [ "main" ]
11     workflow_dispatch:
12
13 jobs:
14   coverage:
15     name: CoverageReport
16     runs-on: ubuntu-latest
17     steps:
18       - uses: actions/checkout@v4
19       - name: Configurando la versión de NET
20         uses: actions/setup-dotnet@v4
21         with:
22           java-version: ${ env.DOTNET_VERSION }
23       - uses: actions/setup-java@v4
24         with:
25           distribution: 'temurin'
26           java-version: '17'
27       - name: Configurando la versión de NET
28         uses: actions/setup-dotnet@v4
29         with:
30           dotnet-version: ${ env.DOTNET_VERSION }
31       - name: Restaurar los paquetes
32         run: dotnet restore
33       - name: Ejecutar pruebas
34         run: dotnet test --collect:"XPlat Code Coverage"
35       - name: ReportGenerator
36         uses: danielpalme/ReportGenerator-GitHub-Action@5.3.7
37         with:
38           reports: ./**/coverage.cobertura.xml
```

11. En el Visual Studio Code o en un terminal, subir o confirmar los cambios al repositorio.



```
HP@DESKTOP-JII26GG MINGW64 ~/Documents/Github trabajos/lab-2024-i-cps-u3-01-lean
droh98 (main)
$ git commit -m "version3"
[main b840fca] version3
1 file changed, 1 insertion(+), 1 deletion(-)

HP@DESKTOP-JII26GG MINGW64 ~/Documents/Github trabajos/lab-2024-i-cps-u3-01-lean
droh98 (main)
$ git push -u origin
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 437 bytes | 437.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/UPT-FAING-EPIS/lab-2024-i-cps-u3-01-leandroh98.git
01c8220..b840fca main -> main
branch 'main' set up to track 'origin/main'.
```

12. En el navegador de internet, en pagina Github del repositorio de su proyecto. En la sección Actions, se podra visualizar el siguiente resultado.

UPT-FAING-EPIS / lab-2024-i-cps-u3-01-leandroh98

Type to search

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Actions

All workflows

Autograding Tests

pages-build-deployment

Tarea Automatizada de ejecución de pr...

Management

Caches

Deployments

Attestations

Runners

All workflows

Showing runs from all workflows

8 workflow runs

pages build and deployment

pages-build-deployment #2: by github-pages bot

now

22s

...

version3

Tarea Automatizada de ejecución de pruebas #3: Commit b840fca pushed by leandroh98

main

2 minutes ago

1m 41s

...

version3

Autograding Tests #6: Commit b840fca pushed by leandroh98

main

2 minutes ago

15s

...

Summary

Jobs

CoverageReport

Run details

Usage

Workflow file

Triggered via push 2 minutes ago

leandroh98 pushed -o- b840fca main

Status

Success

Total duration

1m 41s

Artifacts

2

ci.yml

on: push

CoverageReport 1m 32s

CoverageReport summary

Summary

Summary

Generated on:	07/06/2024 - 18:57:16
Coverage date:	07/06/2024 - 18:57:14
Parser:	Cobertura
Assemblies:	1
Classes:	2
Files:	2
Line coverage:	88.5% (31 of 35)
Covered lines:	31
Uncovered lines:	4
Coverable lines:	35
Total lines:	53
Branch coverage:	75% (3 of 4)
Covered branches:	3
Total branches:	4
Method coverage:	Feature is only available for sponsors
Tag:	3_9821434748

Tag:	3_9821434748
------	--------------

## Coverage

▼ Bank.Domain - 88.5%

Name	Line	Branch
Bank.Domain	88.5%	75%
Bank.Domain.Ciente	100%	
Bank.Domain.CuentaAhorro	85.7%	75%

[Job summary generated at run-time](#)

13. En el navegador de internet, en la pagina de SonarCloud, en el nuevo proyecto creado se podra visualizar el resultado de la ejecución

The screenshot shows the SonarCloud web interface. The left sidebar displays the project 'BancaApp' with a 'PUBLIC' status and a star icon. The main content area is titled 'Branch Summary' for the 'main' branch. A prominent red 'Failed' status is shown for the Quality Gate. Below this, a summary of issues is provided: 13 New Issues, 0 Accepted Issues, 0.0% Coverage (Failed), and 0.0% Duplications. The coverage section indicates that the required coverage is ≥ 80.0%, but only 0.0% was achieved on 9 new lines. The duplications section shows that the required duplication is ≤ 3.0%, but 0.0% was achieved on 141 new lines. The Security Hotspots section shows 0 hotspots. A warning banner at the top right states 'The last analysis has warnings. See details'.

14. En el navegador de internet, ingresar a la dirección generada en el paso 7, de Github Pages y se podrá visualizar el siguiente resultado:

upt-faing-epis.github.io/lab-2024-i-cps-u3-01-leandroh98/#/document/Standalone/feature/eda9773807c393a25eb8f1201c4c29d2

Bank.Domain.Tests

generated Jul 6, 2024, 01:57 PM GMT-5

Living Documentation

Analytics

Filter by Keyword

Filter by

Scenario Result

×

+

−

Test results

⬅

Bank.Domain.Tests

1 Passed0 Failed0 Others

Steps

1 Passed0 Failed0 Others

Como cliente quiero realizar depositos y retiros para modificar mi saldo de cuenta

Cliente deposita en su cuenta un monto y es correcto

Cliente retira en su cuenta un monto y es correcto

Cliente retira en su cuenta un monto negativo y es incorrecto

Feature: Como cliente quiero realizar depositos y retiros para modificar mi saldo de cuenta

Scenario: Cliente deposita en su cuenta un monto y es correcto8ms

Given la nueva cuenta numero 12345

When deposito 10

Then el saldo nuevo deberia ser 10

Scenario: Cliente retira en su cuenta un monto y es correcto< 1ms

Given la nueva cuenta numero 12345

And con saldo 10

When retiro 10

Then el saldo nuevo deberia ser 0

Scenario: Cliente retira en su cuenta un monto negativo y es incorrecto< 1ms

Given la nueva cuenta numero 12345

And con saldo 10

When retiro -10

Then deberia ser error

Generated by SpecFlow+LivingDoc - Give us feedback!

upt-faing-epis.github.io/lab-2024-i-cps-u3-01-leandroh98/#/document/Standalone/feature/eda9773807c393a25eb8f1201c4c29d2

Bank.Domain.Tests

generated Jul 6, 2024, 01:57 PM GMT-5

Living Documentation

Analytics

Features

1100.0%0.0%0.0%1PASSEDFAILOthers

Scenarios

3100.0%0.0%0.0%3PASSEDFAILOthers

Steps

11100.0%0.0%0.0%0.0%0.0%0.0%11PASSEDFAILOTHESKIPPEDNOT EXECUTEDPENDINGUNBOUND

Unused Step Definitions

No step definitions were found in the provided assemblies

Are you missing something? Let us know!

Generated by SpecFlow+LivingDoc - Give us feedback!

## Conclusiones

La sesión de laboratorio permitió automatizar la gestión de pruebas de una aplicación mediante Github Actions, integrando herramientas como SonarCloud para análisis de código y generación de reportes de cobertura. Los participantes aprendieron a configurar su entorno de desarrollo, crear y gestionar tokens y proyectos en SonarCloud, así como a estructurar y ejecutar flujos de trabajo en Github Actions para obtener reportes detallados y desplegar documentación generada automáticamente. Este proceso optimiza la detección de errores y mejora la calidad del software de manera continua y eficiente.