

# Configuración del pipeline y su vínculo con el tablero

Antes de comenzar a crear un pipeline, es importante tener una comprensión básica de cómo funciona *GitHub Actions*.

Los conceptos clave que debes conocer son:

- *Workflow*: Es el flujo de trabajo completo que se ejecuta en respuesta a un evento específico (por ejemplo, un *push* o *pull request*).
- *Job*: Es un trabajo específico dentro de un flujo de trabajo, que consiste en una o más tareas.
- Acción: Es un conjunto de tareas predefinidas que se pueden usar en un trabajo.
- *Step*: Son las tareas predefinidas mencionadas anteriormente.

Utilizaremos esta herramienta para automatizar la mayor cantidad de trabajo que podamos, por lo que se irá modificando a medida que avancemos en el proyecto. En nuestro tablero actual tenemos las siguientes columnas:

- Test Cases Implementation
- Application Implementation
- Testing Automation tools
- Refactoring
- Review

Nuestro pipeline lo tenemos configurado para que corra en el paso *Testing Automation Tools*, también posterior a *Refactoring* y en *Review*.

La configuración del mismo se basa en un archivo YAML que representa el flujo de trabajo, el mismo contiene lo siguiente:

```
name: .NET

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

jobs:
  build:
    name: Build and Test and Check Dependencies

    env:
      SOLUTION: 'ArenaGestor.sln'
      PATH_CODE: './codigo/ArenaGestor'
```

```
runs-on: ubuntu-latest

permissions: write-all

steps:
- uses: actions/checkout@v3

- name: Setup .NET
  uses: actions/setup-dotnet@v3
  with:
    dotnet-version: 5.0.x

- name: Restore dependencies
  run: dotnet restore $PATH_CODE/$SOLUTION

- name: Build
  run: dotnet build --no-restore $PATH_CODE/$SOLUTION

- name: Test Backend
  run: dotnet test --no-build --verbosity normal
$PATH_CODE/$SOLUTION --logger "trx;LogFileName=test-results.trx"
--collect:"XPlat Code Coverage" --results-directory ./coverage || true

- name: Test Report
  uses: dorny/test-reporter@v1
  if: always()
  with:
    name: DotNET Tests
    path: '**/test-results.trx'
    reporter: dotnet-trx
    fail-on-error: true

- name: Code Coverage Report
  uses: irongut/CodeCoverageSummary@v1.3.0
  with:
    filename: 'coverage/**/coverage.cobertura.xml'
    badge: true
    fail_below_min: true
    format: markdown
    hide_branch_rate: false
    hide_complexity: false
    indicators: true
    output: both
```

```
    thresholds: '20 80'

- name: Add Coverage PR Comment
  uses: marocchino/sticky-pull-request-comment@v2
  if: github.event_name == 'pull_request'
  with:
    recreate: true
    path: 'code-coverage-results.md'

- name: Depcheck Backend
  uses: dependency-check/Dependency-Check_Action@main
  id: Depcheck-Backend
  with:
    project: 'ArenaGestor'
    path: 'codigo/ArenaGestor/'
    format: 'HTML'
    out: 'reports'
    args: >
      --failOnCVSS 7
      --enableRetired

- name: Upload Test results
  uses: actions/upload-artifact@master
  with:
    name: Depcheck report
    path: ${github.workspace}/reports

- name: Depcheck Frontend
  uses: dependency-check/Dependency-Check_Action@main
  id: Depcheck-Frontend
  with:
    project: 'ArenaGestor-Front'
    path: 'codigo/ArenaGestorFront/'
    format: 'HTML'
    out: 'reports-front'
    args: >
      --enableRetired

- name: Upload Test results Frontend
  uses: actions/upload-artifact@master
  with:
    name: Depcheck report frontend
    path: ${github.workspace}/reports-front
```

Lo que realiza específicamente en los pasos es:

- Corre el pipeline cuando se realiza un *push* o un *pull request* en las ramas *main* y *develop*.
- Setea .NET 5, realiza un restore para instalar las dependencias y luego realiza un *build*.
- Se efectúan los *tests* del *backend*, aquí se corren los *tests* unitarios y se emplean configuraciones adicionales para poder generar reportes de ejecución de pruebas y análisis de cobertura.
- Uno de los resultados es la vista de ejecución de las pruebas, aquí nos queda en el apartado de la izquierda el análisis de todas las pruebas y su resultado.

The screenshot shows the GitHub Actions interface for a workflow named 'DotNET Tests'. The workflow is in a 'succeeded' state, having completed 15 hours ago in 1s. The left sidebar shows the workflow steps: 'Summary', 'Jobs', 'Run details', 'Usage', and 'Workflow file'. The 'Jobs' section is expanded, showing a single job 'Build and Test and Check Depen...' with a status of 'succeeded'. The 'Run details' section is also expanded, showing the 'Usage' tab. The main content area displays the 'DotNET Tests' job results, indicating that 80 tests passed. A table lists the test suites and their results, showing that all tests passed. Below the table, the specific test results for 'ArenaGestor.DataAccessTest.ArtistsManagementTest' are shown, including 'DeleteTest', 'GetById', 'GetTest', 'GetWithFilterByIdTest', 'GetWithFilterThatExistsTest', 'GetWithFilterThatNotExistsTest', 'InsertTest', and 'InsertWithUserTest', all of which passed.

Summary

Jobs

- Build and Test and Check Depen...
- DotNET Tests

Run details

Usage

Workflow file

GitHub Actions / DotNET Tests  
succeeded 15 hours ago in 1s

DotNET Tests ✓

tests 80 passed

✓ coverage/test-results.trx

80 tests were completed in 4s with 80 passed, 0 failed and 0 skipped.

Test suite	Passed	Failed	Skipped	Time
ArenaGestor.DataAccessTest.ArtistsManagementTest	10✓			1s
ArenaGestor.DataAccessTest.BandsManagementTest	8✓			166ms
ArenaGestor.DataAccessTest.ConcertsManagementTest	13✓			167ms
ArenaGestor.DataAccessTest.CountrysManagementTest	3✓			9ms
ArenaGestor.DataAccessTest.GendersManagementTest	8✓			19ms
ArenaGestor.DataAccessTest.MusicalProtagonistManagementTest	4✓			21ms
ArenaGestor.DataAccessTest.RolesManagementTest	2✓			9ms
ArenaGestor.DataAccessTest.SessionManagementTest	4✓			35ms
ArenaGestor.DataAccessTest.SoloistsManagementTest	8✓			78ms
ArenaGestor.DataAccessTest.TicketManagementTest	10✓			89ms
ArenaGestor.DataAccessTest.TicketStatusManagementTest	1✓			6ms
ArenaGestor.DataAccessTest.UsersManagementTest	9✓			30ms

✓ ArenaGestor.DataAccessTest.ArtistsManagementTest

- ✓ DeleteTest
- ✓ GetById
- ✓ GetTest
- ✓ GetWithFilterByIdTest
- ✓ GetWithFilterThatExistsTest
- ✓ GetWithFilterThatNotExistsTest
- ✓ InsertTest
- ✓ InsertWithUserTest

- Luego, en caso de ser un *pull request*, nos mostrará la cobertura de código de las pruebas por línea (*Line Rate*), por caminos que puede tomar la ejecución (*Branch Rate*) y la complejidad del paquete (Relación entre la cantidad de líneas y caminos que puede tomar el código).

github.com/federicoalonso/isa2-ob/pull/48

Mis archivos -... Nube - Armad... Wiki Estudio Seguridad Trello Kaspersky Sec... Google Draw... Casa mi-bse https://chat.o... Config

Feature/gitactions #48  
federicoalonso wants to merge 29 commits into develop from feature/gitactions

- add all permissions ✓ 3f4f4fe
- add complexity to report ✓ 886dcfa
- change name ✓ a45a191

github-actions bot commented yesterday

Code Coverage 38%

Package	Line Rate	Branch Rate	Complexity	Health
ArenaGestor.Business	92%	89%	312	✓
ArenaGestor.BusinessInterface	100%	100%	0	✓
ArenaGestor.Domain	69%	39%	243	—
ArenaGestor.DataAccessInterface	100%	100%	0	✓
ArenaGestor.Extensions	47%	100%	15	—
ArenaGestor.BusinessHelpers	92%	83%	7	✓
ArenaGestor.Business	92%	89%	312	✓
ArenaGestor.BusinessInterface	100%	100%	0	✓
ArenaGestor.Domain	69%	39%	243	—
ArenaGestor.DataAccessInterface	100%	100%	0	✓
ArenaGestor.Extensions	47%	100%	15	—
ArenaGestor.BusinessHelpers	92%	83%	7	✓
ArenaGestor.DataAccess	0%	0%	152	✗
ArenaGestor.API	69%	89%	85	—
ArenaGestor.DataAccessFactory	0%	100%	3	✗
ArenaGestor.Business	0%	0%	312	✗
ArenaGestor.BusinessInterface	100%	100%	0	✓
ArenaGestor.APIContracts	78%	100%	218	—
ArenaGestor.Domain	24%	0%	243	—

- Posteriormente realiza un análisis de dependencias, tanto de *backend* como de *frontend*. Luego del análisis queda un artefacto por cada uno de los escaneos, el de *backend* no presenta mayores problemas, sólo una vulnerabilidad no crítica:

Summary

Jobs

Build and Test and Check Dependencies

DotNET Tests

Run details

Usage

Workflow file

Build and Test and Check Dependencies

The target framework 'net5.0' is out of support and will not receive security updates in the future. Please refer to [this article](#) for more information.

Build and Test and Check Dependencies

The target framework 'net5.0' is out of support and will not receive security updates in the future. Please refer to [this article](#) for more information.

Build and Test and Check Dependencies

The target framework 'net5.0' is out of support and will not receive security updates in the future. Please refer to [this article](#) for more information.

Build and Test and Check Dependencies

The target framework 'net5.0' is out of support and will not receive security updates in the future. Please refer to [this article](#) for more information.

Build and Test and Check Dependencies

The target framework 'net5.0' is out of support and will not receive security updates in the future. Please refer to [this article](#) for more information.

Build and Test and Check Dependencies

The target framework 'net5.0' is out of support and will not receive security updates in the future. Please refer to [this article](#) for more information.

Build and Test and Check Dependencies

The target framework 'net5.0' is out of support and will not receive security updates in the future. Please refer to [this article](#) for more information.

Build and Test and Check Dependencies

The target framework 'net5.0' is out of support and will not receive security updates in the future. Please refer to [this article](#) for more information.

Build and Test and Check Dependencies

The target framework 'net5.0' is out of support and will not receive security updates in the future. Please refer to [this article](#) for more information.

Build and Test and Check Dependencies

The target framework 'net5.0' is out of support and will not receive security updates in the future. Please refer to [this article](#) for more information.

Artifacts

Produced during runtime

Name	Size
<div><div></div>Depcheck report</div>	730 KB
<div><div></div>Depcheck report frontend</div>	257 KB

- El escaneo de dependencias del Frontend sí presenta múltiples vulnerabilidades críticas, por lo que tuvimos que quitarle las restricciones para que pueda pasar el pipeline.



Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AIS-15 condition, and there are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's risk. In no event shall it or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

[How to read the report](#) | [Suppressing false positives](#) | [Getting Help: github issues](#)

♡ Sponsor

Project: ArenaGestor-Front

Scan Information ([show all](#)):

- dependency-check version: 8.2.1
- Report Generated On: Sat, 16 Apr 2023 22:26:45 GMT
- Dependencies Scanned: 14 (14 unique)
- Vulnerable Dependencies: 13
- Vulnerabilities Found: 25
- Vulnerabilities Suppressed: 0
- ...

Summary

Display: [Showing Vulnerable Dependencies \(click to show all\)](#)

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
<a href="#">decode-uri-component@0.2.0</a>		pkg:npm/decode-uri-component@0.2.0	HIGH	2		3
<a href="#">engine.io@6.2.0</a>		pkg:npm/engine.io@6.2.0	MEDIUM	2		3
<a href="#">http-cache-semantics@4.1.0</a>		pkg:npm/http-cache-semantics@4.1.0	HIGH	2		3
<a href="#">json5@1.0.1</a>		pkg:npm/json5@1.0.1	HIGH	2		3
<a href="#">json5@2.2.1</a>		pkg:npm/json5@2.2.1	HIGH	2		3
<a href="#">loader-utils@1.4.0</a>		pkg:npm/loader-utils@1.4.0	CRITICAL	6		3
<a href="#">loader-utils@2.0.2</a>		pkg:npm/loader-utils@2.0.2	CRITICAL	6		3
<a href="#">loader-utils@3.2.0</a>		pkg:npm/loader-utils@3.2.0	HIGH	4		3
<a href="#">minimatch@3.0.4</a>		pkg:npm/minimatch@3.0.4	HIGH	1		3
<a href="#">socket.io-parser@4.0.4</a>		pkg:npm/socket.io-parser@4.0.4	CRITICAL	2		3
<a href="#">terser@5.11.0</a>		pkg:npm/terser@5.11.0	HIGH	2		3
<a href="#">ua-parser-js@0.7.31</a>		pkg:npm/ua-parser-js@0.7.31	HIGH	2		3
<a href="#">webpack@5.67.0</a>		pkg:npm/webpack@5.67.0	HIGH	2		3