

TP 7 - Code Coverage, Análisis estático de Código y Pruebas de Integración

4.1 Agregar Code Coverage a nuestras pruebas unitarias de backend y frontend e integrarlas junto con sus resultados en nuestro pipeline de build.

4.1.1 En el directorio raíz de nuestro proyecto Angular instalar el siguiente paquete:

```
caro@e14:~/Documents/tp_ingsw3/AngularWebApi/EmployeeCrudAngular$ npm install karma-coverage --save-dev

up to date, audited 942 packages in 5s

120 packages are looking for funding
  run `npm fund` for details

11 vulnerabilities (7 moderate, 4 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

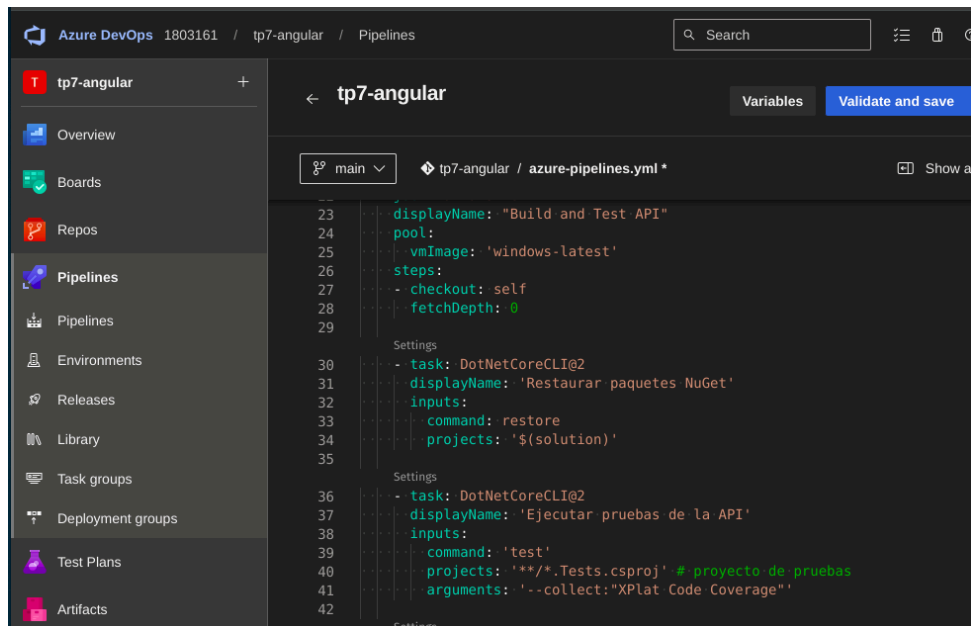
4.1.2 Editar nuestro archivo karma.conf.js para que incluya reporte de cobertura

```
karma.conf.js x
karma.conf.js > ...
1  module.exports = function (config) {
2    config.set({
3      frameworks: ['jasmine', '@angular-devkit/build-angular'],
4      plugins: [
5        require('karma-jasmine'),
6        require('karma-chrome-launcher'),
7        require('karma-junit-reporter'),
8        require('karma-coverage'),
9        require('@angular-devkit/build-angular/plugins/karma')
10     ],
11     reporters: ['progress', 'junit', 'coverage'],
12     junitReporter: {
13       outputDir: 'test-results',
14       outputFile: 'test-results.xml',
15       useBrowserName: false
16     },
17     coverageReporter: {
18       type: 'lcov',
19       dir: require('path').join(__dirname, './coverage'),
20       subdir: '.',
21       file: 'lcov.info'
22     },
23     preprocessors: {
24       // Añade los archivos que deseas instrumentar para la cobertura
25       'src/**/*.ts': ['coverage'], // Asegúrate de instrumentar los archivos de tu aplicación
26     },
27     port: 9876,
28     colors: true,
29     logLevel: config.LOG_INFO,
30     autoWatch: true,
31     browsers: ['ChromeHeadless'],
32     singleRun: true,
33     restartOnFileChange: true
34   });
35 }
```

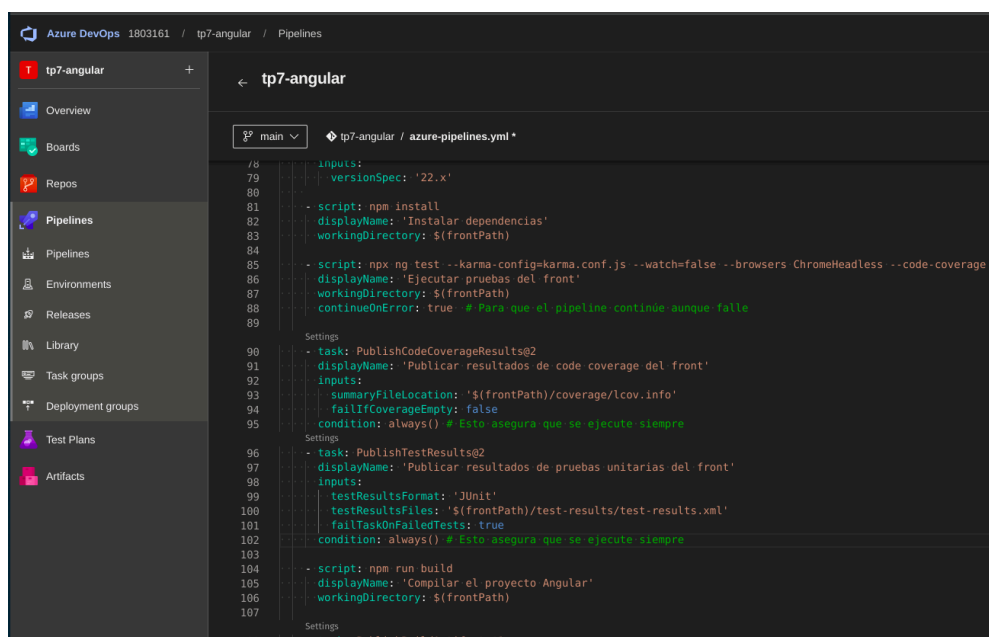
4.1.3 En el dir raíz del proyecto EmployeeCrudApi.Tests ejecutar:

```
caro@e14:~/Documents/tp_ingsw3/AngularWebApi/EmployeeCrudApi.Tests$ dotnet add package coverlet.collector
Determining projects to restore...
Writing /tmp/tmpCoxdBN.tmp
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/lib/dotnet/sdk/8.0.108/trustedroots/codesignctl.pem'.
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/lib/dotnet/sdk/8.0.108/trustedroots/timestampctl.pem'.
info : Adding PackageReference for package 'coverlet.collector' into project '/home/caro/Documents/tp_ingsw3/AngularWebApi/EmployeeCrudApi.Tests/EmployeeC
```

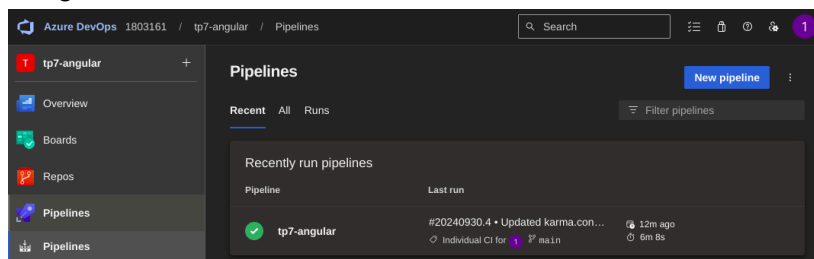
4.1.4 Agregar a nuestro pipeline ANTES del Build de Back la tarea de test con los argumentos especificados y la de publicación de resultados de cobertura:



4.1.5 Agregar a nuestro pipeline ANTES del Build de front la tarea de test y la de publicación de los resultados.



4.1.6 Ejecutar el pipeline y analizar el resultado de las pruebas unitarias y la cobertura de código.



tp7-angular

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

#20240930.4 • Updated karma.conf.js

Run new

This run is being retained as one of 3 recent runs by main (Branch).

View retention leases

SummaryTestsCode Coverage

Triggered by 1803161View 2 changes

Repository and versiontp7-angularmainc304f43

Time started and elapsedToday at 16:126m 8s

Related0 work items3 published

Tests and coverage100% passed45.77% covered

Jobs

Name	Status	Duration
Build and Test API	Success	3m 11s
Build and Test Angular	Success	2m 14s

tp7-angular

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

#20240930.4 • Updated karma.conf.js

Run new

This run is being retained as one of 3 recent runs by main (Branch).

View retention leases

SummaryTestsCode Coverage

Summary

2 Run(s) Completed (2 Passed, 0 Failed)

12Total tests

12Passed0Failed0Others

100%Pass percentage

5s 862msRun duration

0Tests not reported

BugLink

Filter by test or run name

TagsTest runColumn Options

Aborted (+1)

Hooray! There are no test failures.

Change the test runtime filter in view tests relevant to you

tp7-angular

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

#20240930.4 • Updated karma.conf.js

Run new

This run is being retained as one of 3 recent runs by main (Branch).

View retention leases

SummaryTestsCode Coverage

Build configuration: empty | Build platform: empty

Folder view

45.77%Line Coverage

10Modules

142Total Lines

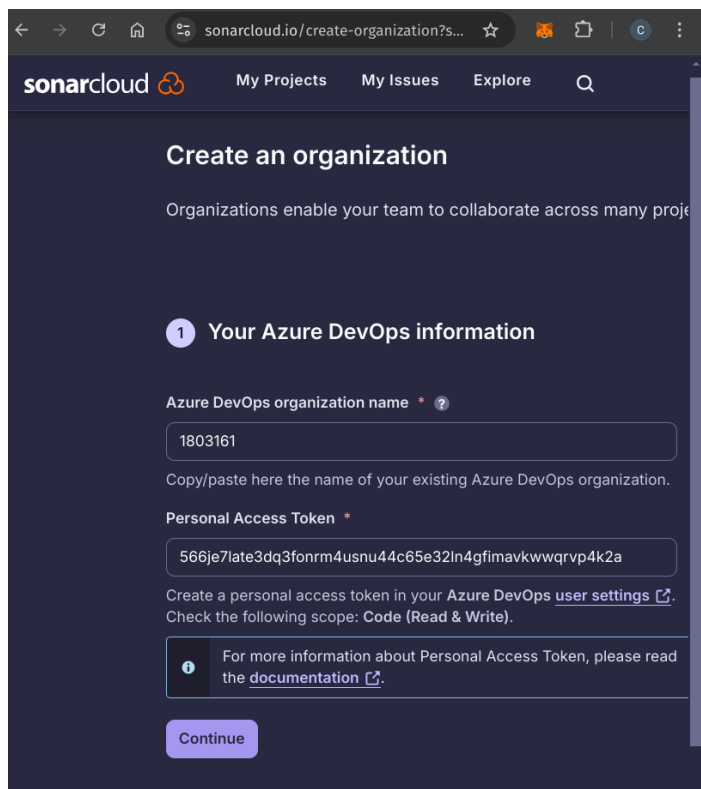
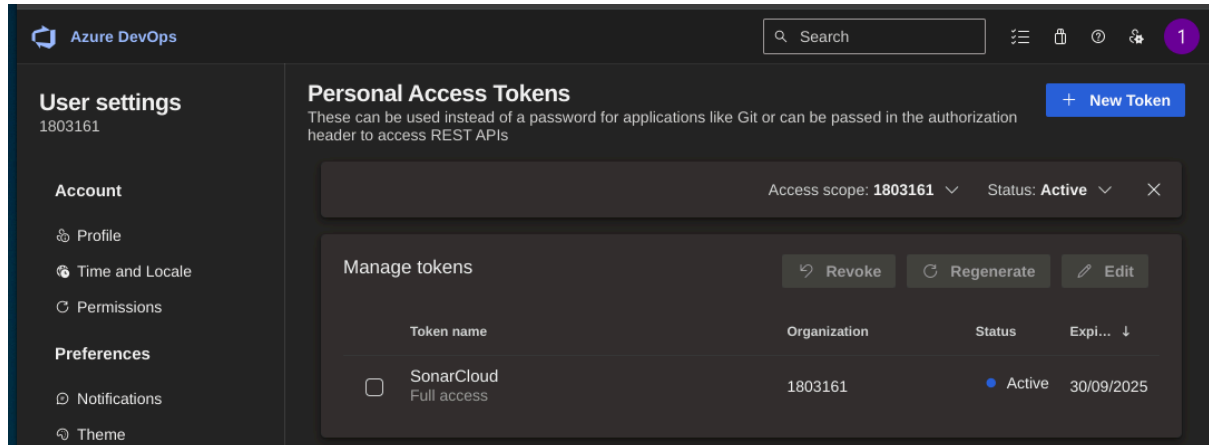
65Covered Lines

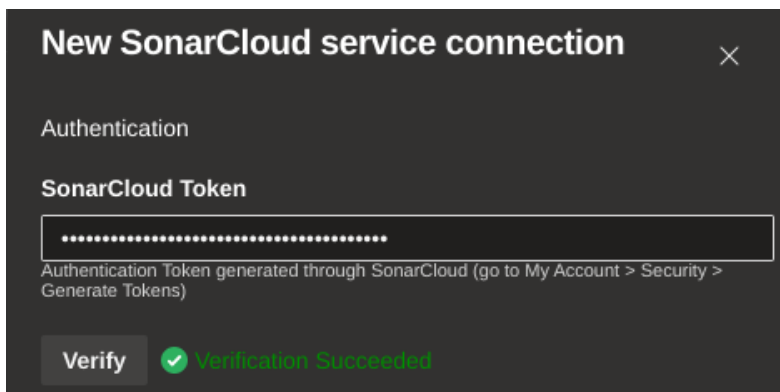
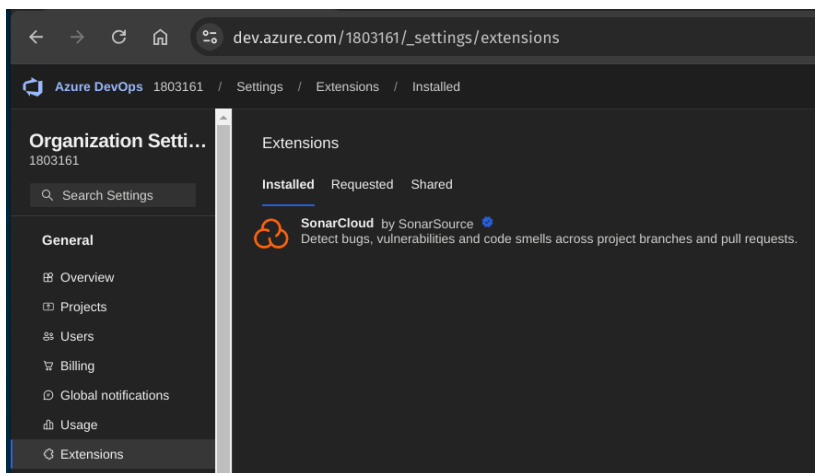
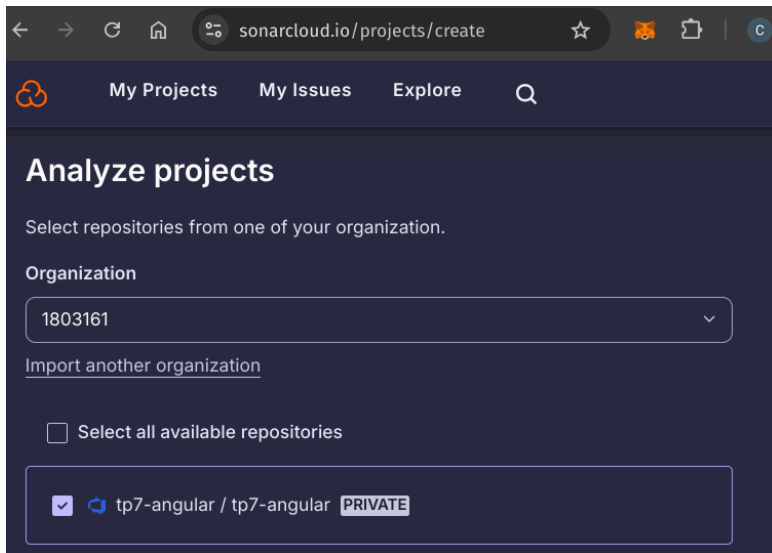
Filter by module name

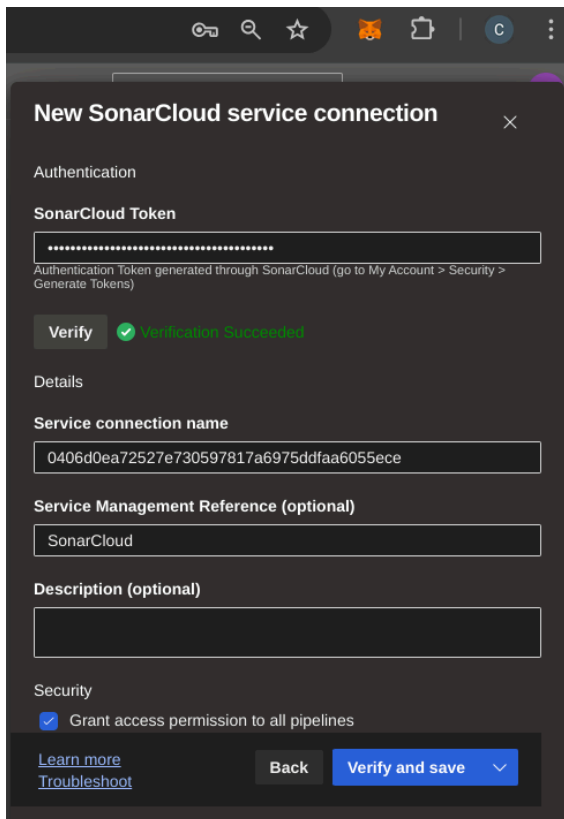
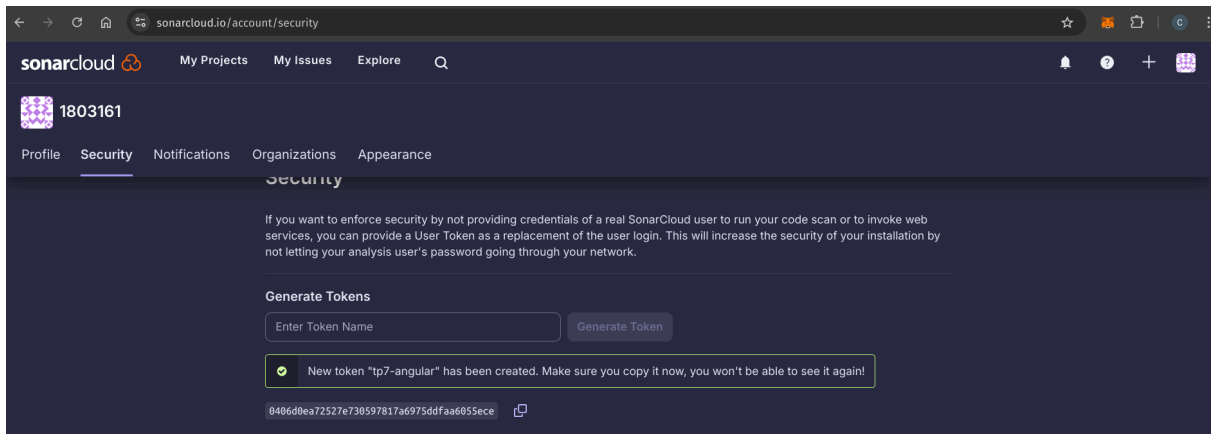
Modules	Covered Lines	Total Lines	Line Coverage	Coverage Chart
D:\a\1s\EmployeeCrudApi\EmployeeCrudApi\Controllers\EmployeeController.cs	36	38	94.74%	
D:\a\1s\EmployeeCrudApi\EmployeeCrudApi\Data\ApplicationDbContext.cs	4	4	100.00%	
D:\a\1s\EmployeeCrudApi\EmployeeCrudApi\Models\Employee.cs	3	3	100.00%	
D:\a\1s\EmployeeCrudApi\EmployeeCrudApi\Program.cs	0	24	0.00%	
src\app\addemployee\addemployee.component.ts	8	32	25.00%	
src\app\app.component.ts	2	2	100.00%	
src\app\employee.model.ts	3	3	100.00%	
src\app\employee.service.ts	3	13	23.08%	
src\app\employee\employee.component.ts	5	22	22.73%	
src\environments\environment.ts	1	1	100.00%	

4.2 Agregar Análisis Estático de Código con SonarCloud: Demostración de cómo integrar SonarCloud en un pipeline de CI/CD y cómo leer los reportes de análisis estático.

4.2.1 Integraremos SonarCloud para analizar el código fuente. Configurar SonarCloud en nuestro pipeline siguiendo instructivo 5.1







Colocaremos una tarea de SonarCloud ANTES de nuestra tarea de Build

```
← tp7-angular

main ▾ tp7-angular / azure-pipelines.yml *

30 - task: DotNetCoreCLI@2
31   displayName: 'Restaurar paquetes NuGet'
32   inputs:
33     command: restore
34     projects: '$(solution)'
35
36 Settings
37 - task: DotNetCoreCLI@2
38   displayName: 'Ejecutar pruebas de la API'
39   inputs:
40     command: 'test'
41     projects: '**/*.Tests.csproj' # proyecto de pruebas
42     arguments: '--collect:"XPlat Code Coverage"'
43
44 Settings
45 - task: SonarCloudPrepare@2
46   inputs:
47     SonarCloud: '0406d0ea72527e730597817a6975ddfaa6055ece' #Nombre de nuestra Service Conn
48     organization: '1803161' #Nombre de nuestra organizacion SonarCloud
49     scannerMode: 'MSBuild'
50     projectKey: '1803161_tp7-angular' #Key de nuestro proyecto en SonarCloud
51     projectName: 'tp7-angular' #Nombre de nuestro proyecto en SonarCloud
52     displayName: 'Prepare SonarCloud'
53
54 Settings
55 - task: PublishCodeCoverageResults@2
56   displayName: 'Publicar resultados de code coverage del back-end'
57   inputs:
58     summaryFileLocation: '$(Agent.TempDirectory)**/*.cobertura.xml'
59     failIfCoverageEmpty: false
```

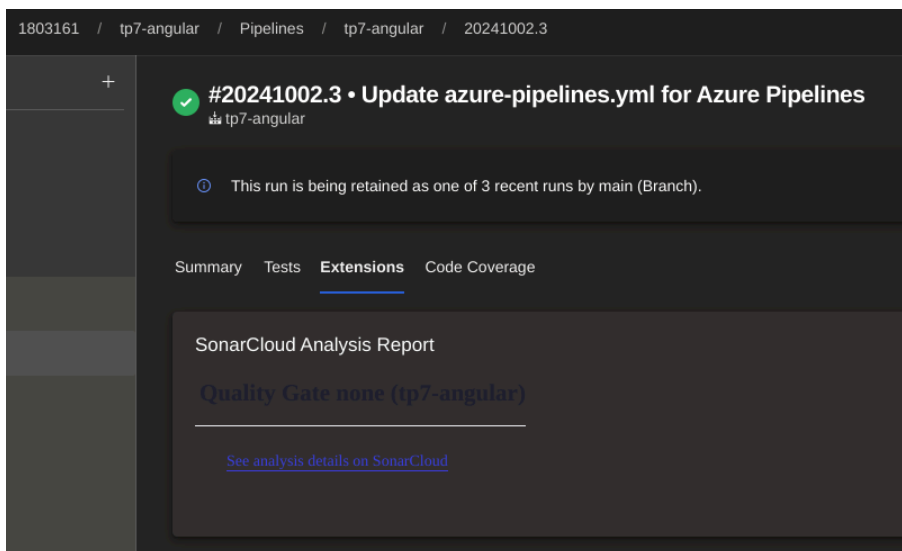
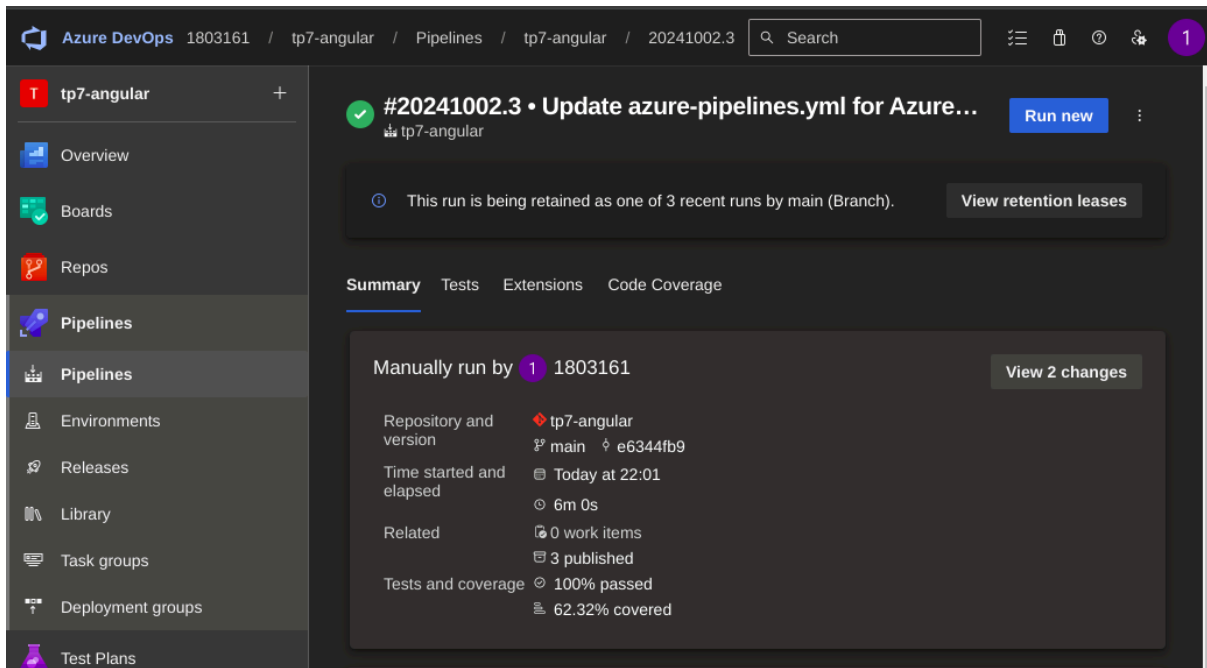
Agregamos otra tarea de SonarCloud DESPUES de nuestra tarea de Build. Click the "Search tasks" field.

```
← tp7-angular

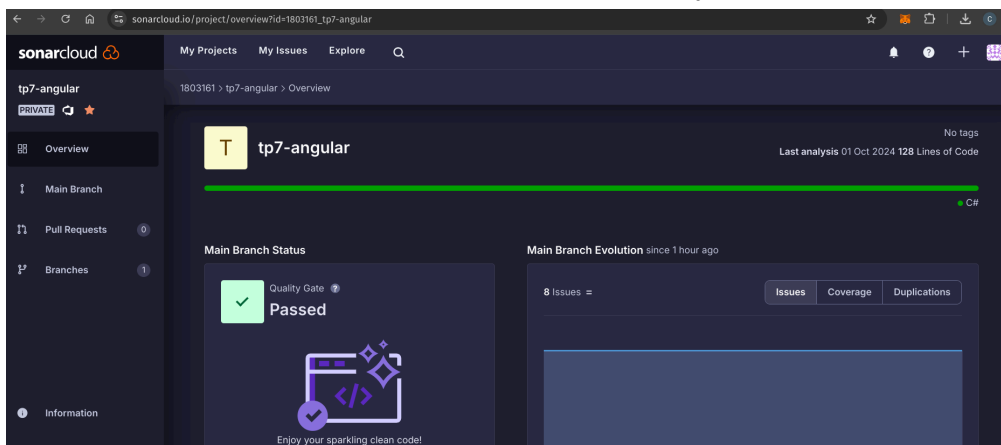
main ▾ tp7-angular / azure-pipelines.yml *

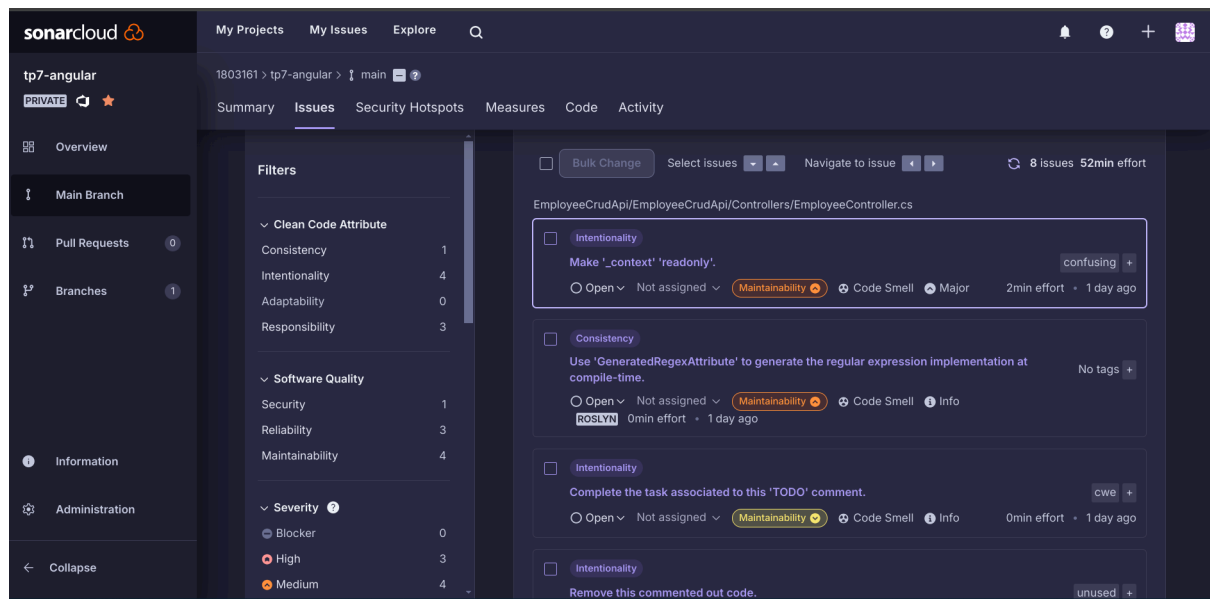
52 - task: PublishCodeCoverageResults@2
53   displayName: 'Publicar resultados de code coverage del back-end'
54   inputs:
55     summaryFileLocation: '$(Agent.TempDirectory)**/*.cobertura.xml'
56     failIfCoverageEmpty: false
57
58 Settings
59 - task: DotNetCoreCLI@2
60   displayName: 'Compilar la API'
61   inputs:
62     command: build
63     projects: '$(solution)'
64     arguments: '--configuration $(buildConfiguration)'
65
66 Settings
67 - task: DotNetCoreCLI@2
68   displayName: 'Publicar aplicación'
69   inputs:
70     command: publish
71     publishWebProjects: True
72     arguments: '--configuration $(buildConfiguration) --output $(Build.ArtifactStagingDirectory)'
73     zipAfterPublish: true
74
75 Settings
76 - task: SonarCloudAnalyze@2
77   inputs:
78     jdkversion: 'JAVA_HOME_17_X64'
79
80 Settings
81 - task: SonarCloudPublish@2
82   inputs:
83     pollingTimeoutSec: '300'
```

4.2.2 Vemos el resultado de nuestro pipeline, en extensions tenemos un link al análisis realizado por SonarCloud



4.2.3 Ir al link y analizar toda la información obtenida. Detallar en la entrega del TP los puntos más relevantes del informe, qué significan y para qué sirven. Corremos el pipeline 2 veces para poder ver el QualityGate.





SonarCloud ayuda a identificar problemas de calidad en tu código, favoreciendo la creación de un código más robusto y seguro. Los puntos más relevantes son:

Clean Code Attribute:

Intentionality: Advertencias sobre aspectos del código para mejorar su claridad.

Consistency: Aspectos donde se puede mejorar la consistencia en la implementación.

Responsibility: Problemas que afectan la responsabilidad del código y su diseño.

Software Quality: (Security, Reliability, Maintainability)

Advertencias de seguridad que requieren atención, la fiabilidad del código que podrían afectar su comportamiento y la mantenibilidad del código.

Severity: Clasifica los problemas según su gravedad y urgencia.

High: Problemas que requieren atención inmediata.

Medium: Problemas de gravedad media que también son importantes.

Low: Problemas menores que no son críticos pero que podrían mejorarse.

Security Hotspots:

High Priority: Authentication

Hay un posible uso de una palabra clave "password" que sugiere un posible hard-coded credential. Esto es una alerta crítica y debe revisarse para evitar problemas de seguridad.

Medium Priority: Denial of Service (DoS):

Advertencia relacionada con DoS.

Low Priority: Insecure Configuration

Se detectó una configuración insegura que debería corregirse para mejorar la seguridad general del sistema.

4.3 Pruebas de Integración con Cypress:

4.3.1 En el directorio raíz de nuestro proyecto Angular instalar el siguiente paquete:

```
caro@e14:~/Documents/tp_ingsw3/AngularWebApi/EmployeeCrudAngular$ npm install cypress --save-dev

added 113 packages, and audited 1055 packages in 38s

132 packages are looking for funding
  run `npm fund` for details

11 vulnerabilities (7 moderate, 4 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

4.3.2 Abrir Cypress:

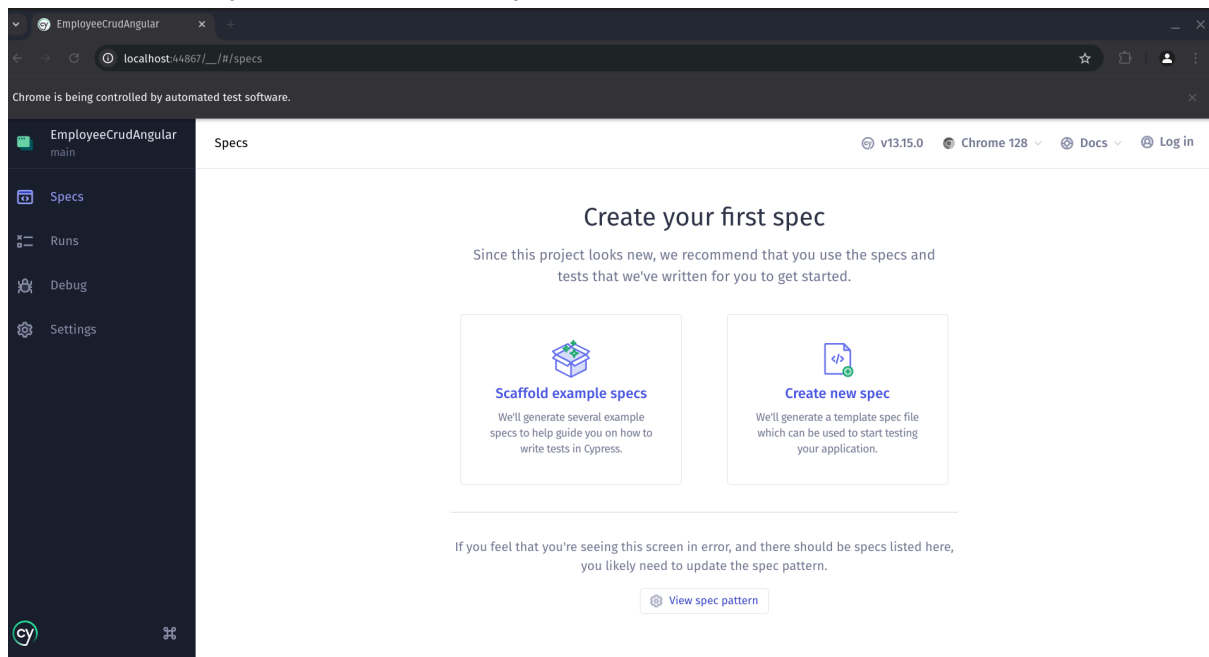
```
caro@e14:~/Documents/tp_ingsw3/AngularWebApi/EmployeeCrudAngular$ npx cypress open
It looks like this is your first time using Cypress: 13.15.0

✓ Verified Cypress! /home/caro/.cache/Cypress/13.15.0/Cypress

Opening Cypress...

DevTools listening on ws://127.0.0.1:44933/devtools/browser/9c738a0a-0f0a-4ae0-950d-60353a5e1265
Missing baseUrl in compilerOptions. tsconfig-paths will be skipped
```

4.3.3 Inicializar Cypress en nuestro proyecto como se indica en el instructivo 5.2



4.3.4 Crear nuestra primera prueba navegando a nuestro front.

The image shows the Cypress test runner interface on the left and the application being tested on the right. The application is 'EmployeeCrudAngular' and displays a list of employees.

Cypress Test Runner (Left):

- Specs: 1 passed, 0 failed, 0 pending, 0 skipped, 0 retries.
- Test Body:
 - visit http://localhost:4200/ (xhr) GET 200 http://localhost:7150/api/Employee/getall
 - get h1
 - assert expected <h1> to contain EmployeeCrudAngular

Application Interface (Right):

EmployeeCrudAngular

Employees: Type to search... Add New Employee

Id	Name	Created Date		
1	Juan Perez	05/09/2024 00:00:00		
2	Carla Ruiz	05/09/2024 22:22:47		
3	Carlos Gomez	06/09/2024 09:16:50		
4	Joaquin Zarate	06/09/2024 09:19:37		
5	Luis Rodriguez	06/09/2024 09:23:31		

4.3.6 Modificar nuestra prueba para que falle. Ejecutamos Cypress en modo headless (sin GUI)

The image shows a terminal window running Cypress in headless mode. The test results are as follows:

```
Running: primer_test.cy.js
(1 of 21)
[152462:1001/231038.481502:ERROR:browser_main_loop.cc(275)] Glib: Failed to set scheduler settings: Operation not permitted

Mi primera prueba
  ✓ Carga correctamente la página de ejemplo (366ms)

1 passing (1s)

(Run Results)
  Tests: 1
  Passing: 1
  Failing: 0
  Pending: 0
  Skipped: 0
  Screenshots: 0
  Video: false
  Duration: 1 second
  Spec Ran: primer_test.cy.js
```

The image shows the VS Code editor with the file 'primer_test.cy.js' open. The code is as follows:

```
describe('Mi primera prueba', () => {
  it('Carga correctamente la página de ejemplo', () => {
    cy.visit('http://localhost:4200/') // Colocar la url local o de Azure
    cy.get('h1').should('contain', 'EmployeeCrudAngularMAL') // Verificar
  })
})
```

```
Running: primer_test.cy.js
(1 of 21)
[155839:1001/231448.226640:ERROR:browser_main_loop.cc(275)] GLib: Failed to set scheduler settings: Operation not permitted

Mi primera prueba
  1) Carga correctamente la página de ejemplo

0 passing (5s)
1 failing

1) Mi primera prueba
  Carga correctamente la página de ejemplo:
    AssertionError: Timed out retrying after 4000ms: expected '<h1>' to contain 'EmployeeCrudAngularMAL'
    at Context.eval (webpack://employee-crud-angular/./cypress/e2e/primer_test.cy.js:4:19)

(Results)

| Tests:      | 1 |
| Passing:    | 0 |
| Failing:    | 1 |
| Pending:    | 0 |
```

EXPLORER

CYPRESS

downloads

e2e

1-getting-started

2-advanced-examples

JS primer_test.cy.js

fixtures

screenshots

2-advanced-examples

primer_test.cy.js

Mi primera prueba -- Carga correctament...

support

JS primer_test.cy.js

Mi primera prueba -- Carga correctamente la página de ejemplo (failed).png

screenshots > primer_test.cy.js > Mi primera prueba -- Carga correctamente la página de ejemplo (failed).png

Specs

primo_test cy.js 00:05

Mi primera prueba

Carga correctamente la página de ejemplo

TEST BODY

1 visit http://localhost:4200/ (xhr) GET 200 http://localhost:7150/api/Employee/getall

2 get h1

3 -assert expected <h1> to contain EmployeeCrudAngularMAL

http://localhost:4200/

EmployeeCrudAngular

Employees: Type to search... Add New Employee

Id	Name	Created Date		
1	Juan Perez	05/09/2024 00:00:00		
2	Carla Ruiz	05/09/2024 22:22:47		
3	Carlos Gomez	06/09/2024 09:16:50		
4	Joaquin Zarate	06/09/2024 09:19:37		
5	Luis Rodriguez	06/09/2024 09:23:31		

4.3.6 Grabar nuestras pruebas para que Cypress genere código automático y genere reportes:

EXPLORER

EMPLOYEECRUDANGULAR

.angular

.vscode

cypress

node_modules

src

test-results

.editorconfig

.gitignore

angular.json

TS cypress.config.ts

karma.conf.js

package-lock.json

package.json

TS cypress.config.ts > ...

1 import { defineConfig } from "cypress";

2

3 export default defineConfig({

4 e2e: {

5 setupNodeEvents(on, config) {

6 // implement node event listeners here

7 },

8 reporter: 'junit', // Configura el reporter a JUnit

9 reporterOptions: {

10 mochaFile: 'cypress/results/results-[hash].xml', // Directorio y nombre de los archivos

11 toConsole: true, // Opcional: imprime los resultados en la consola

12 },

13 },

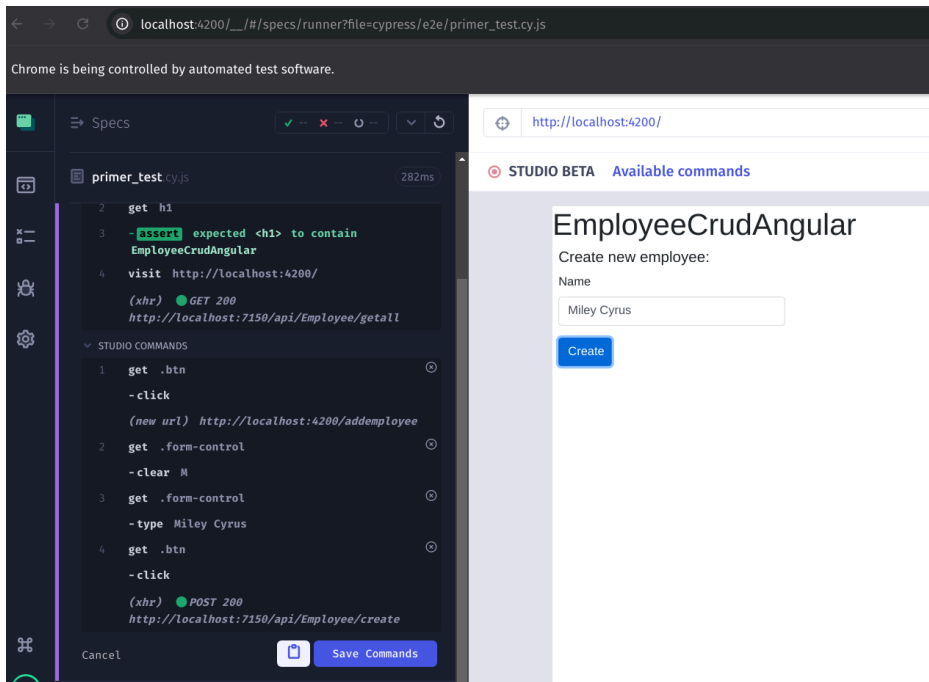
14 experimentalStudio: true,

15 });

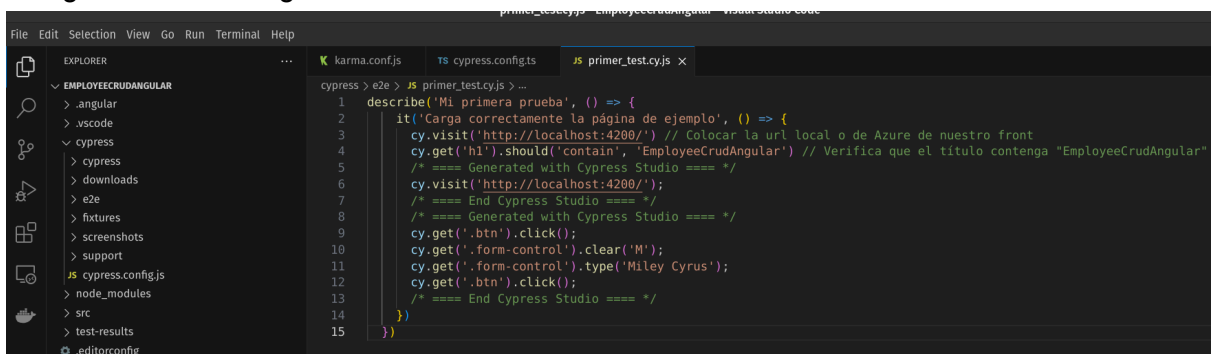
16

caro@e14:~/Documents/tp_ingsw3/AngularWebApi/EmployeeCrudAngular/cypress\$

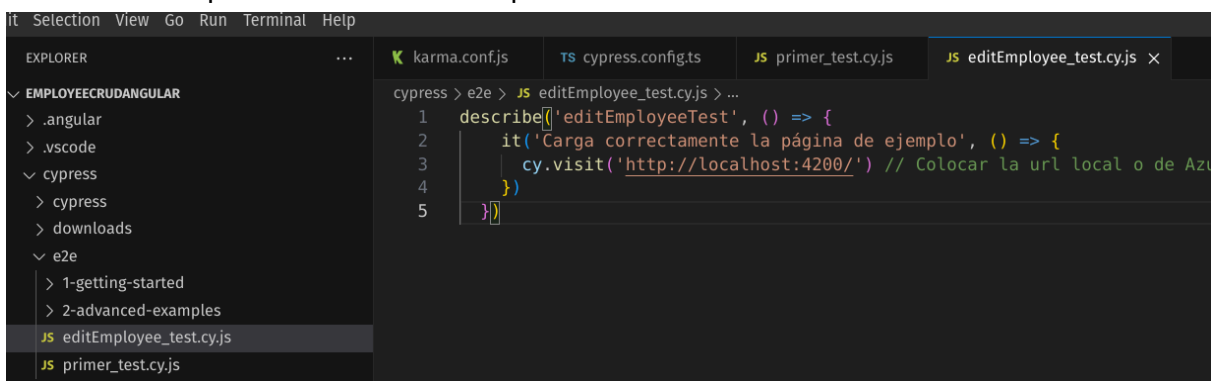
npx cypress open



Nos genera este código de manera automática



4.3.7 Hacemos prueba de editar un empleado



localhost:4200/_/#/specs/runner?file=cypress/e2e/editEmployee_test.cy.js

Chrome is being controlled by automated test software.

Specs

editEmployee_test.cy.js 192ms

STUDIO COMMANDS

```
1 get :nth-child(7) > :nth-child(4) > a > .fa
  -click
  (new url) http://localhost:4200/addEmployee?id=6
  (xhr) GET 200 http://localhost:7150/api/Employee/getall
2 get .form-control
  -clear
3 get .form-control
  -type Lando Norris
4 get .btn
  -click
  (xhr) PUT 200 http://localhost:7150/api/Employee/update
```

Cancel Save Commands

STUDIO BETA Available commands

EmployeeCrudAngular

Create new employee:

Name

Lando Norris

Edit

karma.conf.js TS cypress.config.ts JS primer_test.cy.js JS editEmployee_test.cy.js

```
cypress > e2e > JS editEmployee_test.cy.js > ...
1 describe('editEmployeeTest', () => {
2   it('Carga correctamente la página de ejemplo', () => {
3     cy.visit('http://localhost:4200/') // Colocar la url local o de Azu
4     /* ==== Generated with Cypress Studio ==== */
5     cy.get(':nth-child(7) > :nth-child(4) > a > .fa').click();
6     cy.get('.form-control').clear();
7     cy.get('.form-control').type('Lando Norris');
8     cy.get('.btn').click();
9     /* ==== End Cypress Studio ==== */
10  })
11 })
```

Proyecto de AzureDevOps: <https://dev.azure.com/1803161/tp7-angular>