

7.1 Agregar Code Coverage a nuestras pruebas unitarias de backend y front-end e integrarlas junto con sus resultados en nuestro pipeline de build.

Instalar karma-coverage

```
C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\Angular_WebAPI\NetCore8_CRUD_Sample> npm install karma-coverage --save-dev

added 66 packages, and audited 74 packages in 9s

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

found 0 vulnerabilities
```

Edito el archivo karma.conf.js

```
EmployeeCrudAngular > karma.conf.js > <unknown> > <exports>
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 };
36
```

En el dir raíz del proyecto EmployeeCrudApi.Tests ejecutar:

```
C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\Angular_WebAPI\NetCore8_CRUD_Sample\EmployeeCrudApi.Tests>dotnet add package coverlet.collector
Determinando los proyectos que se van a restaurar...
Writing C:\Users\Manu\AppData\Local\Temp\tmp5525cb.tmp
info : La validación de la cadena de certificados X.509 utilizará el almacén de confianza predeterminado seleccionado por .NET para la firma de código.
info : La validación de la cadena de certificados X.509 utilizará el almacén de confianza predeterminado seleccionado por .NET para la marca de tiempo.
info : Agregando PackageReference para el paquete "coverlet.collector" al proyecto "C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\Angular_WebAPI\NetCore8_CRUD_Sample\EmployeeCrudApi.Tests\EmployeeCrudApi.Tests.csproj".
info : GET https://api.nuget.org/v3/registration5-gz-semver2/coverlet.collector/index.json
info : OK https://api.nuget.org/v3/registration5-gz-semver2/coverlet.collector/index.json 316 ms
info : Restaurando paquetes para C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\Angular_WebAPI\NetCore8_CRUD_Sample\EmployeeCrudApi.Tests\EmployeeCrudApi.Tests.csproj...
info : GET https://api.nuget.org/v3/vulnerabilities/index.json
info : OK https://api.nuget.org/v3/vulnerabilities/index.json 809 ms
info : GET https://api.nuget.org/v3/vulnerabilities/2024.10.02.05.41.40/vulnerability.base.json
info : OK https://api.nuget.org/v3/vulnerabilities/2024.10.02.05.41.40/vulnerability.base.json 200 ms
info : OK https://api.nuget.org/v3/vulnerabilities/2024.10.02.05.41.40/vulnerability.update.json
info : OK https://api.nuget.org/v3/vulnerabilities/2024.10.02.05.41.40/vulnerability.update.json 366 ms
info : El paquete "coverlet.collector" es compatible con todos los marcos de trabajo especificados del proyecto "C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\Angular_WebAPI\NetCore8_CRUD_Sample\EmployeeCrudApi.Tests\EmployeeCrudApi.Tests.csproj".
info : Se actualizó PackageReference para la versión "6.0.2" del paquete "coverlet.collector" en el archivo "C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\Angular_WebAPI\NetCore8_CRUD_Sample\EmployeeCrudApi.Tests\EmployeeCrudApi.Tests.csproj".
info : El archivo de recursos no ha cambiado, así que se omitirá su escritura. Ruta de acceso: C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\Angular_WebAPI\NetCore8_CRUD_Sample\EmployeeCrudApi.Tests\obj\project.assets.json
log : Se ha restaurado C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\Angular_WebAPI\NetCore8_CRUD_Sample\EmployeeCrudApi.Tests\EmployeeCrudApi.Tests.csproj (en 1,68 s).
```

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

```
# ASP.NET Core (.NET Framework)

# Build and test ASP.NET Core projects targeting the full .NET Framework.
# Add steps that publish symbols, save build artifacts, and more:
# https://docs.microsoft.com/azure/devops/pipelines/languages/dotnet-core

trigger:
- main

pool:
  vmImage: 'windows-latest'

variables:
```

```

solution: '**/*.sln'
buildPlatform: 'Any CPU'
buildConfiguration: 'Release'
frontPath: './EmployeeCrudAngular'

stages:
- stage: BuildAndTest
  displayName: "Build and Test API and Front"
  jobs:
  - job: BuildDotnet
    displayName: "Build and Test API"
    pool:
      vmImage: 'windows-latest'
    steps:
    - checkout: self
      fetchDepth: 0

    - task: DotNetCoreCLI@2
      displayName: 'Restaurar paquetes NuGet'
      inputs:
        command: restore
        projects: '$(solution)'

    - task: DotNetCoreCLI@2
      displayName: 'Ejecutar pruebas de la API'
      inputs:
        command: 'test'
        projects: '**/*.Tests.csproj'
        arguments: '--collect:"XPlat Code Coverage"'

    - task: PublishCodeCoverageResults@2
      displayName: 'Publicar resultados de code coverage del back-end'
      inputs:
        summaryFileLocation: '$(Agent.TempDirectory)**/*.cobertura.xml'
        failIfCoverageEmpty: false

    - task: DotNetCoreCLI@2
      displayName: 'Compilar la API'
      inputs:
        command: build
        projects: '$(solution)'
        arguments: '--configuration $(buildConfiguration)'

    - task: DotNetCoreCLI@2
      displayName: 'Publicar aplicación'
      inputs:
        command: publish
        publishWebProjects: true
        arguments: '--configuration $(buildConfiguration) --output $(Build.ArtifactStagingDirectory)'
        zipAfterPublish: true

    - task: PublishBuildArtifacts@1
      displayName: 'Publicar artefactos de compilación'
      inputs:
        PathToPublish: '$(Build.ArtifactStagingDirectory)'
        ArtifactName: 'api-drop'
        publishLocation: 'Container'

  - job: BuildAngular
    displayName: "Build and Test Angular"
    pool:
      vmImage: 'ubuntu-latest'
    steps:
    - task: NodeTool@0
      displayName: 'Instalar Node.js'
      inputs:
        versionSpec: '22.x'

    - script: npm install
      displayName: 'Instalar dependencias'

```

```

workingDirectory: $(frontPath)

- script: npx ng test --karma-config=karma.conf.js --watch=false --browsers ChromeHeadless --code-coverage
  displayName: 'Ejecutar pruebas del front'
  workingDirectory: $(frontPath)
  continueOnError: true # Continue on test failure

- task: PublishCodeCoverageResults@2
  displayName: 'Publicar resultados de code coverage del front'
  inputs:
    summaryFileLocation: '$(frontPath)/coverage/lcov.info'
    failIfCoverageEmpty: false
  condition: always()

- task: PublishTestResults@2
  displayName: 'Publicar resultados de pruebas unitarias del front'
  inputs:
    testResultsFormat: 'JUnit'
    testResultsFiles: '$(frontPath)/test-results/test-results.xml'
    failTaskOnFailedTests: true
  condition: always()

- script: npm run build
  displayName: 'Compilar el proyecto Angular'
  workingDirectory: $(frontPath)

- task: PublishBuildArtifacts@1
  displayName: 'Publicar artefactos Angular'
  inputs:
    PathToPublish: '$(frontPath)/dist'
    ArtifactName: 'front-drop'

```

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

#20241012.1 • EmployeeAngular

TP07-Angular

Run new

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

View retention leases

Summary Tests Code Coverage

Triggered by Manuel Romero Medina

View 2 changes

Repository and version

TP07-Angular

main 258c6191

Time started and elapsed

Today at 15:26

4m 51s

Related

0 work items

3 published

Tests and coverage

100% passed

47.34% covered

Jobs

Name	Status	Duration
Build and Test API	Success	2m 12s
Build and Test Angular	Success	1m 55s

#20241012.1 • EmployeeAngular

TP07-Angular

Run new

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

View retention leases

Summary Tests Code Coverage

Summary

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

14

Total tests

+14

14

0

0

Passed

Failed

Others

100%

Pass percentage

100%

4s 143ms

Run duration

+4s 143ms

0

Tests not reported

Bug Link

Filter by test or run name

Test run Column Options

Tags Test file Owner Aborted (+1)

Summary Tests <u>Code Coverage</u>				
Build configuration: empty Build platform: empty				
Folder view				
47.34%	10	169	80	
Line Coverage	Modules	Total Lines	Covered Lines	
Filter by module name				
Modules	Covered Lines	Total Lines	Line Coverage ↑	Coverage Chart
D:\a\1\EmployeeCrudApi\EmployeeCrudApi\Controllers\EmployeeController.cs	48	52	92.31%	<div><div></div></div>
D:\a\1\EmployeeCrudApi\EmployeeCrudApi\Data\ApplicationDbContext.cs	4	4	100.00%	<div><div></div></div>
D:\a\1\EmployeeCrudApi\EmployeeCrudApi\Models\Employee.cs	3	3	100.00%	<div><div></div></div>
D:\a\1\EmployeeCrudApi\EmployeeCrudApi\Program.cs	0	24	0.00%	<div><div></div></div>
src/app/addemployee/addemployee.component.ts	8	45	17.78%	<div><div></div></div>
src/app/app.component.ts	2	2	100.00%	<div><div></div></div>
src/app/employee.model.ts	3	3	100.00%	<div><div></div></div>
src/app/employee.service.ts	6	13	46.15%	<div><div></div></div>
src/app/employee/employee.component.ts	5	22	22.73%	<div><div></div></div>
src/app/employee/employee.component.ts	1	1	100.00%	<div><div></div></div>

7.2 Agregar Análisis Estático de Código con SonarCloud:

Vínculo primero DevosAzure con SonarCloud

1 Your Azure DevOps information

Azure DevOps organization name *

manuromeromedina12345

Copy/paste here the name of your existing Azure DevOps organization.

Personal Access Token *

hbcoq2hsczbib67e56z3ysftefkznmuitt75lcmpd3ftgkbjmhaj

Create a personal access token in your Azure DevOps [user settings](#).
Check the following scope: Code (Read & Write).

For more information about Personal Access Token, please read the [documentation](#).

Continue

2 Import organization details

Import  manuromeromedina12345 into a SonarCloud organization

Name

manurmTP07

Up to 255 characters

Key *

manurmt07

Organization key must start with a lowercase letter or number, followed by lowercase letters, numbers or hyphens, and must end with a letter or number. Maximum length: 255 characters.

> Add additional info

Project Key ?

manuromeromedina12345_TP07-Angular

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

manuromeromedina12345_TP07-Angular

Extensions

Installed Requested Shared



SonarCloud by SonarSource

Detect bugs, vulnerabilities and code smells across project branches and pull requests.



Manuel Romero Medina

Profile Security Notifications Organizations Appearance

Security

If you want to enforce security by not providing credentials of a real SonarCloud user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Enter Token Name

Generate Token



New token "token1" has been created. Make sure you copy it now, you won't be able to see it again!

05e2a5928ee86197c4e0d59a8ad99d399c7263ea



Token SonarCloud: 2f536ab49e791c3ac07be45e32efed466c7c2a4a

Service connections

New service connection

Filter by keywords

Created by

SonarCloud

- Colocaremos una tarea de SonarCloud ANTES de nuestra tarea de Build

```
- task: SonarCloudPrepare@2
  inputs:
    SonarCloud: 'SonarCloud' #Nombre de nuestra Service Connection a SonarCloud
    organization: 'ingsoft3ucc' #Nombre de nuestra organizacion SonarCloud
    scannerMode: 'MSBuild'
    projectKey: 'manuromeromedina12345_TP07-IngSoft' #Key de nuestro proyecto en SonarCloud
    projectName: 'TP07-Angular' #Nombre de nuestro proyecto en SonarCloud
    displayName: 'Prepare SonarCloud'
```

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

```
- task: SonarCloudAnalyze@2
  inputs:
    jdkversion: 'JAVA_HOME_17_X64'

- task: SonarCloudPublish@2
  inputs:
    pollingTimeoutSec: '300'
```

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

#20241013.2 • Update azure-pipelines.yml for Azure Pipelines

TP07-Angular

Run new

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

View retention leases

Summary

Tests

Extensions

Code Coverage

SonarCloud Analysis Report

Quality Gate passed (TP07-Angular)

Issues

0 New issues

0 Accepted issues

Measures

0 Security Hotspots

[See analysis details on SonarCloud](#)

My Projects

My Issues

Explore

Q

🔔

?

+

🔗

manurmTP07 > TP07-Angular > Overview

T

TP07-Angular

No tags

Last analysis 13 Oct 2024 146 Lines of Code

● C#

Main Branch Status

Quality Gate ?

Passed

Enjoy your sparkling clean code!

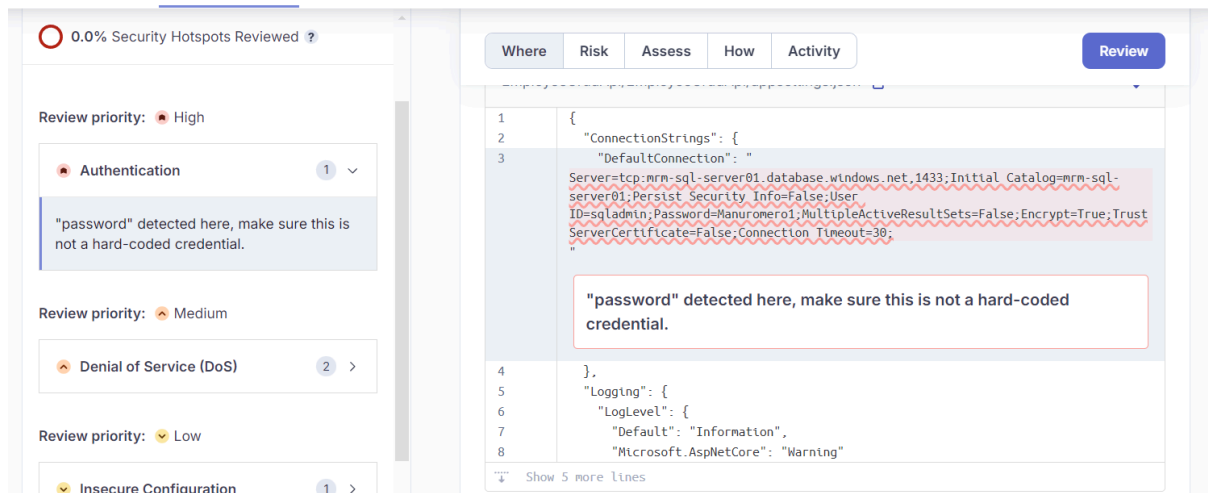
Main Branch Evolution since 8 minutes ago

9 Issues

Issues

Coverage

Duplications



- **Summary:** Un resumen general de la calidad del código. Es como un diagnóstico rápido.
- **Issues:** Aca se encuentra una lista detallada de todos los problemas que SonarCloud ha detectado en el código, desde errores simples hasta potenciales vulnerabilidades de seguridad.
- **Security Hotspots:** Esta sección se enfoca específicamente en los puntos del código que son más vulnerables a ataques.
- **Measures:** Aca se muestran métricas como la cobertura de pruebas, la complejidad del código y otros indicadores que ayudan a evaluar la calidad general del proyecto.
- **Code:** Permite explorar el código directamente desde la interfaz de SonarCloud, lo que facilita la identificación y corrección de problemas.
- **Activity:** Muestra un historial de los cambios realizados en el código y cómo han afectado la calidad del mismo.

4.3 Pruebas de Integración con Cypress:

En el directorio raíz de nuestro proyecto Angular instalar el siguiente paquete: `npm install cypress --save-dev`

```
C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\AngularEmployee\EmployeeCrudAngular>npm install cypress --save-dev
up to date, audited 1055 packages in 4s
132 packages are looking for funding
  run `npm fund` for details
4 low severity vulnerabilities
To address all issues, run:
  npm audit fix
Run `npm audit` for details.
```

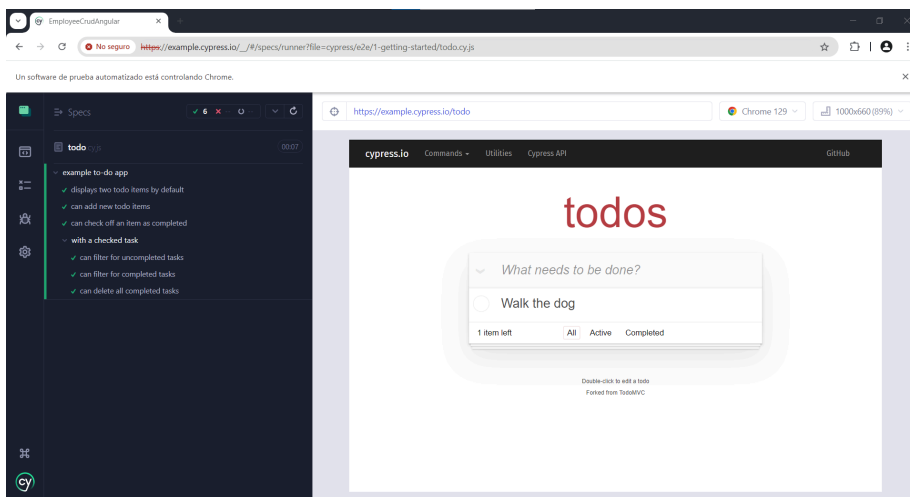
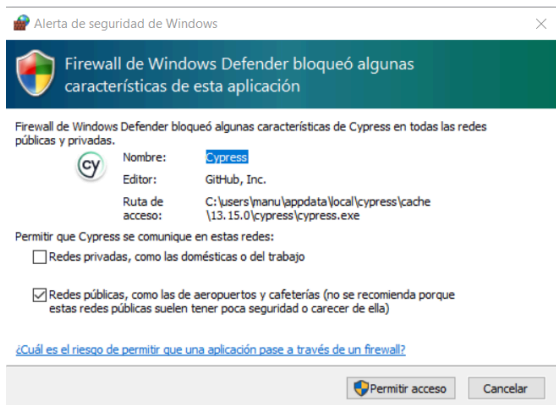
Abrir Cypress: `npm run cypress:open`

```
C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\AngularEmployee\EmployeeCrudAngular>npm run cypress:open
It looks like this is your first time using Cypress: 13.15.0

✓ Verified Cypress! C:\Users\Manu\AppData\Local\Cypress\Cache\13.15.0\Cypress

Opening Cypress...

DevTools listening on ws://127.0.0.1:32143/devtools/browser/57e23113-a652-45bf-996c-de0faab83471
Missing baseUrl in compilerOptions. tsconfig-paths will be skipped
```



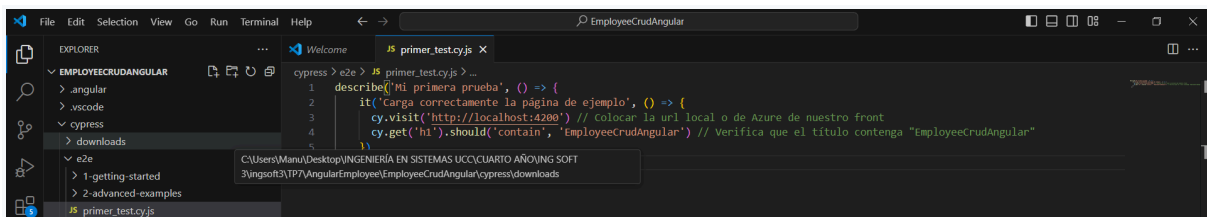
Hacemos un dir

```
C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\AngularEmployee\EmployeeCrudAngular>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: DE26-4E0F

Directorio de C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\AngularEmployee\EmployeeCrudAngular
13/10/2024  15:52    <DIR>        .
13/10/2024  15:52    <DIR>        ..
12/10/2024  15:06    <DIR>        .angular
29/09/2024  19:10             290 .editorconfig
29/09/2024  19:10             590 .gitignore
12/10/2024  15:06    <DIR>        .vscode
01/10/2024  18:13             2.984 angular.json
13/10/2024  15:53    <DIR>        cypress
13/10/2024  15:52             175 cypress.config.ts
01/10/2024  20:08             1.080 karma.conf.js
13/10/2024  15:50    <DIR>        node_modules
13/10/2024  15:50             536.082 package-lock.json
13/10/2024  15:47             1.246 package.json
29/09/2024  19:10             1.100 README.md
12/10/2024  15:08    <DIR>        src
12/10/2024  15:08    <DIR>        test-results
29/09/2024  19:10             277 tsconfig.app.json
29/09/2024  19:10             889 tsconfig.json
29/09/2024  19:10             287 tsconfig.spec.json
          11 archivos             545.000 bytes
           8 dirs  58.327.031.808 bytes libres
```


Crear nuestra primera prueba navegando a nuestro front. En la carpeta cypress/e2e, crear un archivo con el nombre primer_test.js y agregar el siguiente código para probar la página de inicio de nuestro front:

```
describe('Mi primera prueba', () => {
  it('Carga correctamente la página de ejemplo', () => {
    cy.visit('https://as-crud-web-api-qa.azurewebsites.net/') // Colocar la url local o de Azure de nuestro front
    cy.get('h1').should('contain', 'EmployeeCrudAngular') // Verifica que el título contenga "EmployeeCrudAngular"
  })
})
```



Pongo localhost:4200 porque es donde se nos activa el proyecto.

- Correr nuestra primera prueba Si está abierta la interfaz gráfica de Cypress, aparecerá el archivo primer_test.cy.js en la lista de pruebas. Clic en el archivo para ejecutar la prueba.

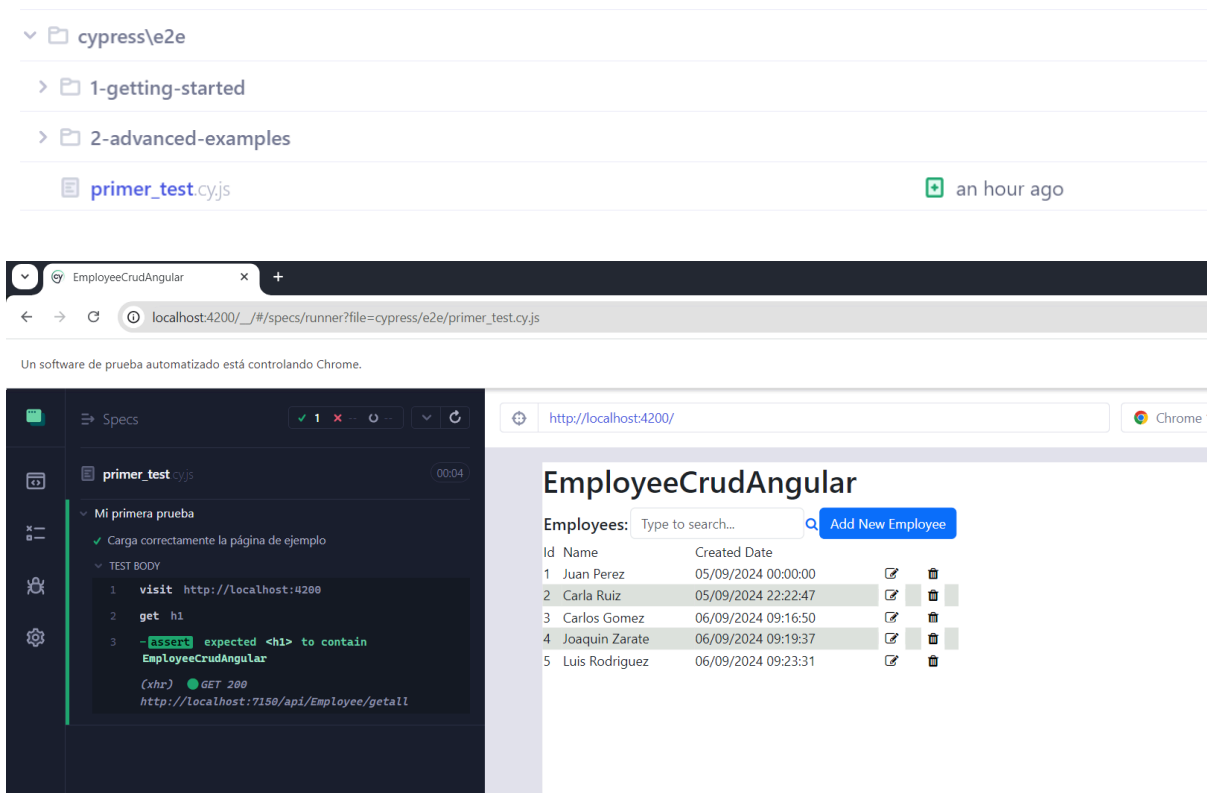
```
C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\AngularEmployee\EmployeeCrudApi>dotnet run --urls "http://localhost:7150"
Compilando...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:7150
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\AngularEmployee\EmployeeCrudApi\EmployeeCrudApi
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (143ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
      SELECT [e].[Id], [e].[CreatedDate], [e].[Name]
      FROM [Employees] AS [e]
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (73ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
      SELECT [e].[Id], [e].[CreatedDate], [e].[Name]
      FROM [Employees] AS [e]
```

```
C:\Users\Manu\Desktop\INGENIERÍA EN SISTEMAS UCC\CUARTO AÑO\ING SOFT 3\ingsoft3\TP7\AngularEmployee\EmployeeCrudAngular>ng serve -o
Initial chunk files | Names | Raw size
styles.css | styles | 313.36 kB |
polyfills.js | polyfills | 88.12 kB |
main.js | main | 16.82 kB |
| Initial total | 418.30 kB

Application bundle generation complete. [4.697 seconds]
▲ [WARNING] Expected ";" but found end of file [css-syntax-error]

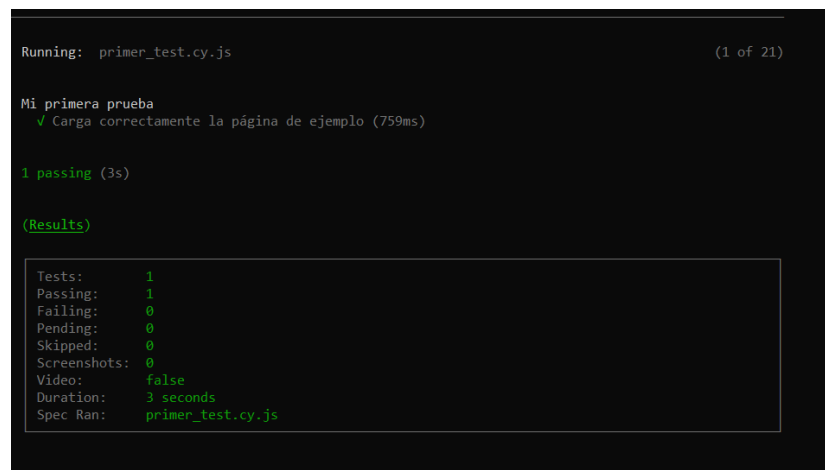
src/styles.css:2:47:
  2 | @import '../node_modules/ngx-toastr/toastr.css'
    |                                         ^
    |                                         ;

Watch mode enabled. Watching for file changes...
  Local: http://localhost:4200/
  press h + enter to show help
```



Es posible ejecutar las pruebas sin la interfaz gráfica, es decir, en modo "headless". Esto se hace mediante el comando:

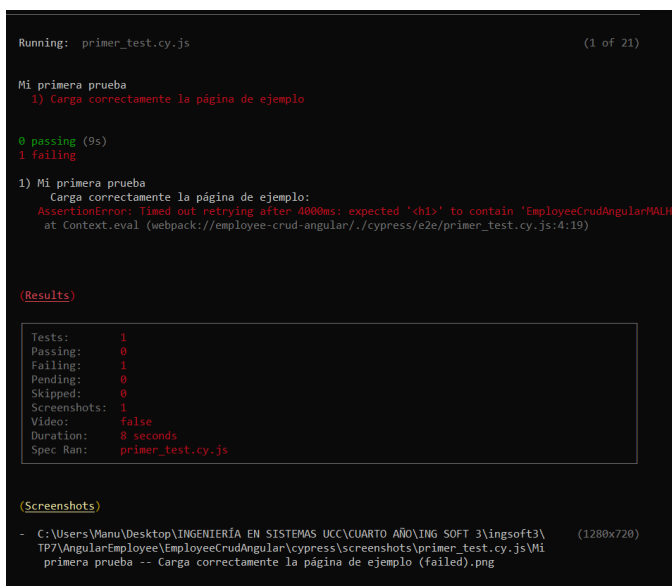
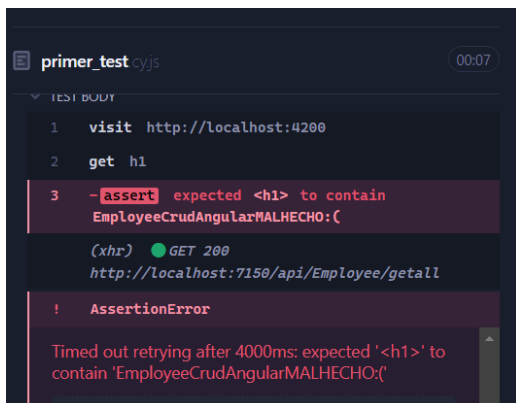
```
npx cypress run
```



Modificar nuestra prueba para que falle.

- Editamos el archivo primer_test.cy.js y hacemos que espere otra cosa en el título
- Ejecutamos cypress en modo headless

```
JS primer_test.cy.js X
cypress > e2e > JS primer_test.cy.js > ...
1 describe('Mi primera prueba', () => {
2   it('Carga correctamente la página de ejemplo', () => {
3     cy.visit('http://localhost:4200') // colocar la url local o de Azure de nuestro front
4     cy.get('h1').should('contain', 'EmployeeCrudAngularMALHECHO:') // Verifica que el título contenga "EmployeeCrudAngular"
5   })
6 })
```



Editamos el archivo cypress.config.ts incluyendo la propiedad experimentalStudio en true y la configuración de reportería.

```
import { defineConfig } from "cypress";
```

```
export default defineConfig({
```

```
  e2e: {
```

```
    setupNodeEvents(on, config) {
```

```
      // implement node event listeners here
```

```
    },
```

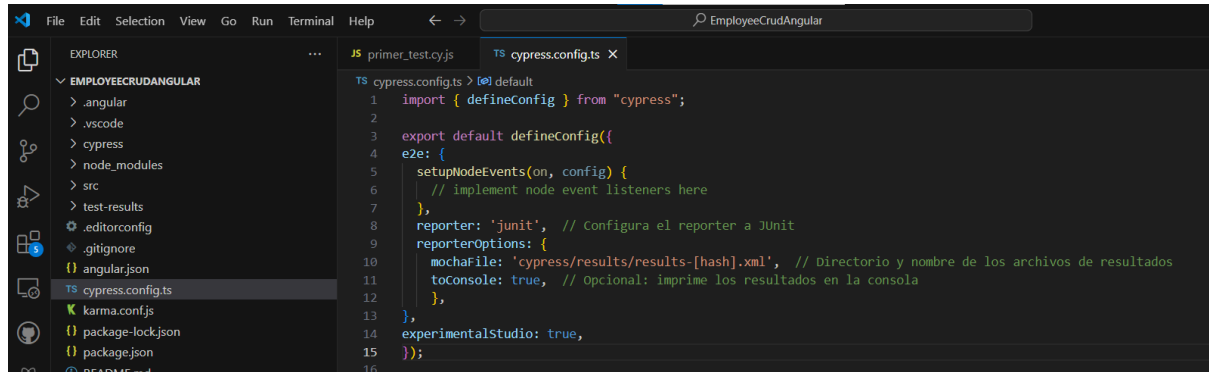
```
    reporter: 'junit', // Configura el reporter a JUnit
```

```
    reporterOptions: {
```

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

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

```
    },  
  },  
  experimentalStudio: true,  
});
```



```
TS cypress.config.ts > [0] default  
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 de resultados  
11       toConsole: true, // Opcional: imprime los resultados en la consola  
12     },  
13   },  
14   experimentalStudio: true,  
15 });  
16
```

Creamos nuevo empleado:

EmployeeCrudAngular

Employees: Q 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			
6	ManuR	13/10/2024 17:22:40			
7	Manu Rome	13/10/2024 17:24:35			
8	Manucho	13/10/2024 17:25:45			
9	Man	13/10/2024 17:32:04			

```
1 visit http://localhost:4200
2 get h1
3 -assert expected <h1> to contain EmployeeCrudAngular
4 get .btn
5 -click
  (new url) http://localhost:4200/addemployee
6 get .form-control
7 -clear Manu Rome
8 get .form-control
9 -type Manu Rome
0 get .form-control
1 -clear Manuch
2 get .form-control
3 -type Manucho
4 get .btn
5 -click
  (xhr) POST 400
  http://localhost:7150/api/Employee/create
```

Hacemos prueba de editar un empleado

- Creamos en cypress/e2e/ un archivo editEmployee_test.cy.js con el siguiente contenido, guardamos y aparecerá en Cypress:

« TP7 > AngularEmployee > EmployeeCrudAngular > cypress > e2e					Buscar en e2e
	Nombre	Fecha de modificación	Tipo	Tamaño	
	1-getting-started	13/10/2024 15:53	Carpeta de archivos		
	2-advanced-examples	13/10/2024 15:53	Carpeta de archivos		
	editEmployee_test.cy	13/10/2024 17:41	Archivo JavaScript	0 KB	
	primer_test.cy	13/10/2024 17:26	Archivo JavaScript	1 KB	

```
JS primer_test.cy.js JS editEmployee_test.cy.js X TS cypress.config.ts
cypress > e2e > JS editEmployee_test.cy.js > describe('editEmployeeTest') callback > it('Carga correctamente la página de ejemplo') callback
1 describe('editEmployeeTest', () => {
2   it('Carga correctamente la página de ejemplo', () => {
3     cy.visit('https://localhost:4200') // Colocar la url local o de Azure de nuestro front
4   })
5 })
```

Debug

Settings

- spies_stubs_clocks.cy.js
- storage.cy.js
- traversal.cy.js
- utilities.cy.js
- viewport.cy.js
- waiting.cy.js
- window.cy.js
- editEmployee_test.cy.js
- primer_test.cy.js

Specs

1

editEmployee_test cy.js 354ms

editEmployeeTest

✓ Carga correctamente la página de ejemplo

TEST BODY

1 visit http://localhost:4200

(xhr) GET 200

http://localhost:7150/api/Employee/getall

