

# Github Actions - Práctica

En este tutorial vamos a conseguir que funcione GitHub Actions para crear nuestra estrategia de integración continua.

Trabajar en equipo es complicado, y más si la organización es muy grande.

Es muy posible que queramos asegurarnos que el código que tenemos en producción se compila, está verificada su sintaxis y reglas de codificación, está correctamente empaquetado, se ejecutan las pruebas automáticas satisfactoriamente y despliega automáticamente.

De ese modo, cada vez que alguien suba código al repositorio, podemos asegurarnos que cumple todos estos criterios (o nos envía un correo diciendo que alguien se lo está cargando de modo inmediato) y podemos etiquetar, crear una rama de producción y desplegar remotamente (o lo que se nos ocurra).

Queremos que cada vez que subamos código al repositorio este se compile, verifique la sintaxis, formatee y se pasen los test. Vamos a hacerlo todo con un proyecto desde cero para tenerlo para el futuro. Si bien los comandos porque pueden cambiar, lo importante es el concepto.

Lo primero que vamos a hacer es crear un nuevo repositorio llamado **practica-github-actions**.

Creamos repositorio en nuestro GitHub.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

### Repository template

Start your repository with a template repository's contents.

No template ▾

Owner \*



Repository name \*

Daniel-MundosE

/ practica-github-actions



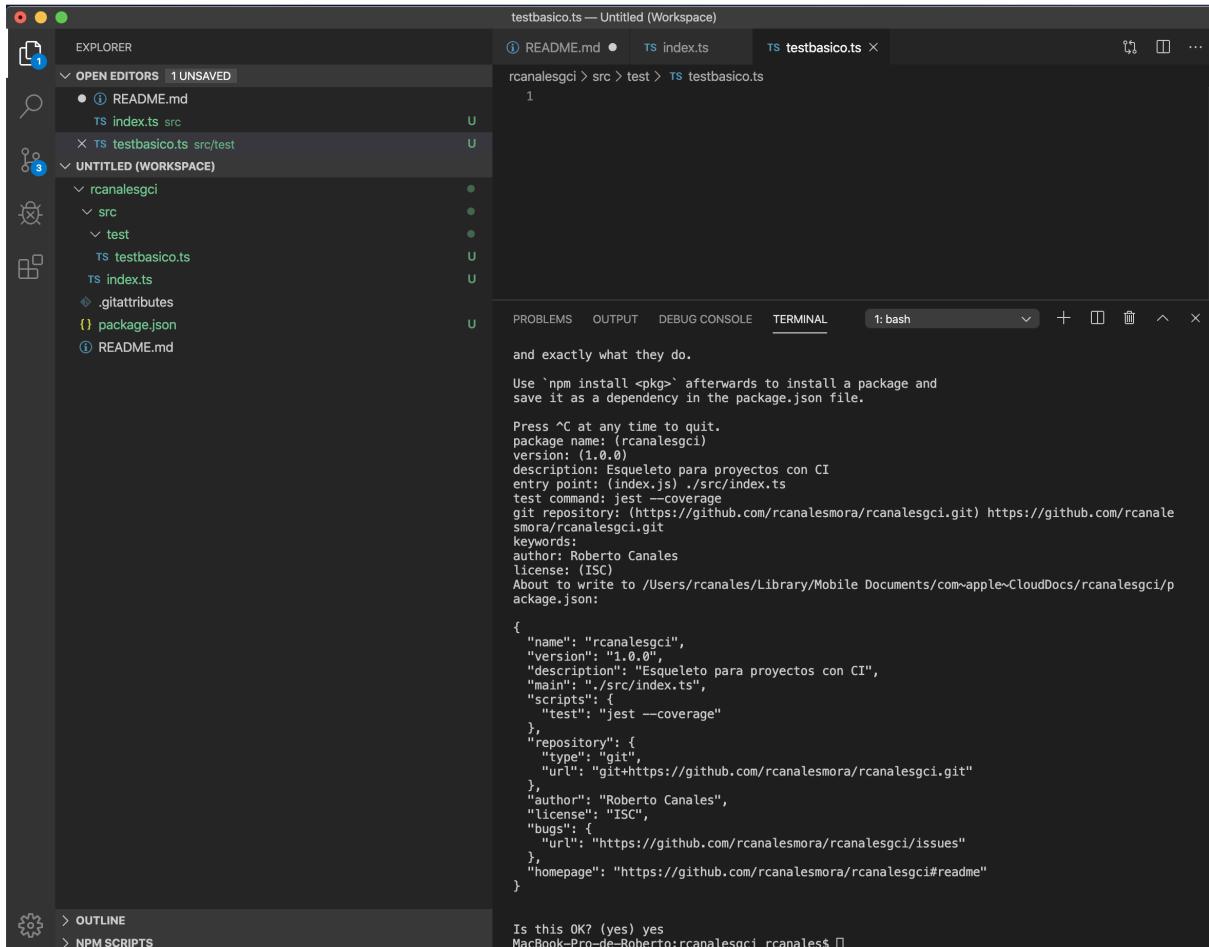
Great repository names are short and descriptive. Is practica-github-actions available? How about **legendary-goggles**?

Luego en Git y creamos el repositorio.

```
mkdir practica-github-actions
cd practica-github-actions
git init
git add -A
git commit -m 'Primer Commit'
git remote add origin git@github.com:[usuario-Git]/practica-github-actions.git
git push -u -f origin master
```

Vamos a MS Visual Studio Code. Comprobamos que en la carpeta creada en local está el fichero .gitattributes.

Creamos un directorio src y un test dentro. He creado un fichero llamado testbasico.ts. Ojo que luego me he arrepentido y los patrones por defecto asignan al nombre de los test “.test.\*” así que atentos luego que lo voy a cambiar porque si no tendría que cambiar el patrón y es complicarse la vida innecesariamente.



```
testbasico.ts — Untitled (Workspace)
① README.md • TS index.ts TS testbasico.ts
rcanalesgci > src > test > TS testbasico.ts
1
and exactly what they do.

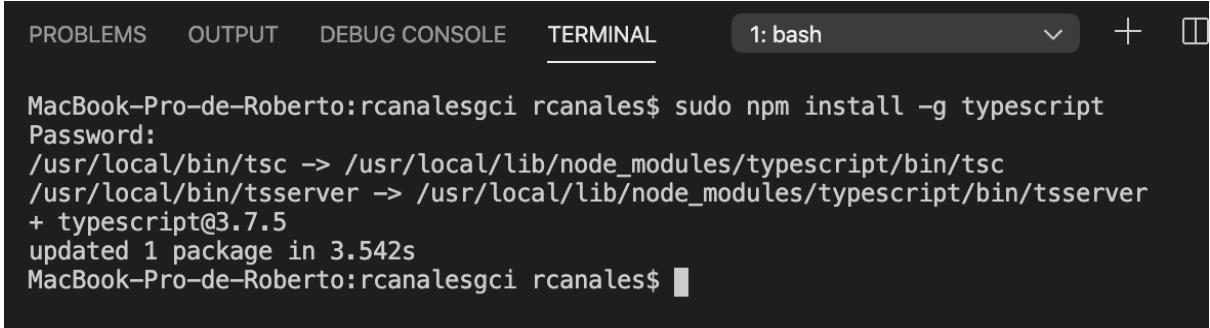
Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (rcanalesgci)
version: (1.0.0)
description: Esqueleto para proyectos con CI
entry point: (index.js) ./src/index.ts
test command: jest --coverage
git repository: (https://github.com/rcañalesmora/rcañalesgci.git) https://github.com/rcañale
smora/rcañalesgci.git
keywords:
author: Roberto Canales
license: (ISC)
About to write to /Users/rcañales/Library/Mobile Documents/com~apple~CloudDocs/rcañalesgci/p
ackage.json:
{
  "name": "rcañalesgci",
  "version": "1.0.0",
  "description": "Esqueleto para proyectos con CI",
  "main": "./src/index.ts",
  "scripts": {
    "test": "jest --coverage"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/rcañalesmora/rcañalesgci.git"
  },
  "author": "Roberto Canales",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/rcañalesmora/rcañalesgci/issues"
  },
  "homepage": "https://github.com/rcañalesmora/rcañalesgci#readme"
}

Is this OK? (yes) yes
MacBook-Pro-de-Roberto:rcañalesgci_rcañales$
```

Instalo la última versión estable de TypeScript

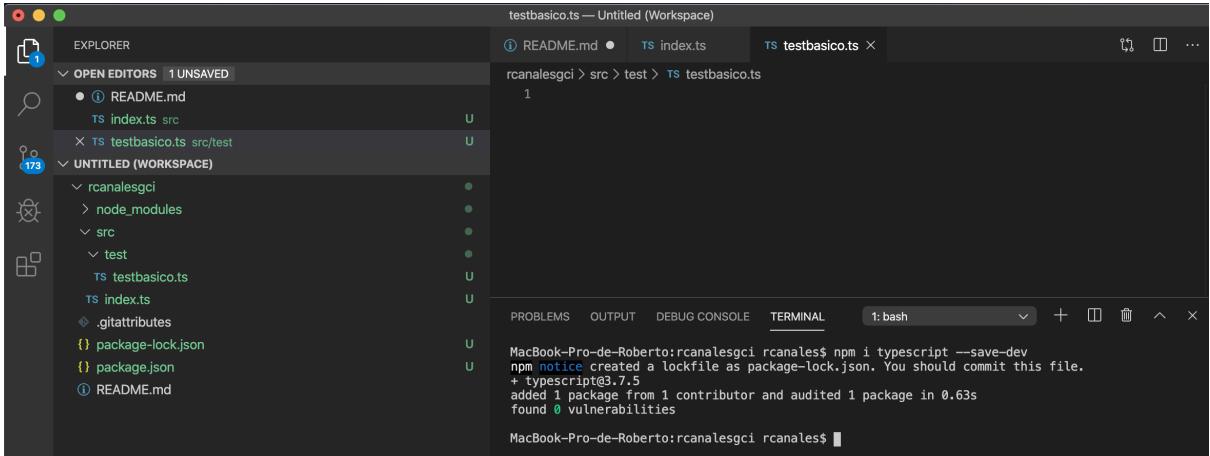
```
sudo npm install -save-dev typescript
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash + [ ]  
MacBook-Pro-de-Roberto:rcanalesgci rcanales$ sudo npm install -g typescript  
Password:  
/usr/local/bin/tsc -> /usr/local/lib/node_modules/typescript/bin/tsc  
/usr/local/bin/tsserver -> /usr/local/lib/node_modules/typescript/bin/tsserver  
+ typescript@3.7.5  
updated 1 package in 3.542s  
MacBook-Pro-de-Roberto:rcanalesgci rcanales$
```

Instalamos TypeScript localmente en el proyecto

```
npm install --save-dev typescript
```



The screenshot shows the VS Code interface with the following details:

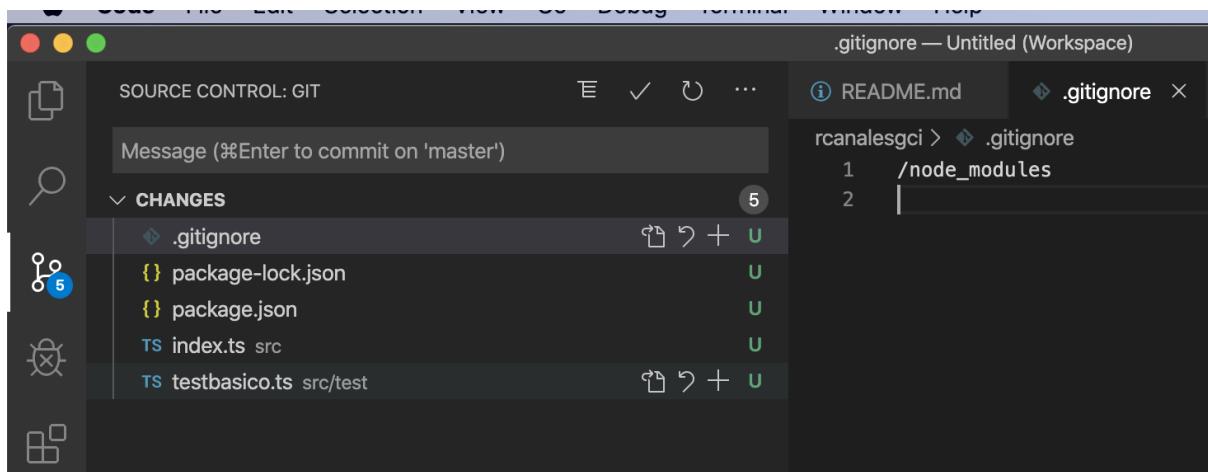
- EXPLORER:** Shows the project structure:
  - OPEN EDITORS: README.md (U), index.ts (U)
  - UNTITLED (WORKSPACE): rcanalesgci (U), node\_modules (U), src (U), test (U), testbasico.ts (U), index.ts (U), .gitattributes (U), package-lock.json (U), package.json (U), README.md (U)
- TERMINAL:** Shows the command `npm install --save-dev typescript` being run and its output:

```
testbasico.ts — Untitled (Workspace)
① README.md ● TS index.ts TS testbasico.ts ×
rcanalesgci > src > test > TS testbasico.ts
1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash + [ ]  
MacBook-Pro-de-Roberto:rcanalesgci rcanales$ npm i typescript --save-dev  
npm notice created a lockfile as package-lock.json. You should commit this file.  
+ typescript@3.7.5  
added 1 package from 1 contributor and audited 1 package in 0.63s  
found 0 vulnerabilities  
MacBook-Pro-de-Roberto:rcanalesgci rcanales$
```

Sacamos el directorio `/node_modules` de GIT para no subir los miles de ficheros con módulos y dependencias a GitHub.

Esto se hace creando el fichero `.gitignore`

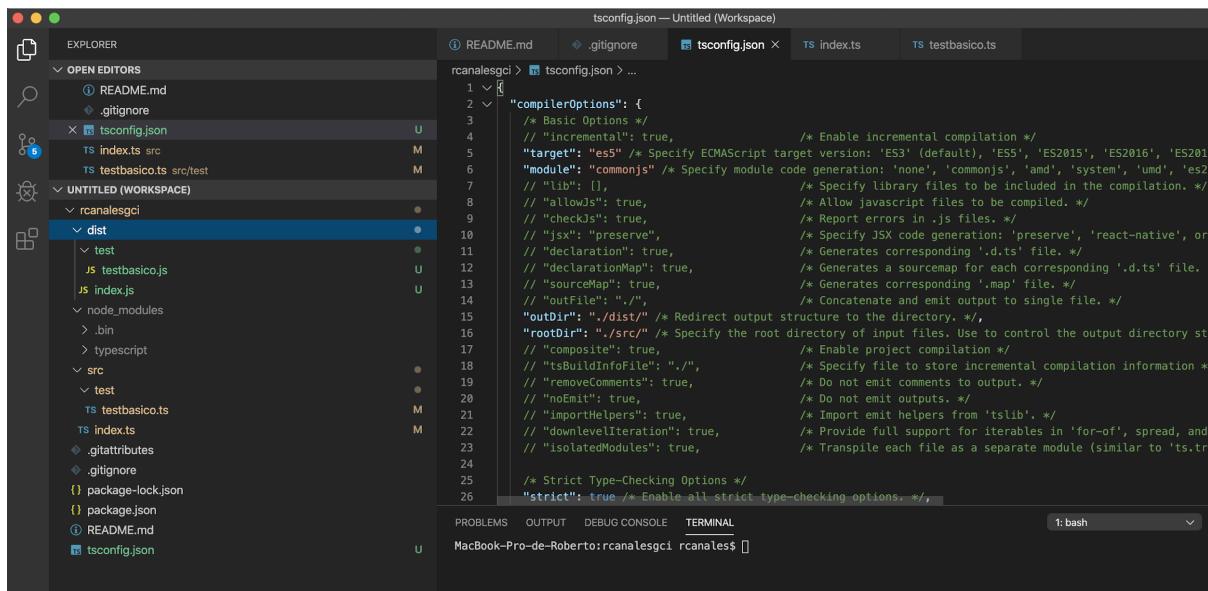


Creamos el fichero tsconfig.json a través de

tsc --init

Modificamos el fichero tsconfig.json para añadir el directorio de las fuentes y la salida.

Ponemos el outDir y rootDir.



Instalamos ahora el sistema de pruebas Jest. Usamos el comando

npm install --save-dev jest.

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
1: bash + □

MacBook-Pro-de-Roberto:rcanalesgci rcanales$ npm install --save-dev jest
npm WARN deprecated left-pad@1.3.0: use String.prototype.padStart()

> fsevents@1.2.11 install /Users/rcanales/Library/Mobile Documents/com~apple~CloudDocs/rcanalesgci/node_modules/fsevents
> node-gyp rebuild

  SOLINK_MODULE(target) Release/.node
  CXX(target) Release/obj.target/fse/fsevents.o
  SOLINK_MODULE(target) Release/fse.node
+ jest@24.9.0
added 528 packages from 375 contributors and audited 882112 packages in 19.348s

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

  found 0 vulnerabilities

MacBook-Pro-de-Roberto:rcanalesgci rcanales$
```

Inicializamos el sistema de pruebas con

`jest --init`

para generar el fichero de configuración.

```
MacBook-Pro-de-Roberto:rcanalesgci rcanales$ jest --init
✓ It seems that you already have a jest configuration, do you want to override it? ... yes
The following questions will help Jest to create a suitable configuration for your project
✓ Choose the test environment that will be used for testing > jsdom (browser-like)
✓ Do you want Jest to add coverage reports? ... yes
✓ Automatically clear mock calls and instances between every test? ... yes
Configuration file created at /Users/rcanales/Library/Mobile Documents/com~apple~CloudDocs/rcanalesgci/jest.config.js
MacBook-Pro-de-Roberto:rcanalesgci rcanales$
```

Recordar de renombrar el fichero del test a `suma.test.js`.

Ahora con el comando `jest` ya lo tenemos funcionado.

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files like `README.md`, `.gitignore`, `tsconfig.json`, `jest.config.js`, `index.js`, `index.ts`, `summa.test.ts`, `summa.test.js`, and `package.json`.
- Code Editor:** Displays the `suma.test.ts` file containing Jest test cases for sum operations.
- Terminal:** Shows the output of the test run, indicating 1 test passed, 3 total tests, and a time of 0.95s.
- Status Bar:** Shows the current branch as `master*` and other status indicators.

Ahora instalamos Parcel. Creamos un fichero index.html e index.ts.

Usamos el comando `sudo npm install -g parcel-bundler`

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files like `index.html`, `index.ts`, `package.json`, and `tsconfig.json`.
- Code Editors:** One editor shows `index.html` with some basic HTML code, and the other shows `index.ts` with a simple TypeScript function.
- Terminal:** Shows the command `sudo npm install -g parcel-bundler` being run, along with the resulting npm output and error messages related to file permissions and dependencies.
- Status Bar:** Shows the current branch as `master*` and other status indicators.

En package.json creamos los scripts para desde npm para simplificar las invocaciones desde el entorno y luego usarlos desde Github Actions.

```
rcanalesgci > {} package.json > {} bugs
1   {
2     "name": "rcanalesgci",
3     "version": "1.0.0",
4     "description": "Esqueleto para proyectos con CI",
5     "main": "./dist/index.js",
6     "scripts": {
7       "start": "parcel src/index.html",
8       "build": "parcel build src/index.html",
9       "test": "jest --coverage",
```

Con npm podemos ejecutar cada Script desde la línea de comando tipo npm start

```
MacBook-Pro-de-Roberto:rcanalesgci rcanales$ npm start
> rcanalesgci@1.0.0 start /Users/rcanales/Library/Mobile Documents/com~apple~CloudDocs/rcanalesgci
> parcel src/index.html
Server running at http://localhost:1234
Built in 369ms.
```

Con npm test

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
^C
MacBook-Pro-de-Roberto:rcanalesgci rcanales$ npm test

> rcanalesgci@1.0.0 test /Users/rcanales/Library/Mobile Documents/com~apple~CloudDocs/rcanalesgci
> jest --coverage

ts-jest[config] (WARN) Unable to find the root of the project where ts-jest has
PASS  src/test/suma.test.ts
  ✓ suma sin parametros (2ms)
  ✓ suma con dos parametros
  ✓ suma con parametros variables

  console.log src/index.ts:352
    hola mundo

-----|-----|-----|-----|-----|-----|
File      | % Stmt | % Branch | % Funcs | % Lines | Uncovered Line #s |
-----|-----|-----|-----|-----|-----|
All files |   100  |     100  |     100  |     100  |           |
index.ts  |   100  |     100  |     100  |     100  |           |
-----|-----|-----|-----|-----|-----|
Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        0.879s, estimated 2s
Ran all test suites.
MacBook-Pro-de-Roberto:rcanalesgci rcanales$
```

Npm run-script build

```
MacBook-Pro-de-Roberto:rcanalesgci rcanales$ npm run-script build
MacBook-Pro-de-Roberto:rcanalesgci rcanales$ npm run-script build

> rcanalesgci@1.0.0 build /Users/rcanales/Library/Mobile Documents/com~apple~CloudDocs/rcanalesgci
> parcel build src/index.html

+ Built in 42ms.

dist/src.66b35617.js      1.4 KB  4ms
dist/src.66b35617.js.map    567 B   2ms
dist/index.html            108 B   2ms
MacBook-Pro-de-Roberto:rcanalesgci rcanales$
```

Ahora instalamos eslint en el entorno local con npm install eslint —save-dev

```

MacBook-Pro-de-Roberto:rcanalesgci rcanales$ npm install eslint --save-dev
npm WARN acorn-jsx@5.1.0 requires a peer of acorn@^6.0.0 || ^7.0.0 but none is installed. You must install peer dependencies yourself.

+ eslint@6.8.0
added 62 packages from 53 contributors and audited 882291 packages in 20.114s

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

found 0 vulnerabilities

MacBook-Pro-de-Roberto:rcanalesgci rcanales$ █

```

Inicializamos el fichero `.eslintrc.js` simplemente diciendo las opciones con `eslint —init`

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows the project structure with files like `index.html`, `index.ts`, `summa.test.ts`, `test`, `src`, `node_modules`, and `dist`.
- Terminal View:** Displays the command `./node_modules/.bin/eslint --init` being run, followed by a series of questions about the project setup. The responses are:
  - How would you like to use ESLint? To check syntax and find problems
  - What type of modules does your project use? JavaScript modules (import/export)
  - Which framework does your project use? None of these
  - Does your project use TypeScript? Yes
  - Where does your code run? Browser, Node
  - What format do you want your config file to be in? JavaScript
 The config that you've selected requires the following dependencies:
  - @typescript-eslint/eslint-plugin@latest @typescript-eslint/parser@latest
  - Would you like to install them now with npm? Yes
 The terminal also shows the installation of dependencies, the audit report, and the creation of the `.eslintrc.js` file.
- Status Bar:** Shows the current branch is `master*`, and the file status is `1 master 0`.

Ejecutamos para comprobar que funciona. Lo hace correctamente encontrando que en `index.ts` hay variables que no se usan.

```
MacBook-Pro-de-Roberto:rcanalesgci rcanales$ ./node_modules/.bin/eslint ./src/index.ts
/Users/rcanales/Library/Mobile Documents/com~apple~CloudDocs/rcanalesgci/src/index.ts
  5:5  error  'a' is assigned a value but never used  no-unused-vars
  6:5  error  'b' is assigned a value but never used  no-unused-vars

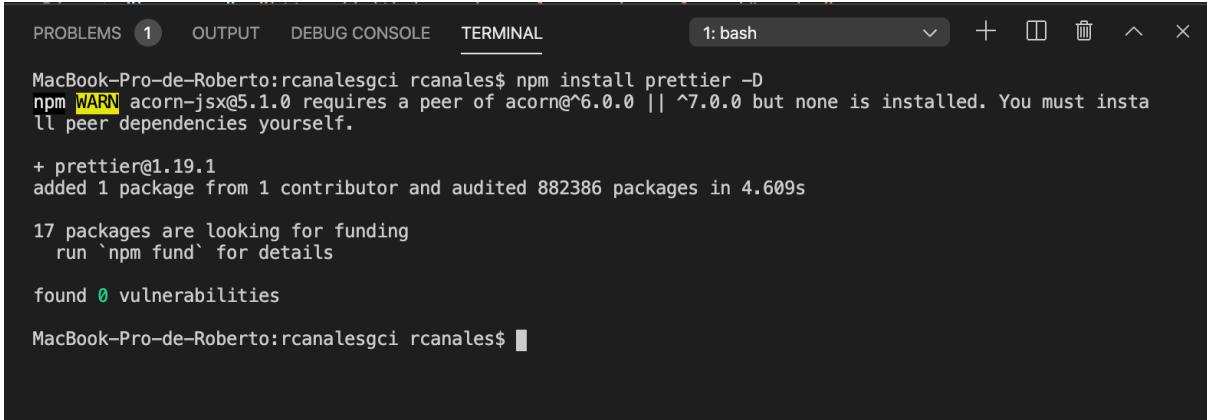
✖ 2 problems (2 errors, 0 warnings)

MacBook-Pro-de-Roberto:rcanalesgci rcanales$
```

Ahora actualizamos los scripts para incluir las opciones de Lint.

```
1  {
2    "name": "rcanalesgci",
3    "version": "1.0.0",
4    "description": "Esqueleto para proyectos con CI",
5    "main": "./dist/index.js",
6    "scripts": {
7      "start": "parcel src/index.html",
8      "build": "parcel build src/index.html",
9      "test": "jest --coverage",
10     "lint": "npm run lint:js && npm run lint:css",
11     "lint:js": "eslint src/**/*.{ts,js}",
12     "lint:css": "eslint \"src/**/*.{ts,js}\"",
```

Solo nos queda ahora instalar Prettier, el formateador. Usamos el comando npm install prettier -D



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
1: bash + □ □ ^ ×
MacBook-Pro-de-Roberto:rcanalesgci rcanales$ npm install prettier -D
npm WARN acorn-jsx@5.1.0 requires a peer of acorn@^6.0.0 || ^7.0.0 but none is installed. You must install peer dependencies yourself.

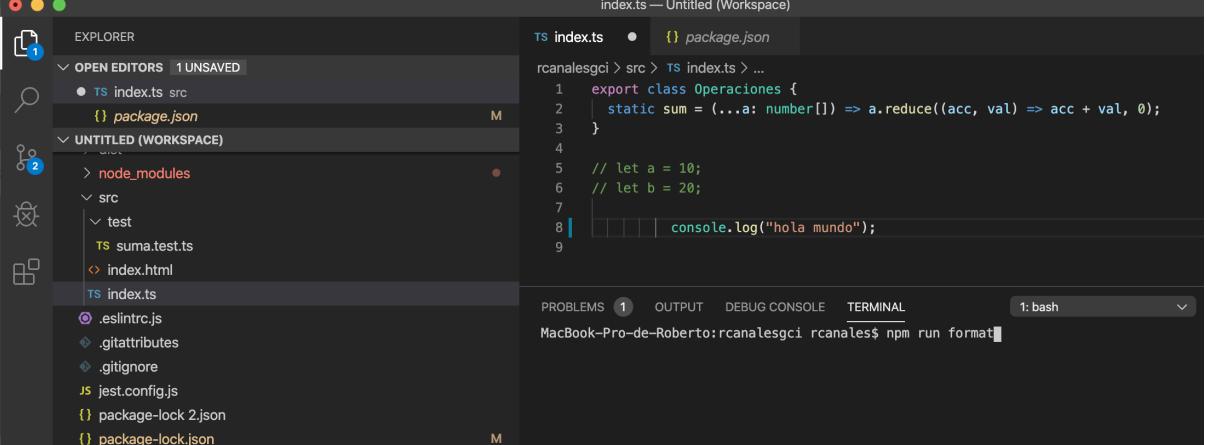
+ prettier@1.19.1
added 1 package from 1 contributor and audited 882386 packages in 4.609s

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

found 0 vulnerabilities

MacBook-Pro-de-Roberto:rcanalesgci rcanales$
```

Deformamos un poco un fichero y ejecutamos con npm run format.



The screenshot shows the VS Code interface. In the Explorer sidebar, there are two open editors: 'index.ts' and 'package.json'. The 'index.ts' editor contains the following code:

```
index.ts — Untitled (Workspace)
rcanalesgci > src > ts index.ts > ...
1 export class Operaciones {
2   static sum = (...a: number[]) => a.reduce((acc, val) => acc + val, 0);
3 }
4
5 // let a = 10;
6 // let b = 20;
7
8 console.log("Hola mundo");
9
```

In the bottom right corner, the terminal window shows the command: 'MacBook-Pro-de-Roberto:rcanalesgci rcanales\$ npm run format'.

Así quedan los comandos. Podemos ahora ejecutar todos los comandos con facilidad y comprobar que formar cambia el fichero al formato deseado.

Veamos nuestro package.json (Creado a partir del de Cesar Alberca <https://github.com/OlympiCSS/web/blob/dev/package.json>). Está simplificado respecto al suyo. Los invito a visitar su proyecto sobre todo por las acciones .yml de GitHub Actions ;-).

JavaScript

```
1      {
2      "name": "practica-github-actions",
3      "version": "1.0.0",
4      "description": "Esqueleto para proyectos con CI",
5      "main": "./dist/index.js",
6      "scripts": {
7          "start": "parcel src/index.html",
8          "build": "parcel build src/index.html",
9          "test": "jest --coverage",
10         "lint": "npm run lint:js && npm run lint:css",
11         "lint:js": "eslint src/**/*.{ts}",
12         "lint:css": "eslint \"src/**/*.{ts}\"",
13         "format": "prettier --write \"src/**/*.{js,jsx,ts,tsx,json,css,md}\""
14     },
15     "repository": {
16         "type": "git",
17         "url": "git+https://github.com/Daniel-MundosE/practica-github-actions.git"
18     }
}
```

```
19     "author": "Daniel Ojcius",
20
21     "license": "ISC",
22
23     "bugs": {
24
25         "url": "https://github.com/Daniel-MundosE/practica-github-actions/issues"
26
27     },
28
29     "homepage":
30         "https://github.com/Daniel-MundosE/practica-github-actions#readme",
31
32     "devDependencies": {
33
34         "@typescript-eslint/eslint-plugin": "^2.17.0",
35
36         "@typescript-eslint/parser": "^2.17.0",
37
38         "eslint": "^6.8.0",
39
40         "jest": "^24.9.0",
41
42         "prettier": "^1.19.1",
43
44         "typescript": "^3.7.5"
45
46     }
47 }
```

Este es el aspecto (abajo a la izquierda) que tienen todos los scripts que hemos creado.

Ahora que tenemos los scripts y la infraestructura básica, vamos a Github Actions y creamos una básica. Elegimos el WorkFlow Node.js

Nos crea un fichero básico .yml.

Cada vez que hagamos un push instalará, construirá y probará.

Cambiamos en nuestro MS Visual Studio Local (podría ser en Github directamente con el editor) cualquier fichero y hacemos commit y push a GitHub.

Comprobamos que no funciona porque nos hemos dejado alguna dependencia.

The screenshot shows a GitHub Actions CI log for a repository named 'Node CI'. The log title is 'Node CI / build (12.x)' and it indicates a failure 25 minutes ago in 43s. The log details the following steps:

- Set up job (3s)
- Run actions/checkout@v1 (1s)
- Use Node.js 12.x (3s)
- npm install, build, and test** (36s)
  - Run npm ci (1s)
  - fsevents@1.2.11 install /home/runnner/work/rkanalesgci/rkanalesgci/node\_modules/fsevents (8s)
  - node-gyp rebuild (10s)
  - make: Entering directory '/home/runnner/work/rkanalesgci/rkanalesgci/node\_modules/fsevents/build' (11s)
  - SOLINK\_MODULE(target) Release/obj.target/.node (13s)
  - COPY Release/.node (14s)
  - make: Leaving directory '/home/runnner/work/rkanalesgci/rkanalesgci/node\_modules/fsevents/build' (15s)
  - added 536 packages in 8.616s (16s)
  - rcanalesgci@1.0.0 build /home/runnner/work/rkanalesgci/rkanalesgci (17s)
  - parcel build src/index.html (19s)
  - Built in 10.58s. (20s)
  - dist/src.d22824ed.js 1.38 KB 7.51s (23s)
  - dist/src.d22824ed.js.map 548 B 2ms (24s)
  - dist/index.html 108 B 2.81s (25s)
  - rcanalesgci@1.0.0 test /home/runnner/work/rkanalesgci/rkanalesgci (27s)
  - jest --coverage (28s)

Annotations: 1 failure

Si se fijan, me da un error relacionado con ts-jest

Con este comando, en nuestro PC, en MS Visual Studio Code, instalamos la parte que nos falta en nuestro entorno de desarrollo

```
npm install -save-dev jest typescript ts-jest @types/jest
```

Hacemos commit y push. Y ya GitHub ejecuta correctamente sin hacer nada, recordar que está enganchado a cada push.

Los invito a que visiten este enlace donde explica cada línea del fichero .yml y <https://dev.to/bnb/an-unintentionally-comprehensive-introduction-to-github-actions-ci-build>

```
- uses: actions/checkout@v1
```

Tells our workflow that we're going to be using the GitHub Action that can be found at `actions/checkout` which maps to the GitHub org/repo at [github.com/actions/checkout]. It's also worth noting that `@v1` which is a tagged and released version that can be found in the [GitHub Releases](#) for the repo.

Relevant docs:

- [actions/checkout](#), an action that checks out your repository to `$GITHUB_WORKSPACE` in the virtual environment.
- [Workflow syntax docs - jobs.<job\\_id>.steps.uses](#)

```
- name: Use Node.js ${{ matrix.node-version }} on ${{ matrix.os }}
```

Bueno, con esto tenemos ya GitHub Actions instalado.

Dejo para próximos tutoriales crear una rama de producción y que, cuando etiquetemos una versión de la rama principal, se transfiera a la de producción y despliegue (ver ficheros de Cesar que ya está resuelto)

Generamos un error voluntario en el código y lo subimos (commit + push) para vez que hace GitHub.

```
rcanalesgci > src > TS index.ts > 📁 Operaciones > 🔑 sum
```

```
1  export class Operaciones {
2      static falloVoluntario sum = (...a: number[]) => a.r
3  }
4
5  // let a = 10;
6  // let b = 20;
7
8  console.log("hola mundo");
9
```

De inmediato recibimos el mensaje de correo electrónico indicando que había fallado la ejecución. Ya tenemos un entorno desatendido e independiente de los desarrolladores y su entorno.

Y podemos ver las consecuencias.

```
▶ ✓ Set up job
▶ ✓ Run actions/checkout@v1
▶ ✓ Use Node.js 12.x
▼ ✘ npm install, build, and test
  41   ● Test suite failed to run
  42
  43     TypeScript diagnostics (customize using `[[jest-config].globals.ts-jest.diagnostics` option):
  44       src/test/suma.test.ts:4:22 - error TS2339: Property 'sum' does not exist on type 'typeof Operaciones'.
  45
  46     4   expect(Operaciones.sum()).toBe(0);
  47         ~~~
  48   src/test/suma.test.ts:8:22 - error TS2339: Property 'sum' does not exist on type 'typeof Operaciones'.
  49
  50     8   expect(Operaciones.sum(1, 2)).toBe(3);
  51         ~~~
  52   src/test/suma.test.ts:12:22 - error TS2339: Property 'sum' does not exist on type 'typeof Operaciones'.
  53
  54     12  expect(Operaciones.sum(1, 2, 3, 4)).toBe(10);
  55         ~~~
  56
  57 -----|-----|-----|-----|-----|
  58 File    | %Stmts | %Branch | %Funcs | %Lines | Uncovered Line #s |
  59 -----|-----|-----|-----|-----|
  60 All files |     0 |      0 |      0 |      0 |                         |
  61
  62 Test Suites: 1 failed, 1 passed, 2 total
  63 Tests:      3 passed, 3 total
  64 Snapshots:  0 total
  65 Time:        2.668s
  66 Ran all test suites.
  67 npm ERR! Test failed. See above for more details.
  68 ##[error]Process completed with exit code 1.

▶ ✓ Complete job
```

Annotations: 1 failure

