

# Herramientas vistas en clase

Descargamos Node.js de <https://nodejs.org/es/> una vez realizada su instalación lo abrimos el cmd y ponemos **npm install** y una vez realizada la instalación de Node.js instalamos typescript usando el comando **npm install -g typescript**.

```
Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.18363.1082]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>npm install
npm WARN saveError ENOENT: no such file or directory, open 'C:\WINDOWS\system32\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'C:\WINDOWS\system32\package.json'
npm WARN system32 No description
npm WARN system32 No repository field.
npm WARN system32 No README data
npm WARN system32 No license field.

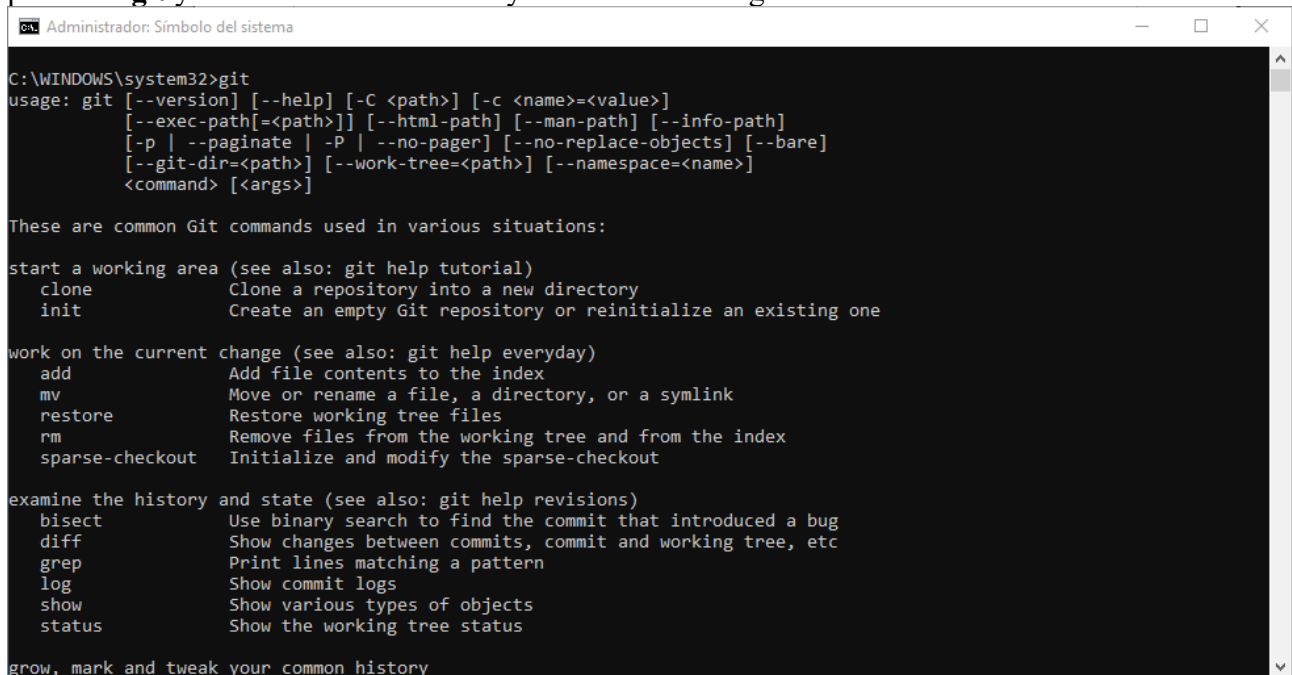
up to date in 0.348s
found 0 vulnerabilities

C:\WINDOWS\system32>
```

```
C:\WINDOWS\system32>npm install -g typescript
C:\Users\elena\AppData\Roaming\npm\tsc -> C:\Users\elena\AppData\Roaming\npm\node_modules\typescript\bin\tsc
C:\Users\elena\AppData\Roaming\npm\tsserver -> C:\Users\elena\AppData\Roaming\npm\node_modules\typescript\bin\tsserver
+ typescript@4.0.3
updated 1 package in 0.859s

C:\WINDOWS\system32>
```

A continuación necesitamos instalar el cliente de Git, <https://git-scm.com/downloads> descargamos y lo instalamos. Una vez instalado comprobamos en el cmd que funciona, para ello abrimos un cmd y ponemos **git** y se nos debe de abrir la ayuda del comando git en el cmd.



```
C:\WINDOWS\system32>git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      <command> [<args>]

These are common Git commands used in various situations:


start a working area (see also: git help tutorial)
  clone                Clone a repository into a new directory
  init                 Create an empty Git repository or reinitialize an existing one

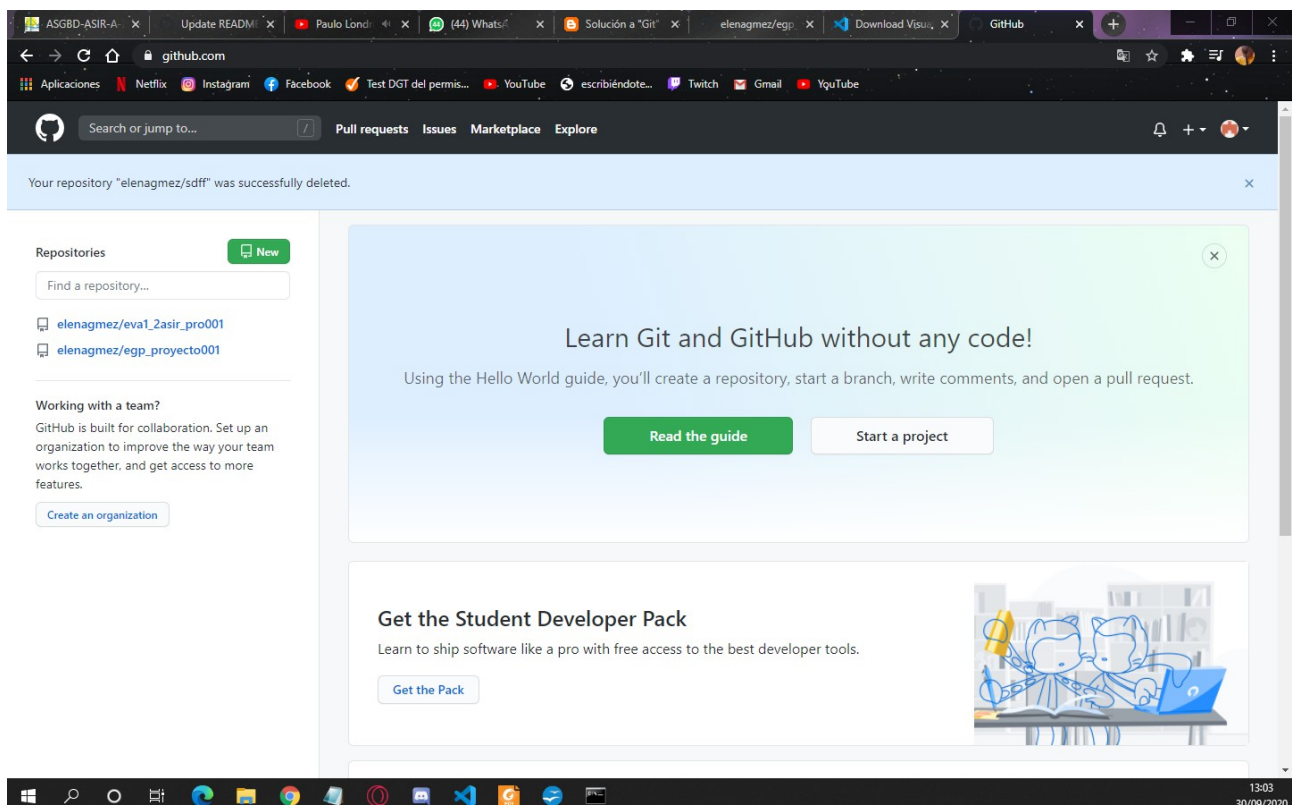

work on the current change (see also: git help everyday)
  add                  Add file contents to the index
  mv                   Move or rename a file, a directory, or a symlink
  restore              Restore working tree files
  rm                   Remove files from the working tree and from the index
  sparse-checkout      Initialize and modify the sparse-checkout


examine the history and state (see also: git help revisions)
  bisect              Use binary search to find the commit that introduced a bug
  diff                Show changes between commits, commit and working tree, etc
  grep                Print lines matching a pattern
  log                 Show commit logs
  show                Show various types of objects
  status              Show the working tree status


grow, mark and tweak your common history
```

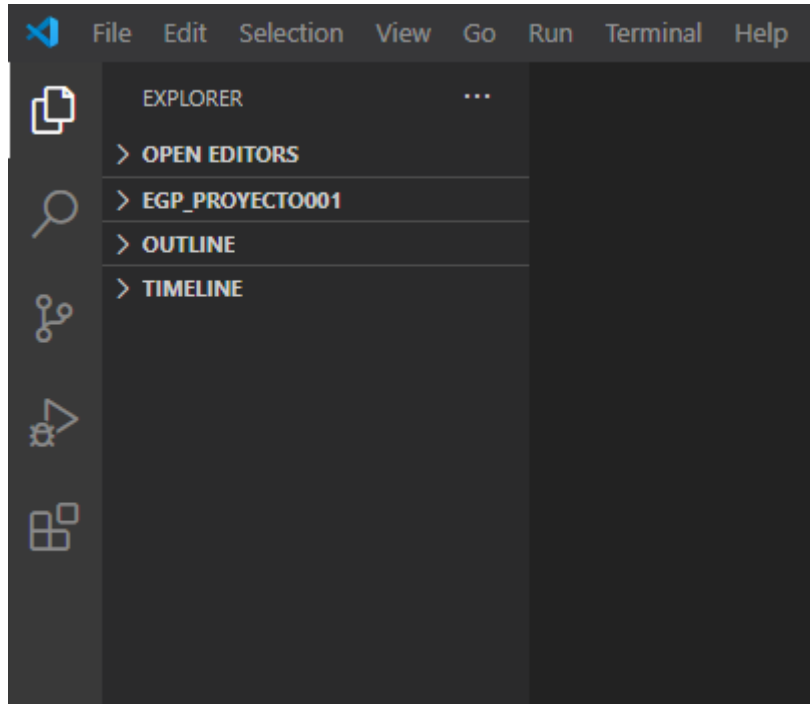
Una vez instalado todo lo anterior necesitamos un editor de código, usaremos Visual Studio Code, <https://code.visualstudio.com/download>.

Una vez tengamos todo lo anterior ya casi estamos listos para empezar nuestro primer proyecto en GitHub pero primero tenemos que crearnos una cuenta <https://github.com/>, una vez creada tenemos que crear un nuevo repositorio para poder subir ahí nuestro proyecto de GitHub.

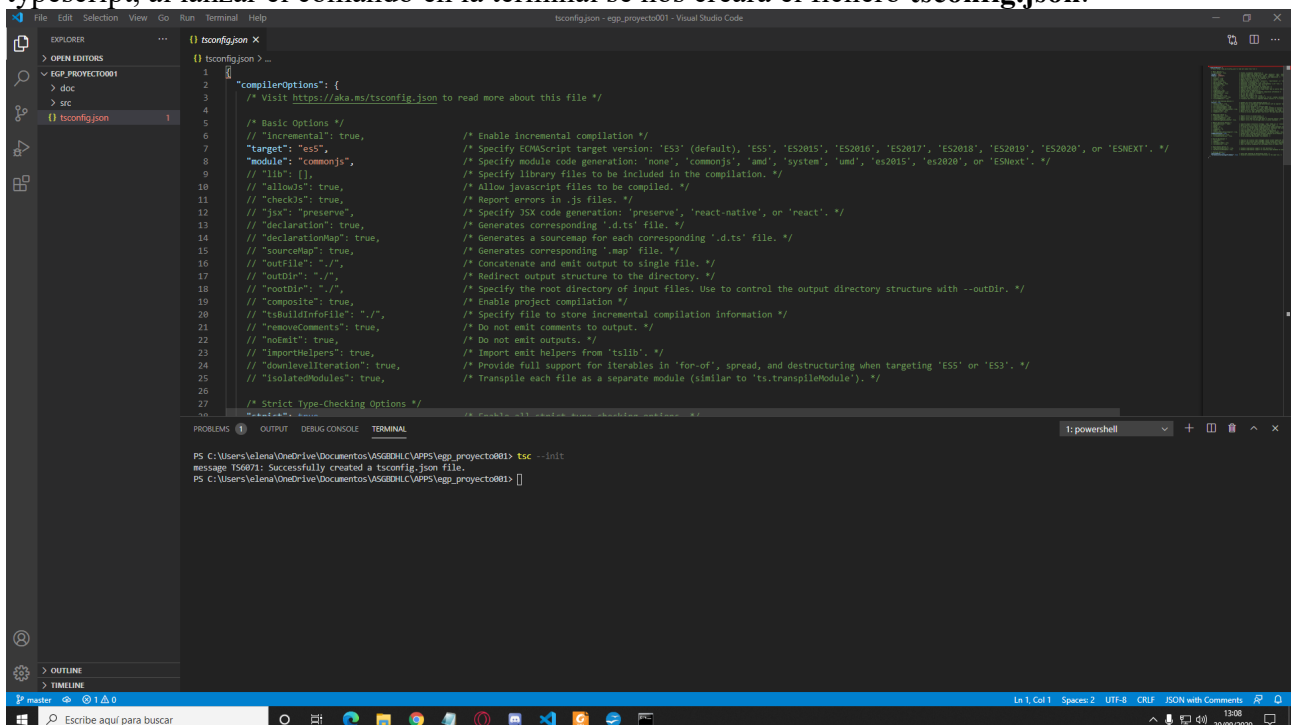


# Crear un proyecto GitHub

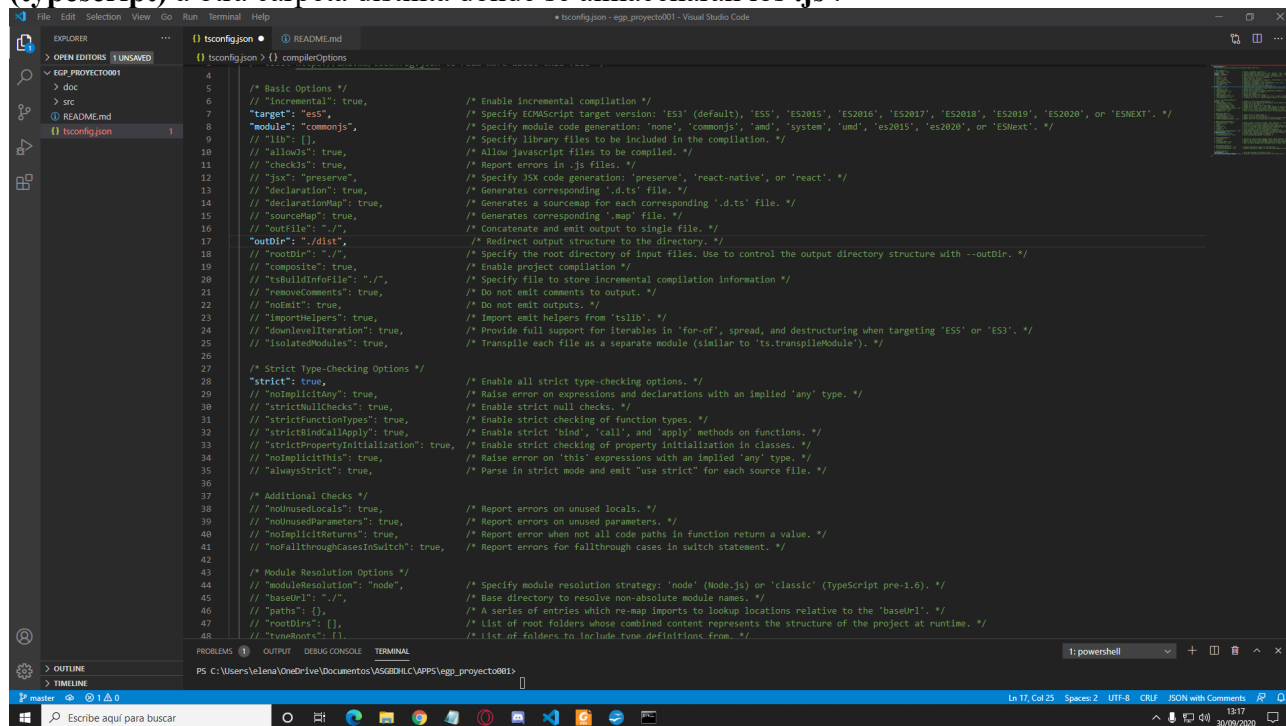
Comencemos a crear un nuevo proyecto en GitHub, para ellos creamos la carpeta del proyecto y lo abrimos en Visual Studio Code.



Primero abrimos un terminal y ponemos el comando **tsc --init**, con esto iniciamos el proyecto con typescript, al lanzar el comando en la terminal se nos creara el fichero **tsconfig.json**.



En este archivo vamos a modificar la línea del **OutDir** para controlar los archivos traspillados de **.ts (typescript)** a otra carpeta distinta dónde se almacenarán los **.js**.



```
4  /* Basic Options */
5  // "incremental" true enables incremental compilation
6  // "target": "es5",
7  // "lib": [],
8  // "allowJs": true,
9  // "checkJs": true,
10 // "jsx": "preserve",
11 // "declaration": true,
12 // "declarationMap": true,
13 // "sourceMap": true,
14 // "outFile": "dist",
15 // "rootDir": "src",
16 // "composite": true,
17 // "tsBuildInfoFile": "tsconfig.tsbuildinfo",
18 // "removeComments": true,
19 // "noEmit": true,
20 // "importHelpers": true,
21 // "downlevelIteration": true,
22 // "isolatedModules": true,
23
24 /* Strict Type-Checking Options */
25 "strict": true,
26 // "strictNullChecks": true,
27 // "strictFunctionTypes": true,
28 // "strictBindCallApply": true,
29 // "strictPropertyInitialization": true,
30 // "strictThis": true,
31 // "alwaysStrict": true,
32
33 /* Additional Checks */
34 // "noUnusedLocals": true,
35 // "noUnusedParameters": true,
36 // "noImplicitReturns": true,
37 // "noFallthroughCasesInSwitch": true,
38
39 /* Module Resolution Options */
40 // "moduleResolution": "node",
41 // "baseUrl": "src",
42 // "paths": {},
43 // "rootDirs": [],
44 // "typeRoots": []
```

Nuestro proyecto tendrá la el archivo tsconfig.json, la carpeta doc, la carpeta src, el archivo README.md, la carpeta dist y el archivo .gitignore.