

MANUAL PROYECTO TYPESCRIPT

Contenido

0. Instalación de las herramientas necesarias.....	2
1. Preparación del entorno de trabajo	5
2. Transpilar TS → JS.....	6
3. Subida a GitHub	8

0. Instalación de las herramientas necesarias

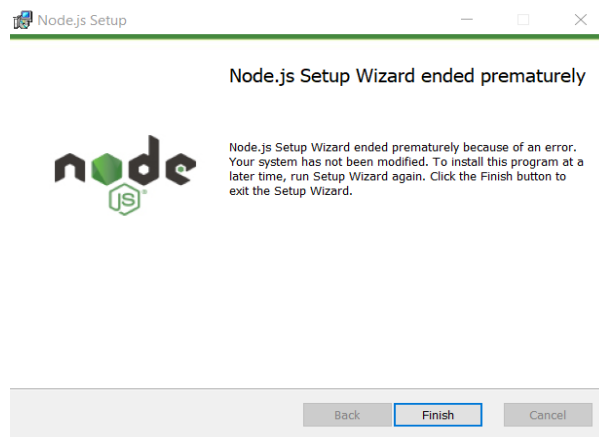
Para realizar un proyecto con **TypeScript** es aconsejable trabajar desde un entorno de desarrollo de código, dada las facilidades y ayudas que este brinda a los desarrolladores. En este caso vamos a usar **Visual Studio Code**. Sin embargo, este no es requisito fundamental ya que puede hacerse todos los procesos desde **PowerShell** de Windows, pero es menos intuitivo.

Typescript es un lenguaje de programación fuertemente tipado construido sobre JavaScript, por lo que necesitamos ciertas herramientas para que funcione en nuestra máquina.

El primer software que vamos a necesitar es **Node.js**. Node es un entorno de ejecución para **JavaScript**. Para descargarnos el programa, nos dirigimos a <https://nodejs.org/es/>

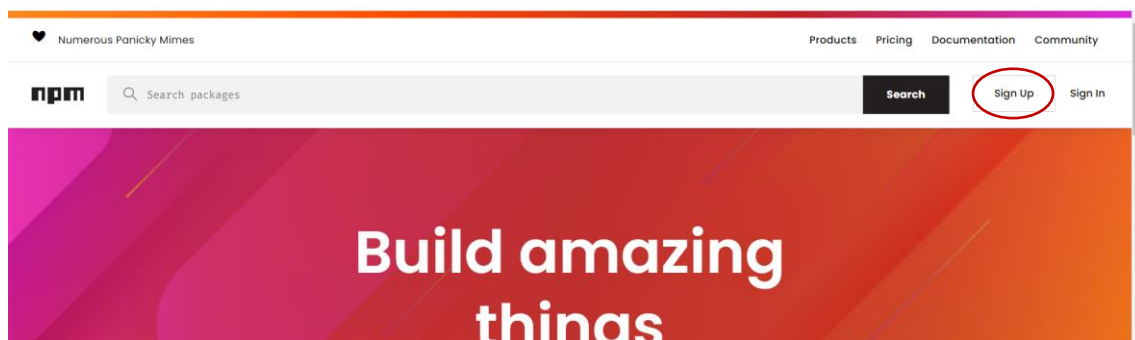


Después de elegir la opción que queramos se descargará automáticamente un ejecutable para su instalación. Seguimos los pasos que nos facilita el asistente de instalación hasta el fin de esta.

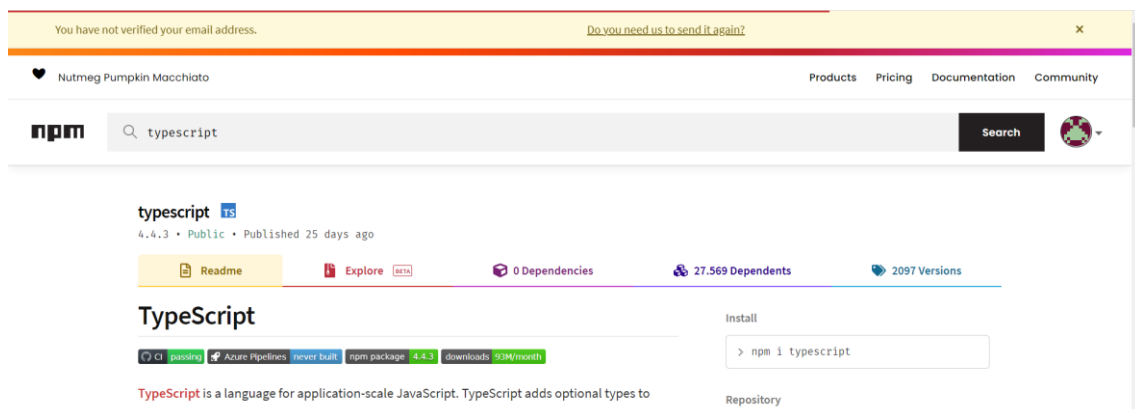


Con la descarga de **Node.js**, podemos avanzar al siguiente paso: Descarga de paquetes para node.

Para descargar **Typescript**, debemos de dirigirnos a la pagina de **node package manager**, <https://www.npmjs.com/>



Si es la primera vez que accedemos a este portal debemos de registrarnos, sin embargo, si ya tenemos cuenta, solo tenemos que identificarnos.



En el buscador ponemos **Typescript** y se nos abrirá una ventana, en la que se explica como instalar el lenguaje en el sistema. La opción -g nos dice que se instalará a nivel global.

Installing

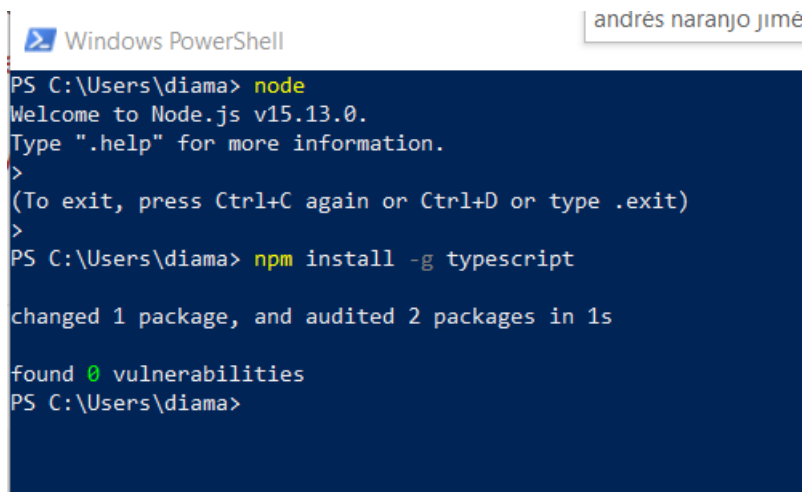
For the latest stable version:

```
npm install -g typescript
```

For our nightly builds:

```
npm install -g typescript@next
```

Estas sentencias deben ser ejecutadas en la consola de **PowerShell** de Windows.



```
Windows PowerShell
PS C:\Users\diama> node
Welcome to Node.js v15.13.0.
Type ".help" for more information.
>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
>
PS C:\Users\diama> npm install -g typescript

changed 1 package, and audited 2 packages in 1s

found 0 vulnerabilities
PS C:\Users\diama>
```

Como se puede comprobar, **Node** se descargó correctamente al igual que **TypeScript**.

1. Preparación del entorno de trabajo

Para iniciar un proyecto, debemos de crear un directorio donde irá alojado. Dentro de este directorio crearemos las carpetas de **doc** y **src**.

Este equipo > Windows (D:) > ASIRsegundo > SGDB > proyecto_01 >					
Nombre	Fecha de modificación	Tipo	Tamaño		
doc	21/09/2021 19:35	Carpeta de archivos			
src	21/09/2021 19:07	Carpeta de archivos			

Desde VSC abrimos nuestro directorio y nos dirigimos a la terminal. Desde la ruta donde está alojado nuestro proyecto, ejecutamos los siguientes comandos:

Npm install → **node_modules**

npm init -y → **package.json**

tsc --init → **tsconfig.json**

Estos comandos nos van a generar una serie de archivos con la configuración por defecto de nuestro proyecto.

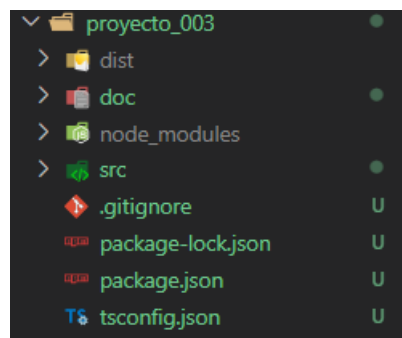
```

proyecto_003 > tsconfig.json > {} compilerOptions
11 // "disableReferencedProjectLoad": true,
12
13 /* Language and Environment */
14 "target": "es6",
15 // "lib": [],
16 // "jsx": "preserve",
17 // "experimentalDecorators": true,
18 // "emitDecoratorMetadata": true,
19 // "jsxFactory": "",
20 // "jsxFragmentFactory": "",
21 // "jsxImportSource": "",
22 // "reactNamespace": "",
23 // "noLib": true,
24 // "useDefineForClassFields": true,
25
26 /* Modules */
27 "module": "commonjs",
28 // "rootDir": "./",
29 // "moduleResolution": "node",
30 // "baseUrl": "./",
31 // "paths": {},
32 // "rootDirs": [],
33 // "typeRoots": [],
34 // "types": [],
35 // "allowUmdGlobalAccess": true,
36 // "resolveJsonModule": true,
37 // "noResolve": true,
38
39 /* JavaScript Support */
40 // "allowJs": true,
41 // "checkJs": true,
42 // "maxNodeModuleJsDepth": 1,
43
44 /* Emit */
45 // "declaration": true,
46 // "declarationMap": true,
47 // "emitDeclarationOnly": true,
48 // "sourceMap": true,
49 // "outFile": "./",
50 "outDir": "./dist",
51 // "removeComments": true,

```

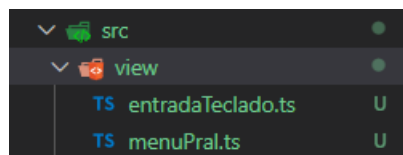
Dentro del archivo **tsconfig.json** debemos cambiar el **"target"** a **"es6"** y en el campo **"outDir"**, pondremos la carpeta en la que se generará el código transpilado, por defecto se elige **"./dist"**

En este momento, tendremos nuestro proyecto con estas carpetas.



Como no queremos que **./dist** ni **./node_modules** se suban a GitHub cuando el proyecto este finalizado, crearemos un **.gitignore** en el que escribiremos el nombre de ambas carpetas.

Dentro de la carpeta **src** se suele crear una carpeta **view** que contiene contenido auxiliar que necesitaremos a lo largo del proyecto, separándose del grueso del proyecto.



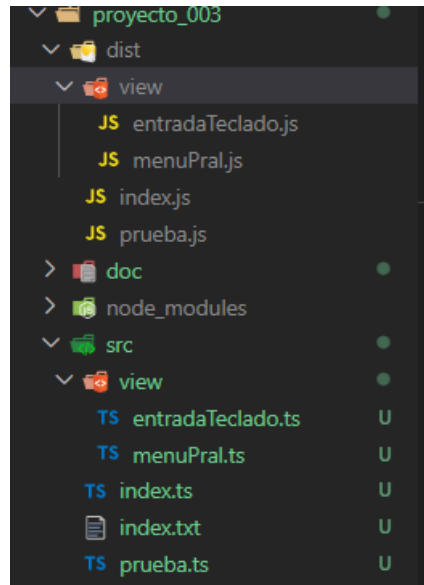
2. Transpilar TS → JS

Una vez que tengamos nuestro código en Typescript listo, nos dirigiremos a la terminal e introduciremos el comando **tsc -w**

```
D:\ASIRsegundo\SGBD\proyecto_003\src>tsc -w
[21:26:17] Starting compilation in watch mode...

[21:26:18] Found 0 errors. Watching for file changes.
```

Una vez realizado este proceso, comprobaremos que la carpeta **dist** tiene los mismos archivos que la carpeta **src**, con la diferencia de que estarán en .js



Para evitar que algún archivo de la carpeta **node_modules** pueda interferir, en el archivo **tsconfig.json** podemos introducir lo siguiente al final de este.

```
94 // "allowUnreachableCode": true,           /* Disable error reporting for unreachable code. */
95
96 /* Completeness */
97 // "skipDefaultLibCheck": true,             /* Skip type checking .d.ts files that are included with TypeScript. */
98 "skipLibCheck": true                       /* Skip type checking all .d.ts files. */
99 ],
100 "exclude": [
101   |   | "node_modules"
102 ]
103 }
104
```

3. Subida a GitHub

Creamos un repositorio para alojar nuestro nuevo proyecto.

Owner ^{*} Repository name ^{*}

NaranjoJimenezAndres2 / proyecto_typescript02 ✓

Great repository names are short and memorable. Need inspiration? How about [fantastic-octo-telegram](#)?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Introducimos los siguientes comandos, y nuestro repositorio quedará alojado en GitHub posibilitando al resto de la comunidad a acceder a ellos.

...or create a new repository on the command line

```
echo "# proyecto_typescript02" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/NaranjoJimenezAndres2/proyecto_typescript02.git
git push -u origin main
```

```
D:\ASIRsegundo\SGBD\proyecto_typescript02>git init
Reinitialized existing Git repository in D:\ASIRsegundo\SGBD\proyecto_typescript02\.git/

D:\ASIRsegundo\SGBD\proyecto_typescript02>git commit -m "first commit"
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        doc/
        package-lock.json
        package.json
        src/
        tsconfig.json

nothing added to commit but untracked files present (use "git add" to track)

D:\ASIRsegundo\SGBD\proyecto_typescript02>
D:\ASIRsegundo\SGBD\proyecto_typescript02>git add .
fatal: ..: '..' is outside repository at 'D:\ASIRsegundo\SGBD\proyecto_typescript02'

D:\ASIRsegundo\SGBD\proyecto_typescript02>git add .
warning: LF will be replaced by CRLF in package-lock.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in package.json.
The file will have its original line endings in your working directory

D:\ASIRsegundo\SGBD\proyecto_typescript02>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   doc/manual proyecto typescript.docx
        new file:   package-lock.json
        new file:   package.json
        new file:   src/index.ts
```



```
D:\ASIRsegundo\SGBD\proyecto_typescript02>git commit -m "first commit"
[master (root-commit) 5686ea1] first commit
9 files changed, 343 insertions(+)
create mode 100644 .gitignore
create mode 100644 doc/manual proyecto_typescript.docx
create mode 100644 package-lock.json
create mode 100644 package.json
create mode 100644 src/index.ts
create mode 100644 src/prueba.ts
create mode 100644 src/view/entradaTeclado.ts
create mode 100644 src/view/menuPral.ts
create mode 100644 tsconfig.json

D:\ASIRsegundo\SGBD\proyecto_typescript02>git branch -M main

D:\ASIRsegundo\SGBD\proyecto_typescript02>git remote add origin https://github.com/NaranjoJimenezAndres2/proyecto_typescript02.git

D:\ASIRsegundo\SGBD\proyecto_typescript02>git push -u origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 12 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (14/14), 976.65 KiB | 834.00 KiB/s, done.
Total 14 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/NaranjoJimenezAndres2/proyecto_typescript02.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.

D:\ASIRsegundo\SGBD\proyecto_typescript02>
```