

# PRIMER PROYECTO DE TYPESCRIPT



Asignatura: Gestión de Base de Datos  
Alumno: Miguel García Holgado  
Fecha entrega: 08/10/2021

# INDICE

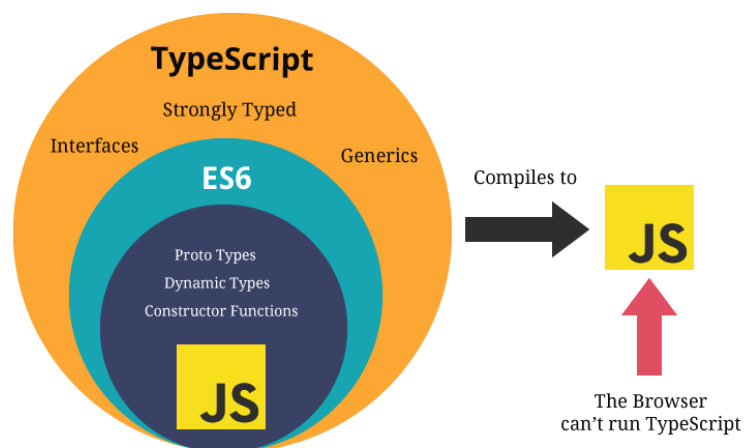
1. Introducción a Typescript.....	1
1.1 ¿Qué es Typescript?.....	1
1.2 ¿Como instalamos Typescript?.....	2
2. Introducción a nodeJS.....	4
2.1 ¿Qué es nodeJS?.....	4
2.2 ¿Como se instala nodeJS?.....	5
3. Preparando el entorno de desarrollo.....	11
3.1 Creación estructura de directorios.....	11
4. Instalar un proyecto existente.....	15

# 1. Introducción a Typescript

## 1.1 ¿Qué es Typescript?

JavaScript es uno de los lenguajes más populares, en parte porque ha evolucionado y mejorado a pasos agigantados en los últimos años.

Sin embargo, Javascript en algún punto fue un lenguaje que presentaba muchos problemas para bases de código grandes, aplicaciones de gran escala y proyectos con muchos años de desarrollo.



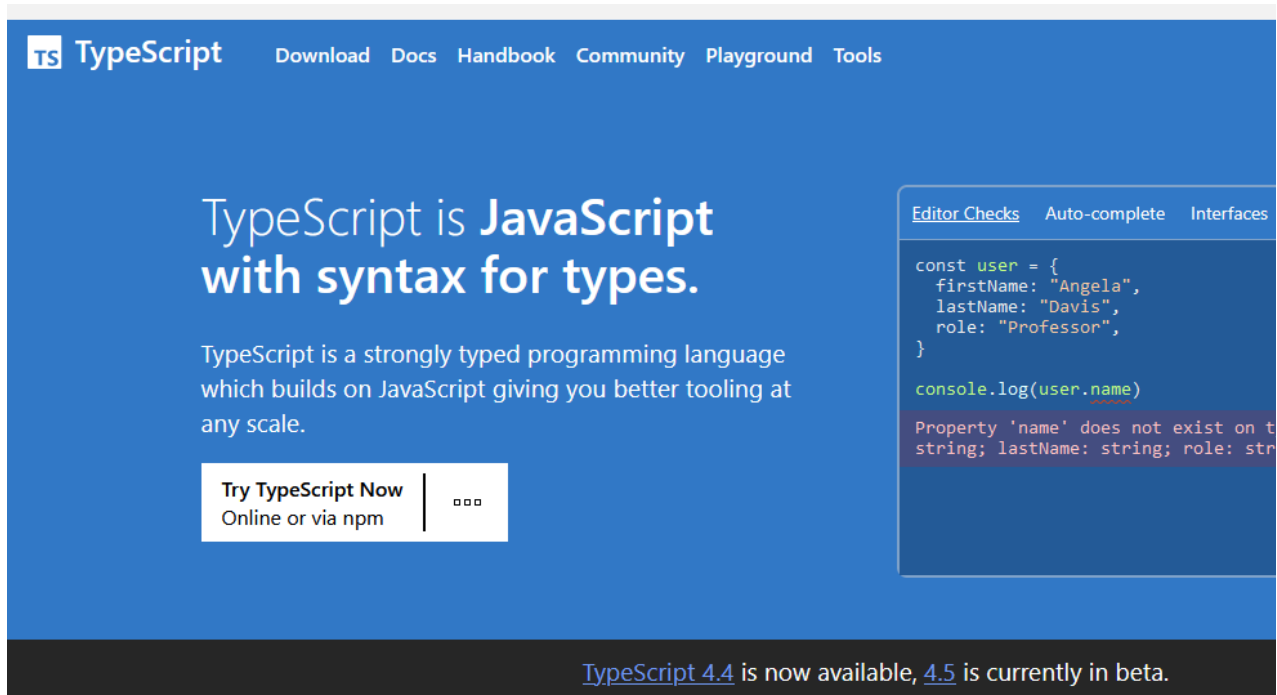
2012 fue el año en que Typescript apareció (luego de 2 años de desarrollo), una solución de Microsoft para el desarrollo de aplicaciones con Javascript a gran escala, Typescript es la solución a muchos de los problemas de JavaScript, está pensado para el desarrollo de aplicaciones robustas.

## 1.2 ¿Como instalamos Typescript?

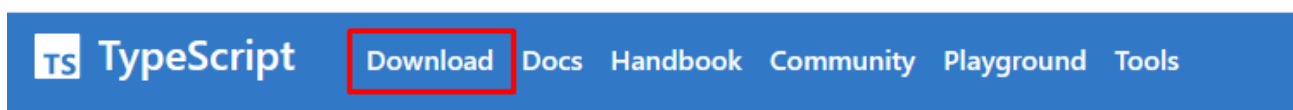
En primer lugar vamos a ir a este enlace →

<https://www.typescriptlang.org/> para poder ir a la página oficial del lenguaje Typescript.

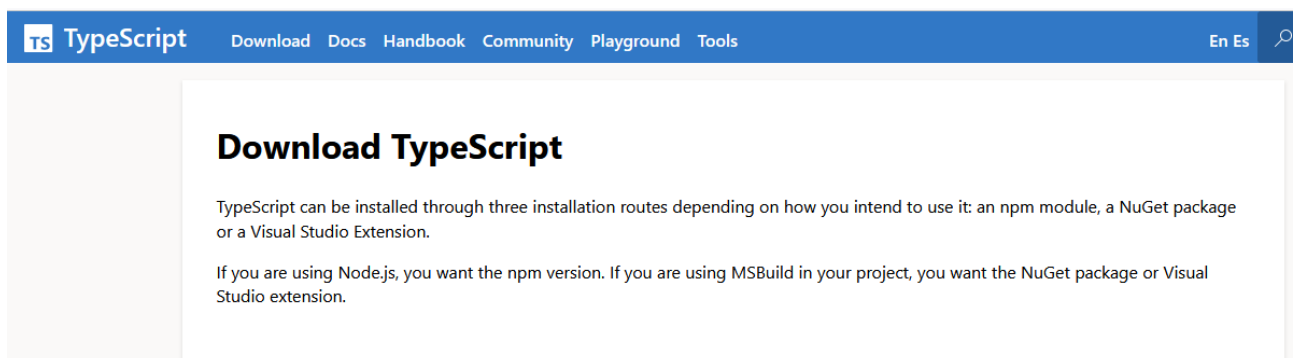
Una vez allí nos saldrá esta pestaña:



Ahora le vamos a dar en el apartado de Download



Nos llevará a la siguiente pestaña:



Ahora nos va a dejar elegir la forma de descargarlo:

#### via npm

TypeScript is available as a [package on the npm registry](#) available as "typescript".

You will need a copy of [Node.js](#) as an environment to run the package. Then you use a dependency manager like [npm](#), [yarn](#) or [pnpm](#) to download TypeScript into your project.

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

[npm](#) [yarn](#) [pnpm](#)

All of these dependency managers support lockfiles, ensuring that everyone on your team is using the same version of the language. You can then run the TypeScript compiler using one of the following commands:

```
npx tsc
```

[npm](#) [yarn](#) [pnpm](#)

#### with Visual Studio

For most project types, you can get TypeScript as a package in NuGet for your MSBuild projects, for example an ASP.NET Core app.

When using NuGet, you can [install TypeScript through Visual Studio](#) using:

- The Manage NuGet Packages window (which you can get to by right-clicking on a project node)
- The NuGet Package Manager Console (found in Tools > NuGet Package Manager > Package Manager Console) and then running:  
`Install-Package Microsoft.TypeScript.MSBuild`

For project types which don't support NuGet, you can use the [TypeScript Visual Studio extension](#). You can [install the extension](#) using Extensions > Manage Extensions in Visual Studio.

- via npm: instalaremos typescript con node package manager el cual veremos como instalarlo mas tarde al instalar nodeJS.
- with Visual Studio: instalando desde Visual Studio una extensión.

Nosotros lo instalaremos por comando npm mas adelante cuando veamos nodeJS.

## 2. Introducción a nodeJS

### 2.1 ¿Qué es nodeJS?

---

Node.js es un entorno de tiempo de ejecución de JavaScript (de ahí su terminación en .js haciendo alusión al lenguaje JavaScript).

Node.js utiliza un modelo de entrada y salida sin bloqueo controlado por eventos que lo hace ligero y eficiente (con entrada nos referimos a solicitudes y con salida a respuestas). Puede referirse a cualquier operación, desde leer o escribir archivos de cualquier tipo hasta hacer una solicitud HTTP.



## 2.2 ¿Como se instala nodeJS?

Para instalar nodeJS vamos a ir a la página web oficial para poder descargar el instalador → <https://nodejs.org/es/>

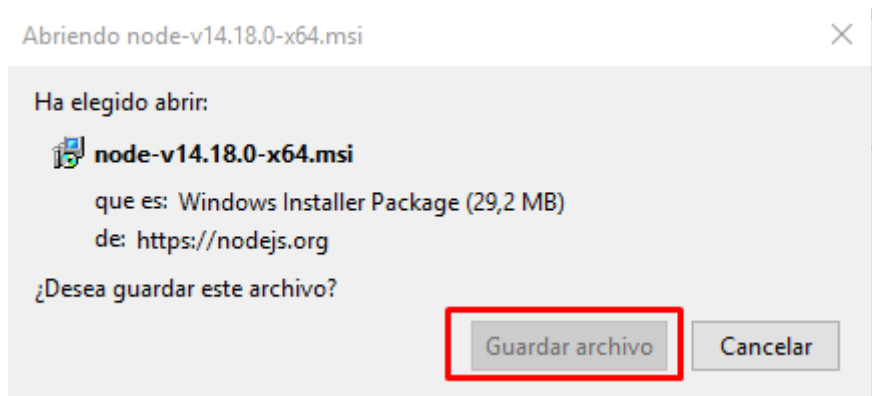
Nos saldrá una página similar a esta:



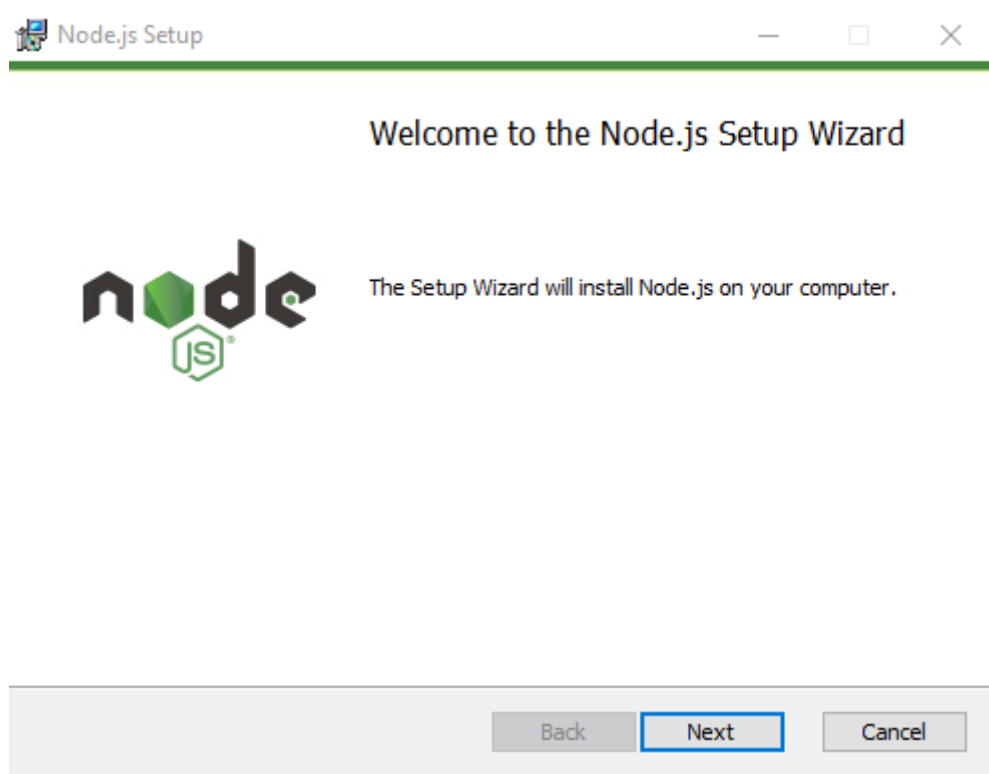
Para hacer la descarga de node.js vamos a darle donde pone descarga para Windows, le daremos click en el botón verde.

Vamos a descargarnos la versión anterior ya que es la más recomendada.

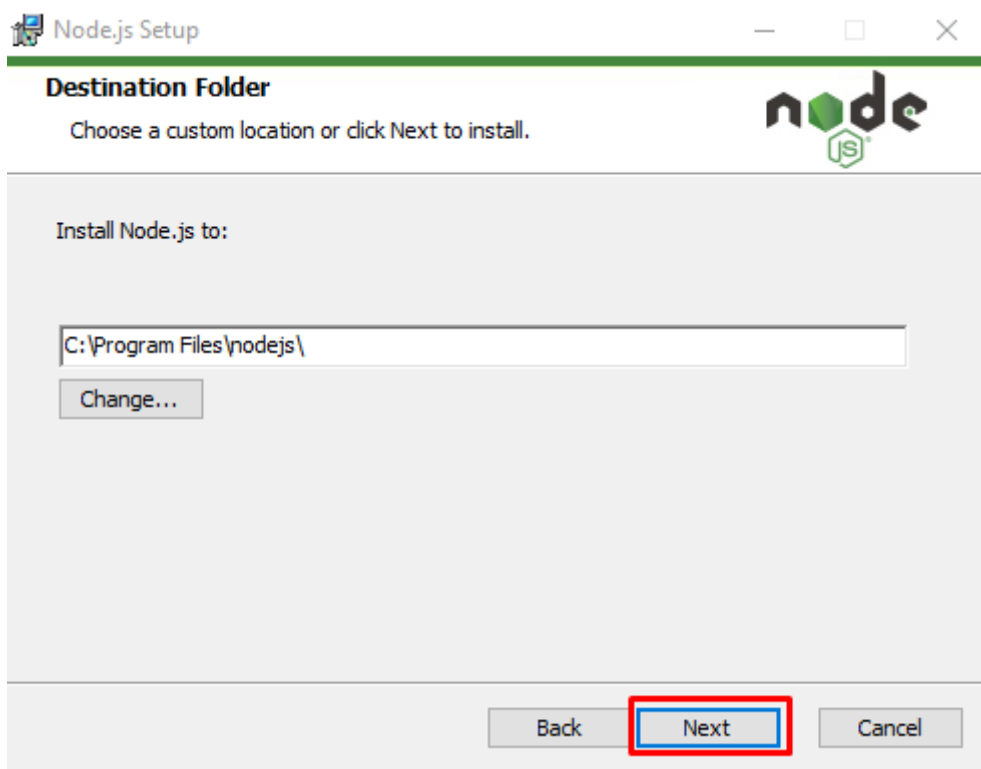
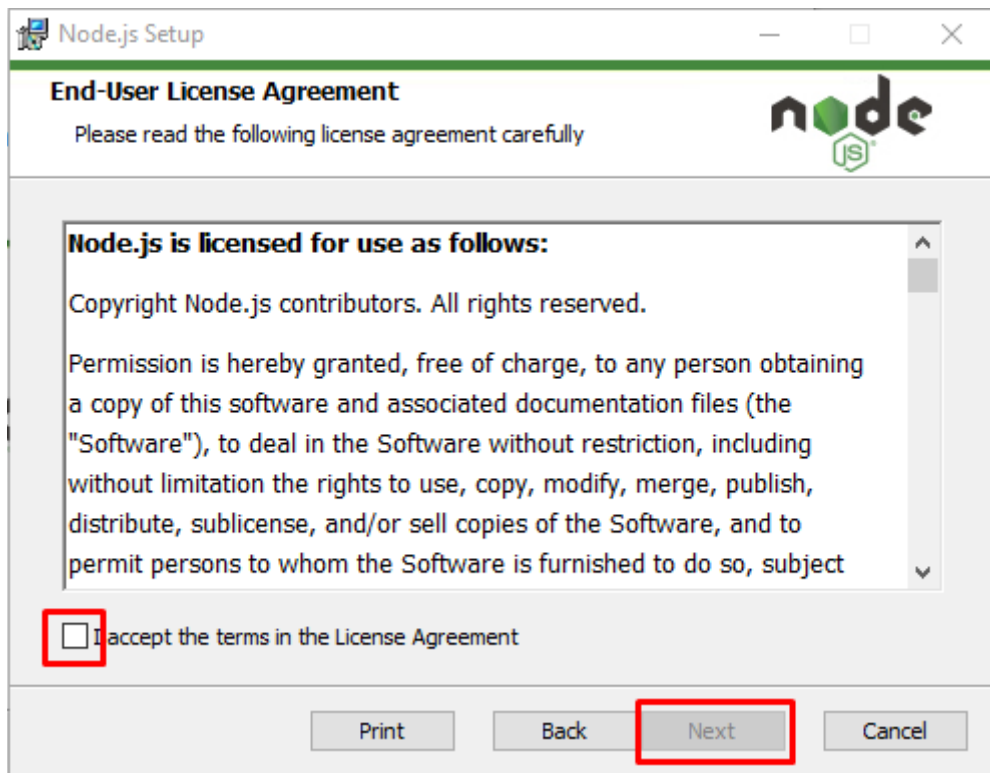


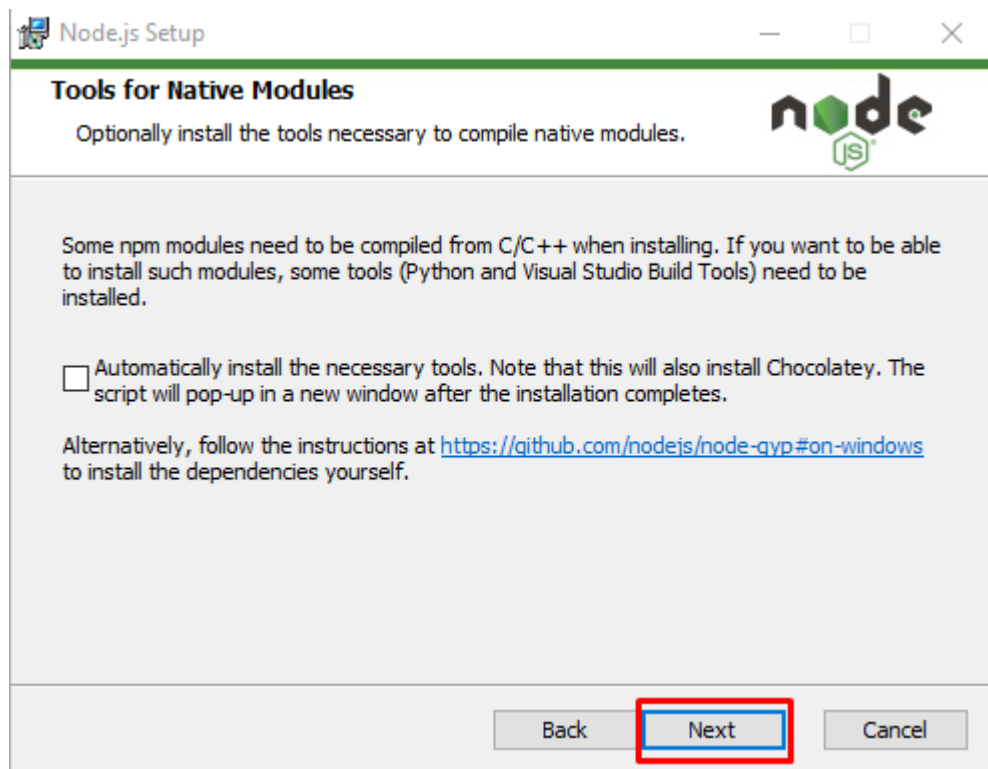
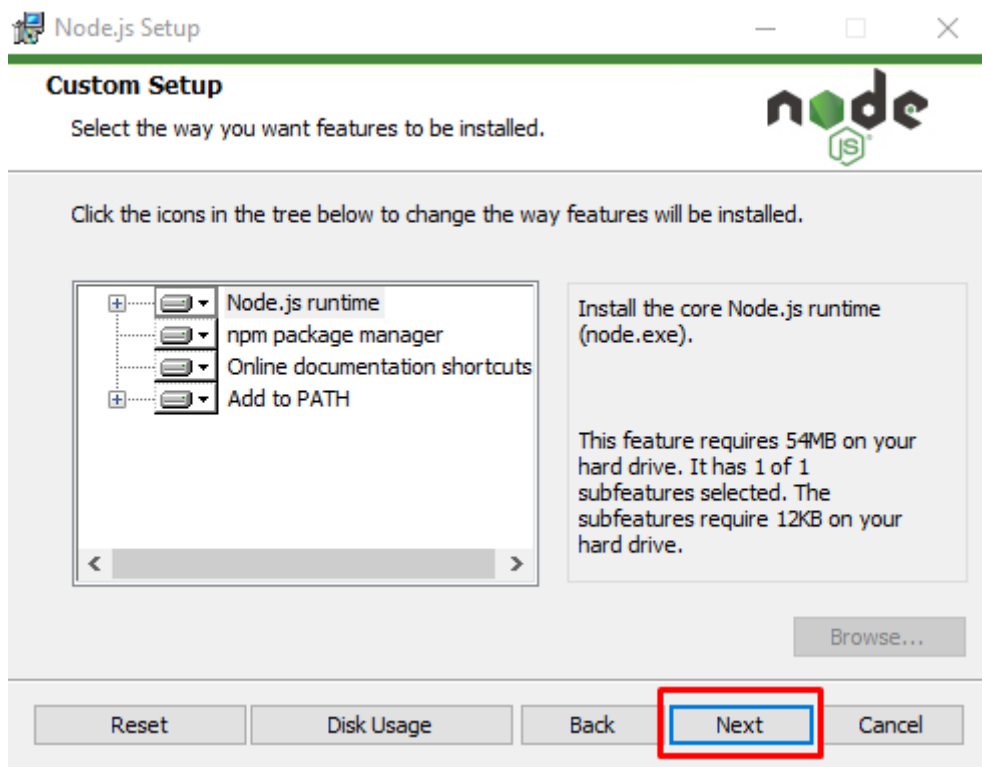


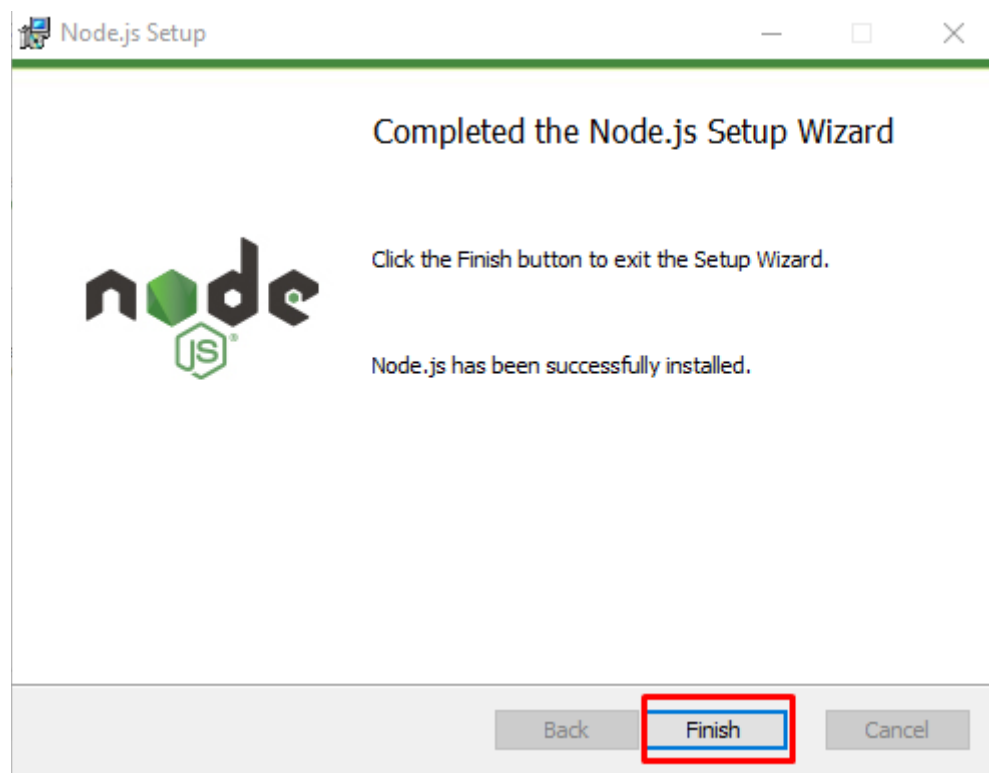
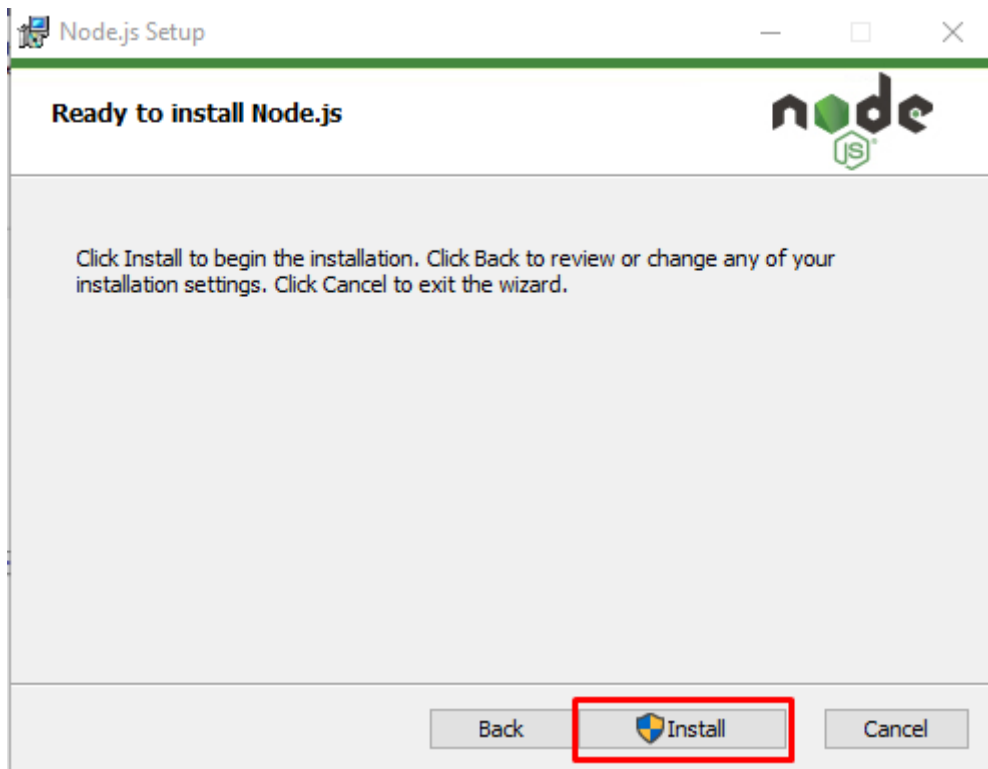
Guardamos el .msi y guardamos el archivos, una vez lo tengamos lo instalamos, para ello le damos doble click y lo ejecutamos.











Una vez hay terminado la instalación de nodeJS vamos a ver si esta instalado en el sistema, para ello vamos a la powershell y escribimos node --version

```
PS C:\Users\usuario> node --version  
v14.18.0
```

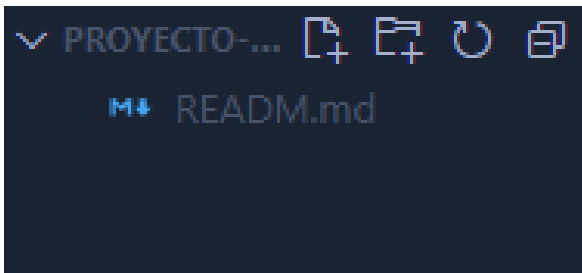
Como podemos ver tenemos instalada la versión que anteriormente instalamos.

## 3. Preparando el entorno de desarrollo

### 3.1 Creación estructura de directorios

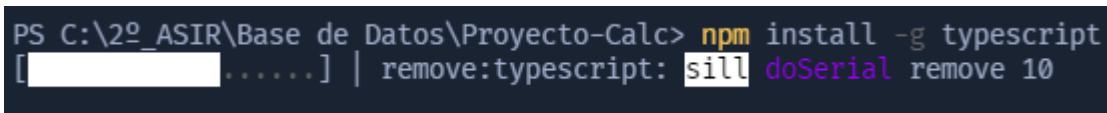
En este apartado vamos a empezar a crear nuestro proyecto, vamos a crear las carpetas que contendrán los archivos para hacer la calculadora y vamos a explicar brevemente para que utilizamos cada archivo.

Ahora mismo nuestro proyecto esta vacío, vamos a empezar por crear el archivo README.md el cual servirá de información a la persona que descargue nuestro proyecto, le explicará de que va.



Después de haber añadido nuestro archivo README.md vamos a empezar con nodeJS, en primer lugar vamos a instalar typescript para ello usaremos este comando, así que abrimos una terminal de visual y lo ejecutamos:

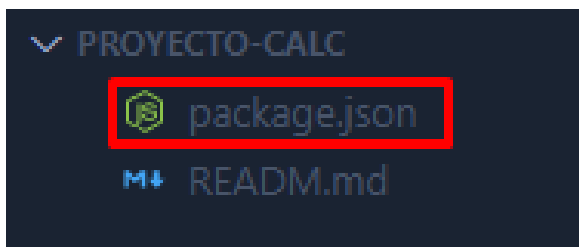
`npm install -g typescript`



Una vez instalado typescript vamos a iniciar npm y cuando lo hagamos se nos creará el archivo package.json, usaremos el siguiente:

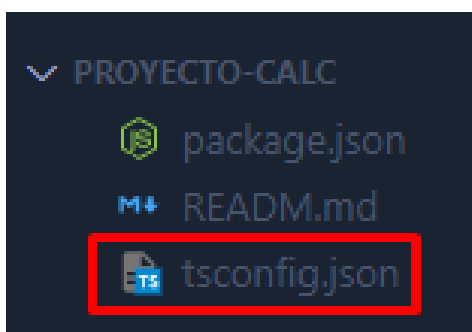
npm init -y

```
PS C:\2º_ASIR\Base de Datos\Proyecto-Calc> npm init -y
```



Ahora vamos a iniciar typescript para ellos usaremos el comando: tsc --init, cuando ejecutemos este comando se nos creará el archivo tsconfig.json.

```
PS C:\2º_ASIR\Base de Datos\Proyecto-Calc> tsc --init  
message TS6071: Successfully created a tsconfig.json file.
```



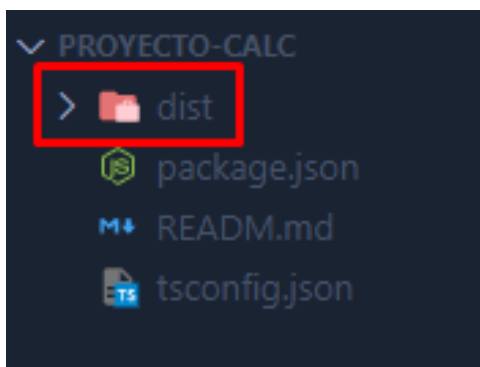
Ahora vamos a dirigirnos al archivo `tsconfig.json` vamos a cambiar la ruta de la línea `outDir` y vamos a cambiar la versión de la 5 a la 6.

Para ello nos dirigimos a la líneas 14 y 50, la línea 14 es para la versión y la 50 para el `outDir`.

```
12
13      /* Language and Environment */
14      "target": "es6",
15      // "lib": [],
16      // "jsx": "preserve",
```

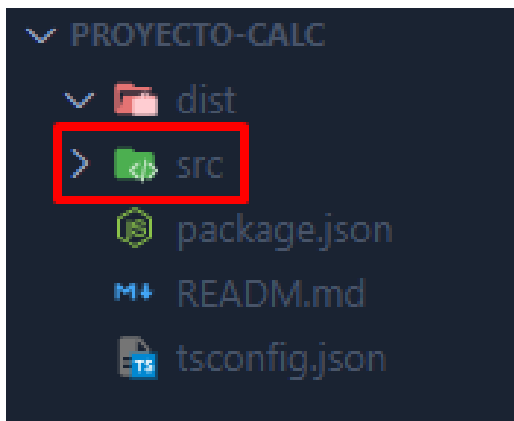
```
48      // "sourceMap": true,
49      // "outFile": "./",
50      "outDir": "./dist",
51      // "removeComments": true,
```

Una vez creada la ruta de salida que es donde se van a mandar los archivos `.js` cuando ejecutemos la compilación de typescript, vamos a tener que crearla.



A screenshot of a file explorer window showing the project structure. The project is named 'PROYECTO-CALC'. Under this project, there is a folder named 'dist' which is highlighted with a red box. Below the 'dist' folder, there are several files: 'package.json', 'README.md', and 'tsconfig.json'.

Seguido de esto crearemos la carpeta src para meter ahí los archivos typescript:

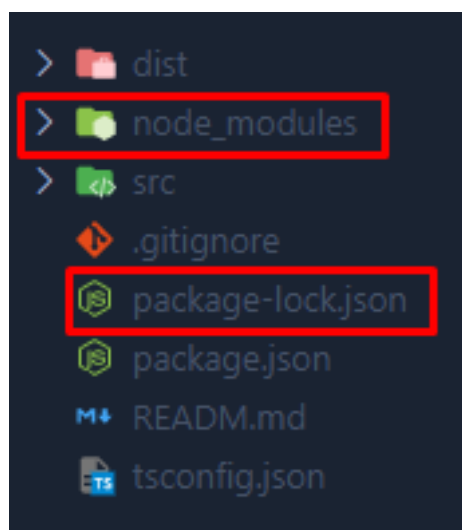


Una vez tengamos estos archivos y carpetas vamos a crear un archivo .gitignore donde irán las cosas que o queremos que se suban a git.

Después de esto vamos a instalar typescript como dependencia, para ello usaremos este comando:

**npm install typescript --save-dev**

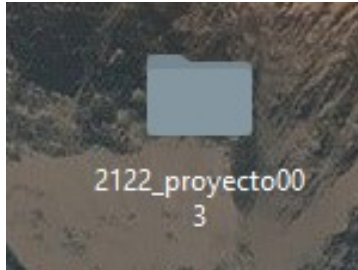
Se nos crean la siguiente carpeta que es donde se almacenarán los módulos que instalemos (esa carpeta la añadiremos al gitignore), también se nos ha creado el package-lock.json en el que está la información de las dependencias.



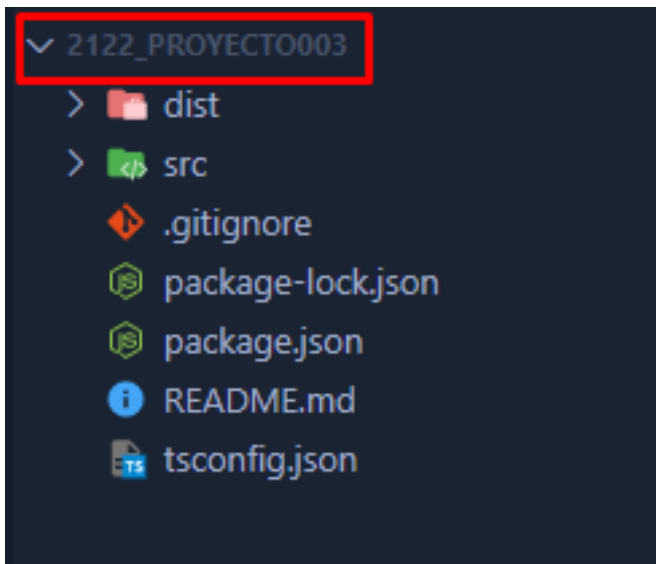


## 4. Instalar un proyecto existente

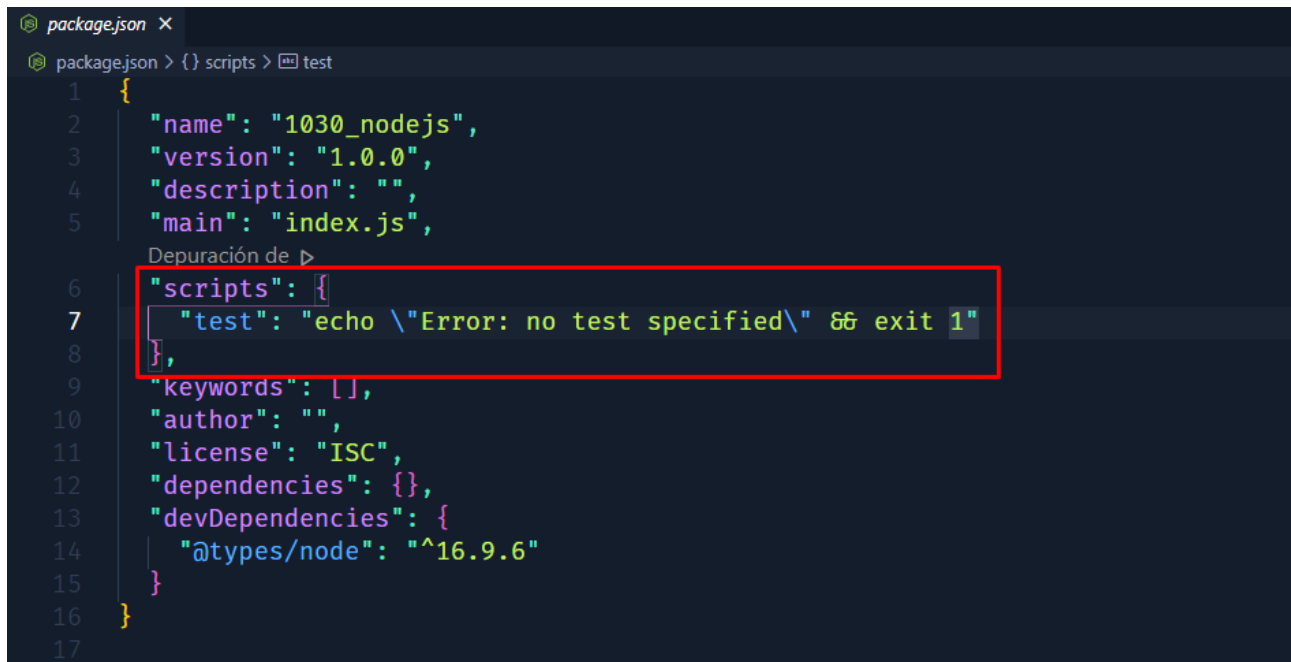
Para ello vamos a usar de prueba el Proyecto003 que nos descargaremos de la moodle.



Vamos a importar el proyecto a Vscode



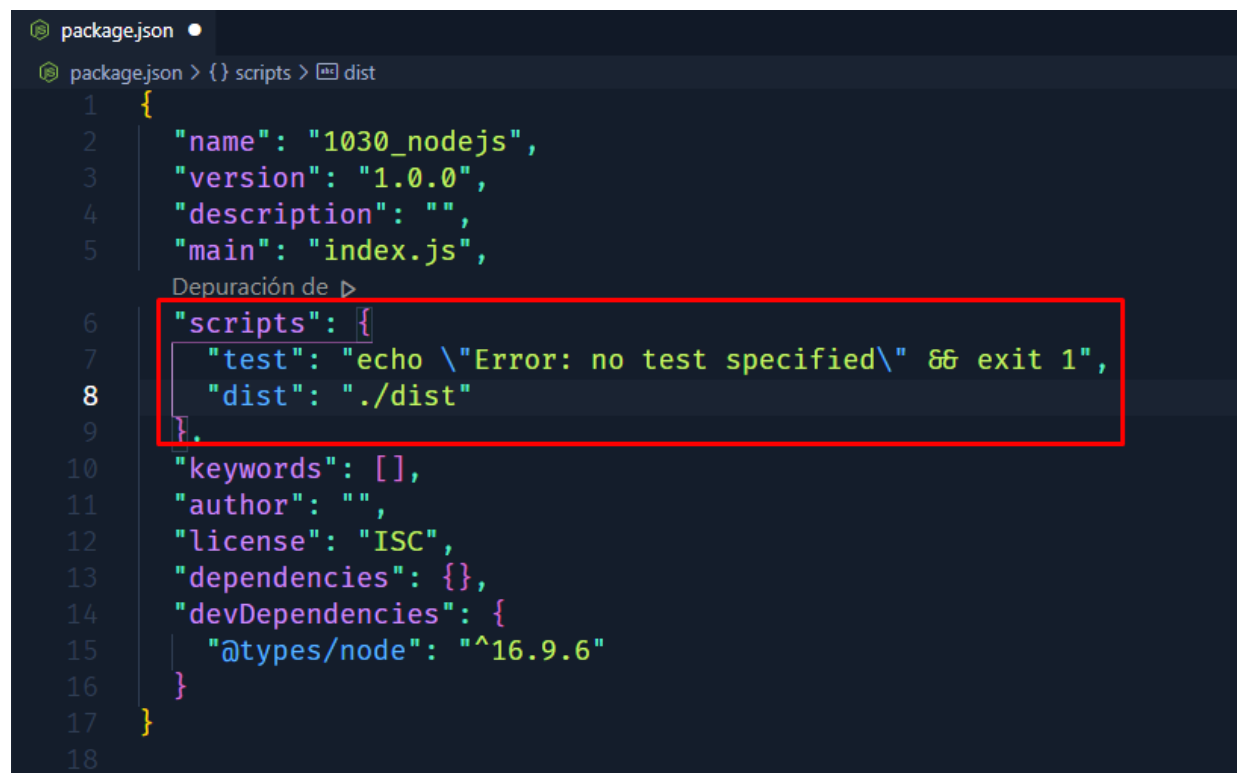
Primero iremos al archivo package.json y verificaremos que hay un script para iniciar el proyecto.



```
package.json x
package.json > {} scripts > test
1 {
2   "name": "1030_nodejs",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "keywords": [],
10  "author": "",
11  "license": "ISC",
12  "dependencies": {},
13  "devDependencies": {
14    "@types/node": "^16.9.6"
15  }
16 }
17
```

Como vemos no esta, asi que añadimos la siguiente linea:

“dist”: “./dist”



```
package.json •
package.json > {} scripts > dist
1 {
2   "name": "1030_nodejs",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1",
8     "dist": "./dist"
9   },
10  "keywords": [],
11  "author": "",
12  "license": "ISC",
13  "dependencies": {},
14  "devDependencies": {
15    "@types/node": "^16.9.6"
16  }
17 }
18
```

Siguiente paso usaremos este comando:

```
npm i --save-dev @types/node
```

Ya que este paquete contiene las definiciones de Node.js

Comprobamos con node dist si funciona nuestra ruta de script y nuestra calculadora

```
PS C:\Users\Mike\Desktop\2122_proyecto003> node dist

1.- Sumar
2.- Multiplicar
3.- Restar
0.- Salir
opción: : 1
Estoy en opción 1
Dame un número: 2
Dame otro número: 2
La suma es 4
```