

# Primer Proyecto TypeScript

Antonio ÁvilaDelgado

2º ASIR A

## **ÍNDICE**

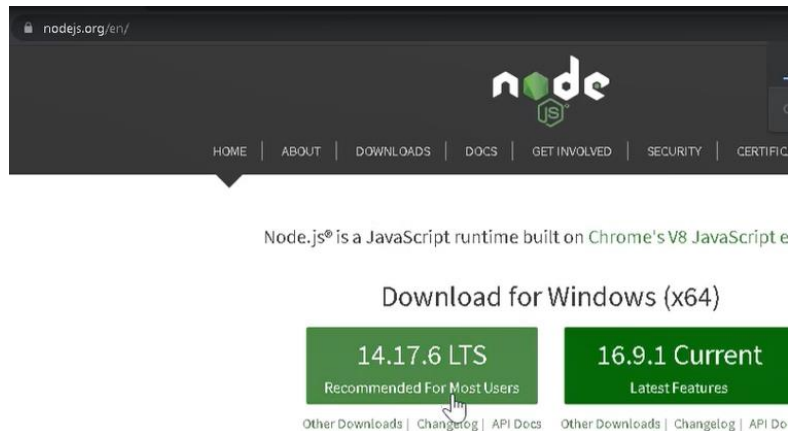
- Introducción.....pág 3.
- Pasos.....pág 3
- Instalar un proyecto ya existente.....pág 21

## INTRODUCCIÓN

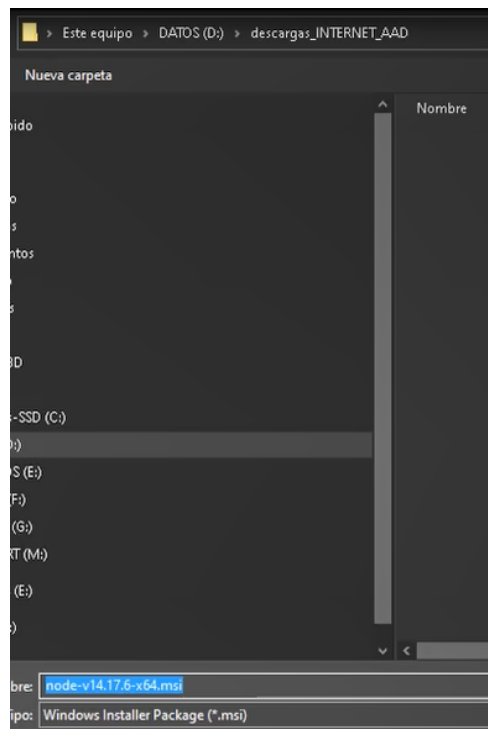
En este documento vamos a explicar, paso a paso, el montaje de un proyecto typescript. En concreto desde la instalación de node.js y el editor a la subida al repositorio. También la instalación de un proyecto que ya exista. Todo se ilustrará con la realización de un proyecto con una estructura similar a la ofrecida en el Proyecto003. Recibirá datos y ofrecerá resultados como: ver si un número es primo, comparar números, tareas con fechas o similares ...

## PASOS

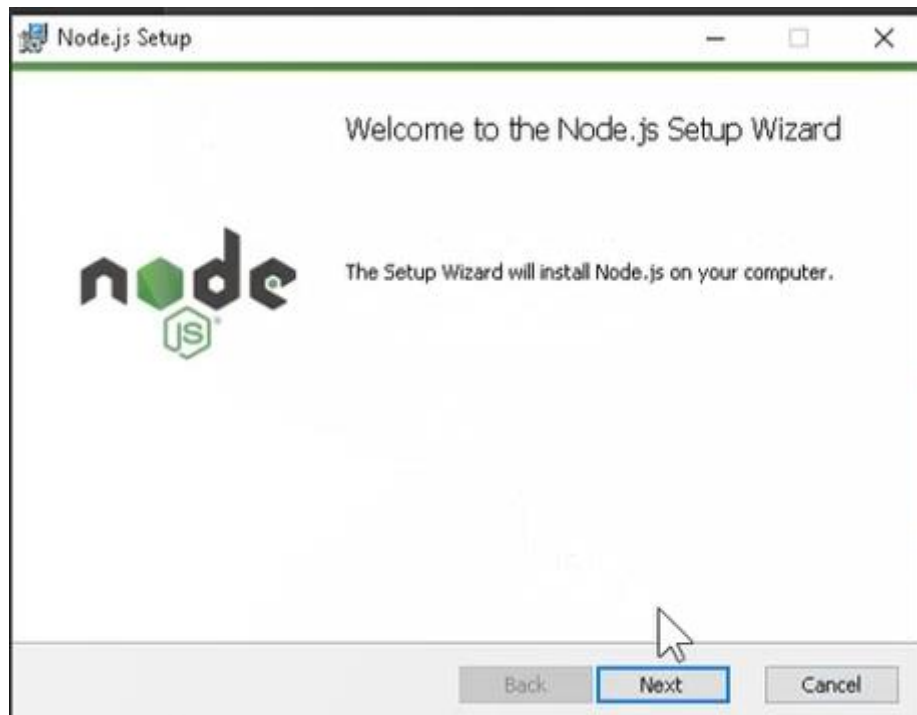
En primer lugar instalamos **node.js** desde su página oficial



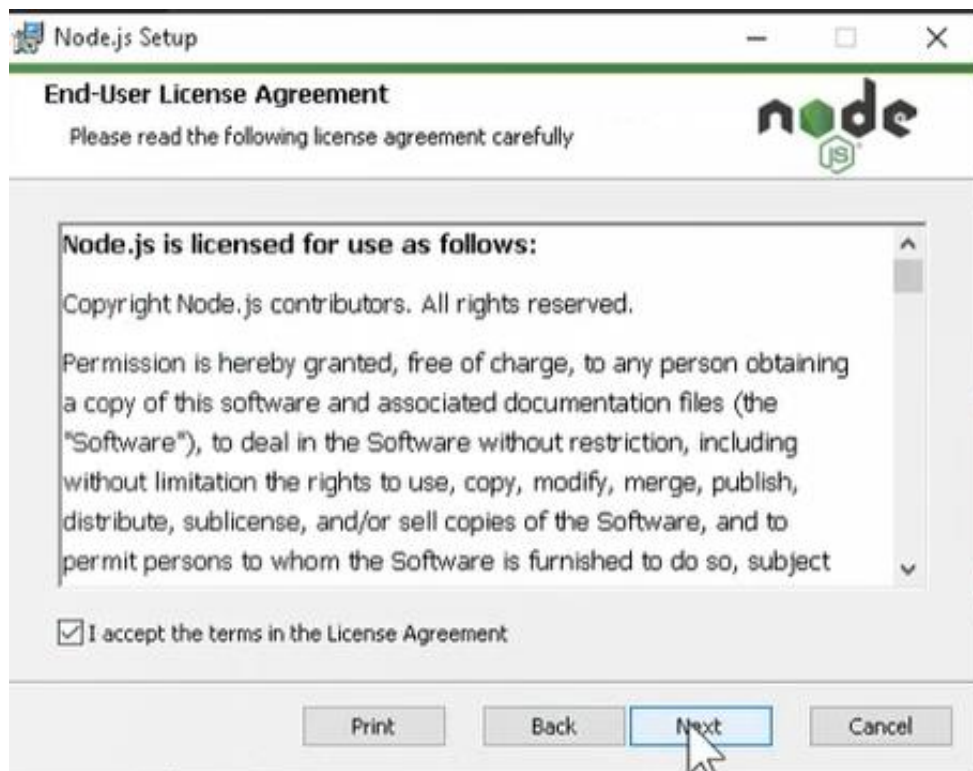
Al pulsar sobre la versión recomendada para la mayoría de los usuarios (14.17.6 LTS), sale:



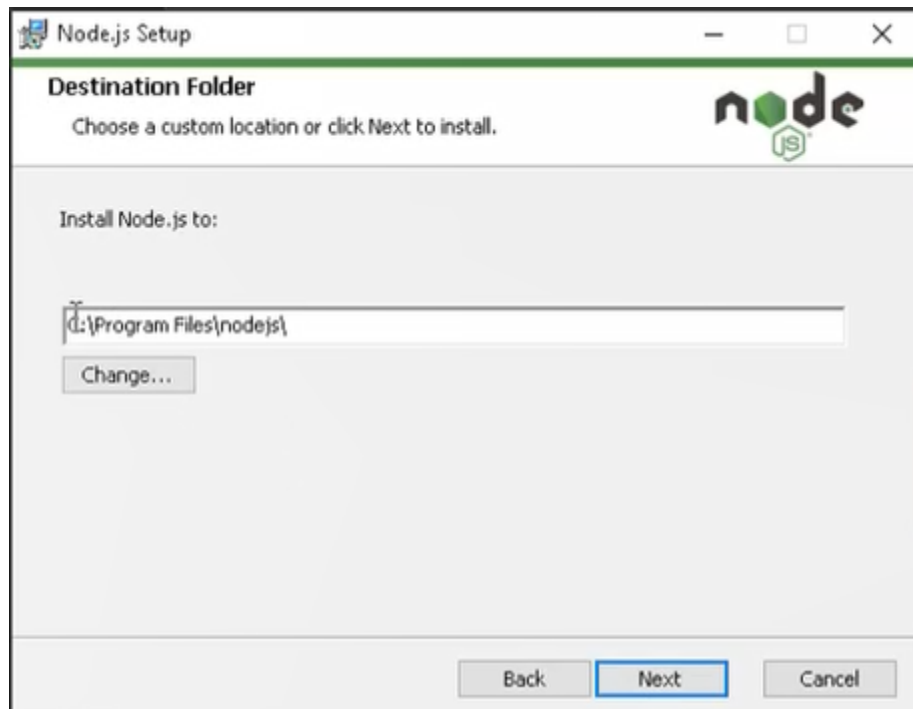
Ponemos el instalador \*.msi en la carpeta adecuada de nuestro PC para lanzar la instalación.



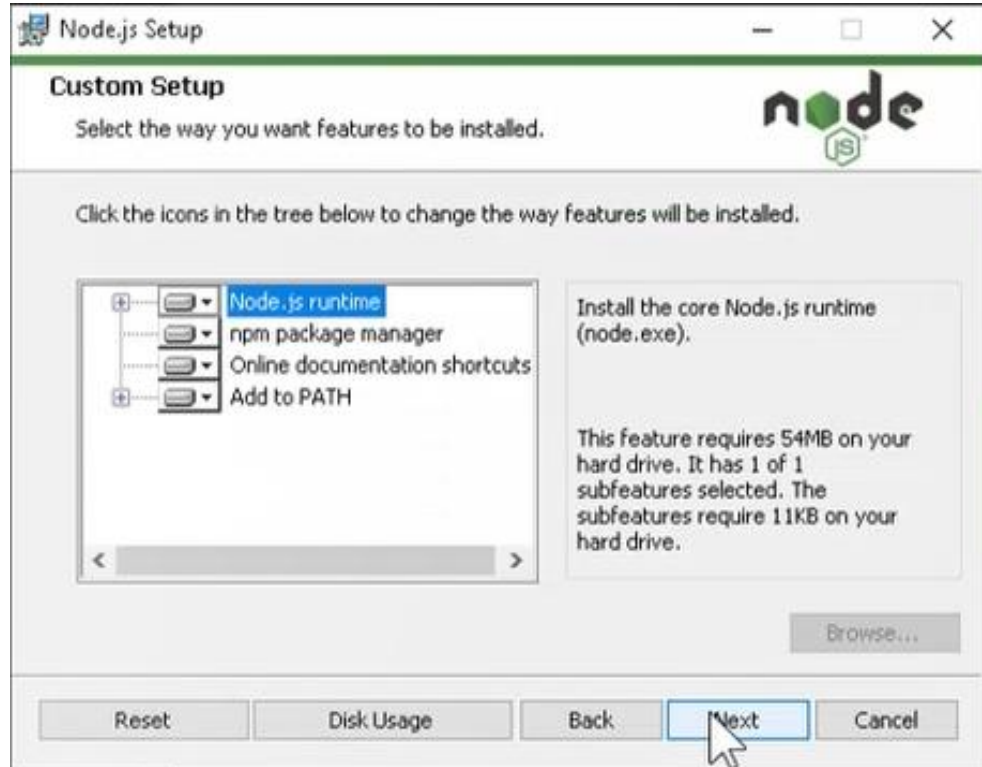
Pulsamos en [Next] y nos sale lo siguiente:



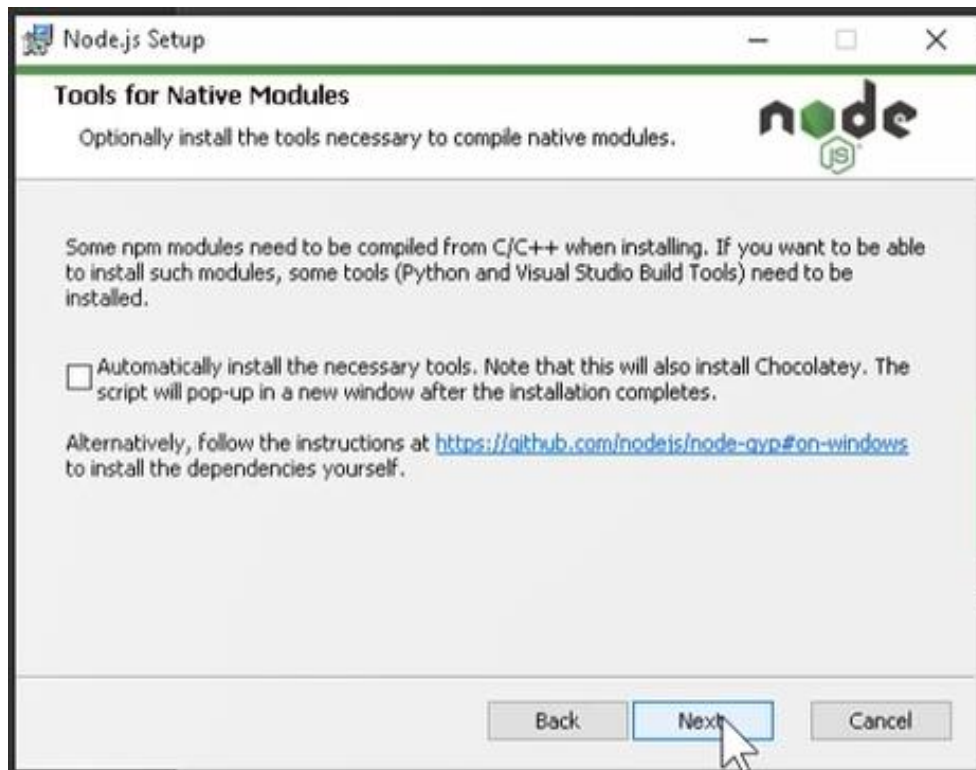
Aceptamos la licencia después de leerla. Pulsamos en [Next] y nos sale lo siguiente:



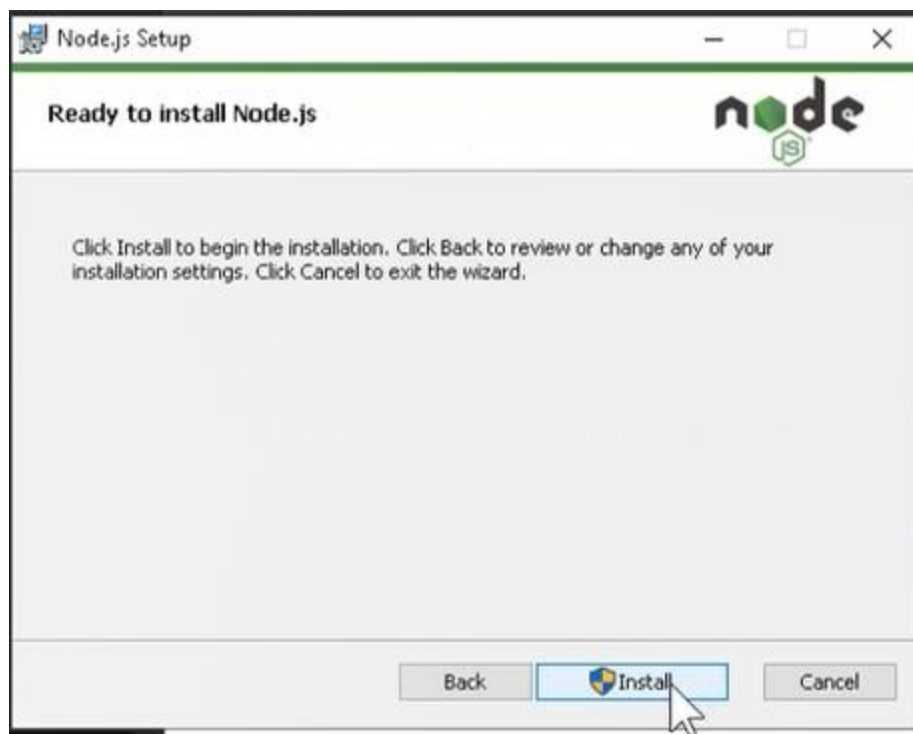
Elegimos la carpeta de destino por de la instalación. Pulsamos [Next] y nos sale lo siguiente:



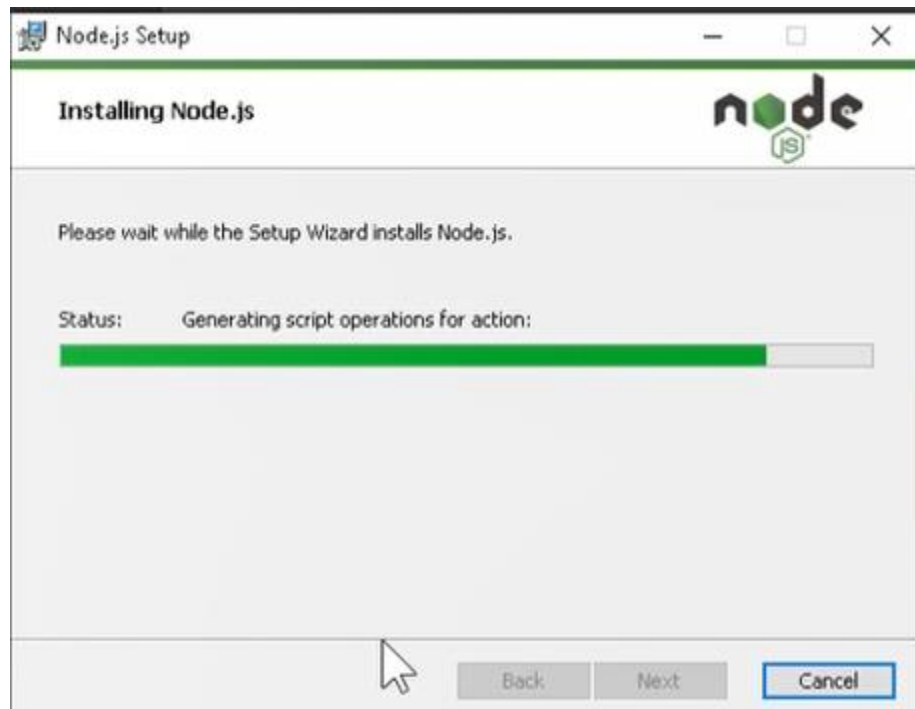
Aceptamos los componentes a instalar. Pulsamos [Next] y nos sale lo siguiente:



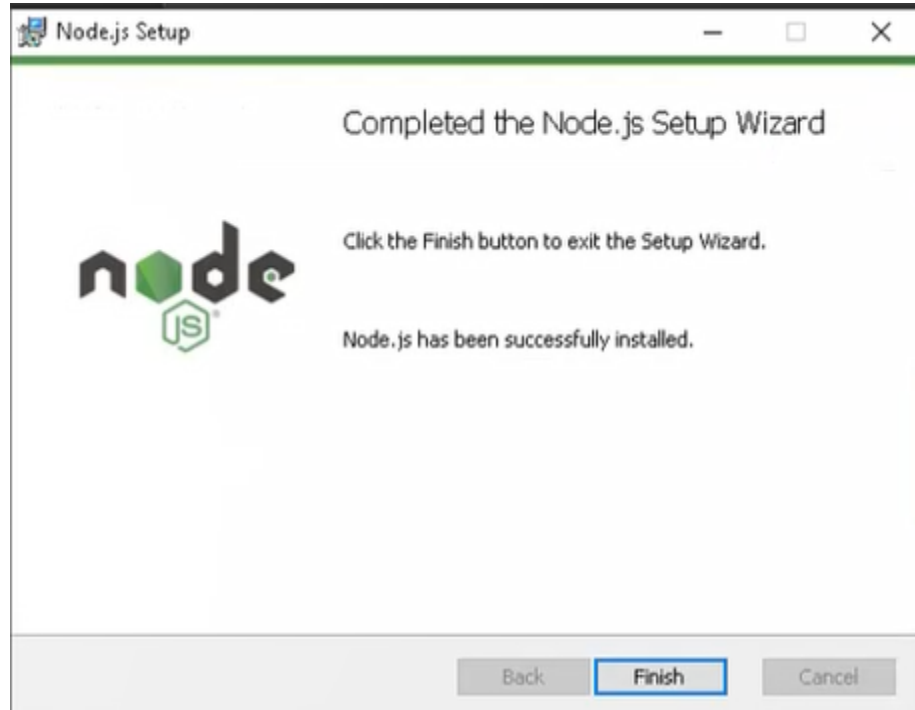
Dejamos las opciones como están. Pulsamos [Next] y nos sale lo siguiente:



Pulsamos [Install] y nos sale lo siguiente:

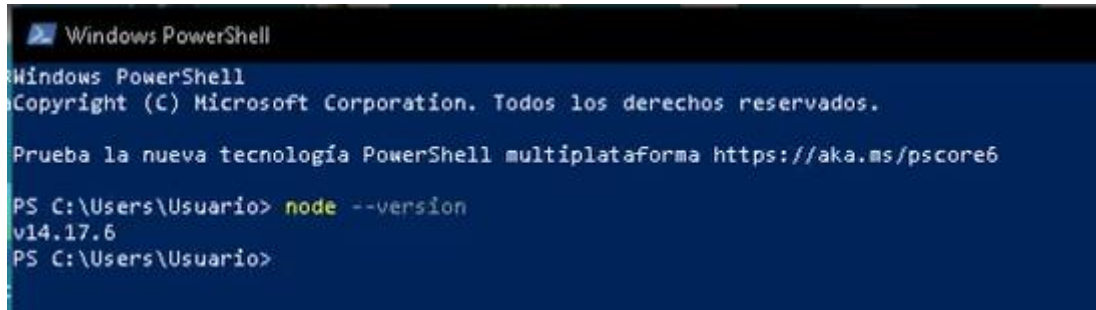


La instalación avanza hacia su terminación, y nos sale lo siguiente al llegar al final:



Pulsamos [Finish] y la pantalla desaparece.

Ahora comprobamos que la instalación fue correcta y la versión también, abriendo una terminal



```

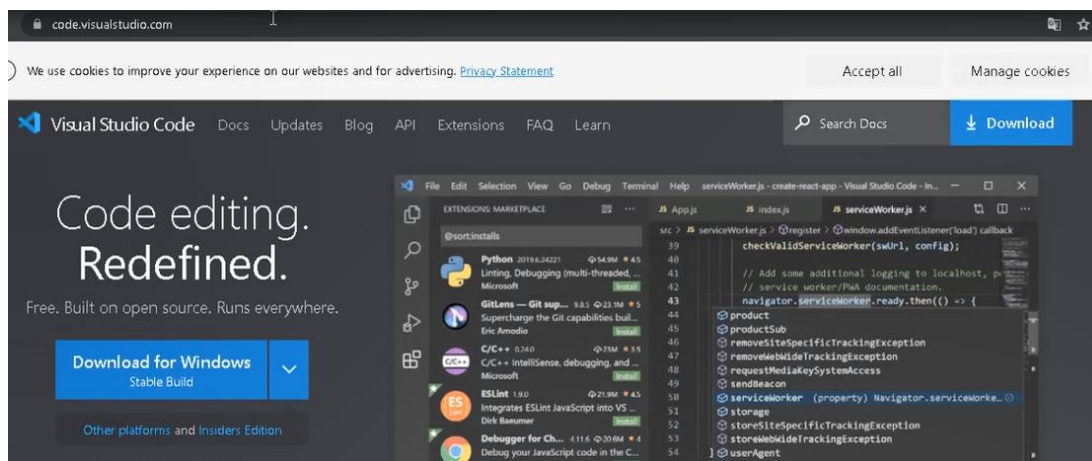
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

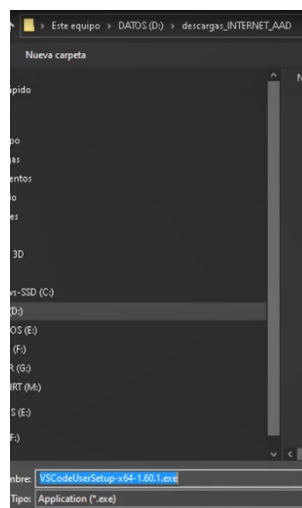
PS C:\Users\Usuario> node --version
v14.17.6
PS C:\Users\Usuario>
  
```

Al haber escrito `node --version` y haber recibido la versión correcta eso prueba que funciona.

A continuación, procedemos a la instalación del editor (**Visual Studio Code**) que usaremos para la subida al repositorio. Vamos a su página oficial:

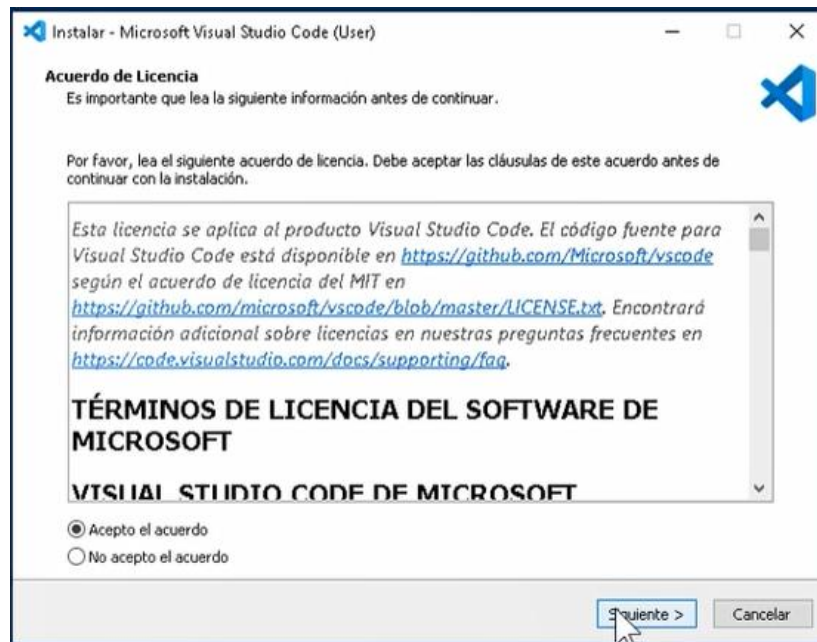


Pulsamos el botón [Download for Windows] y nos aparece lo siguiente:

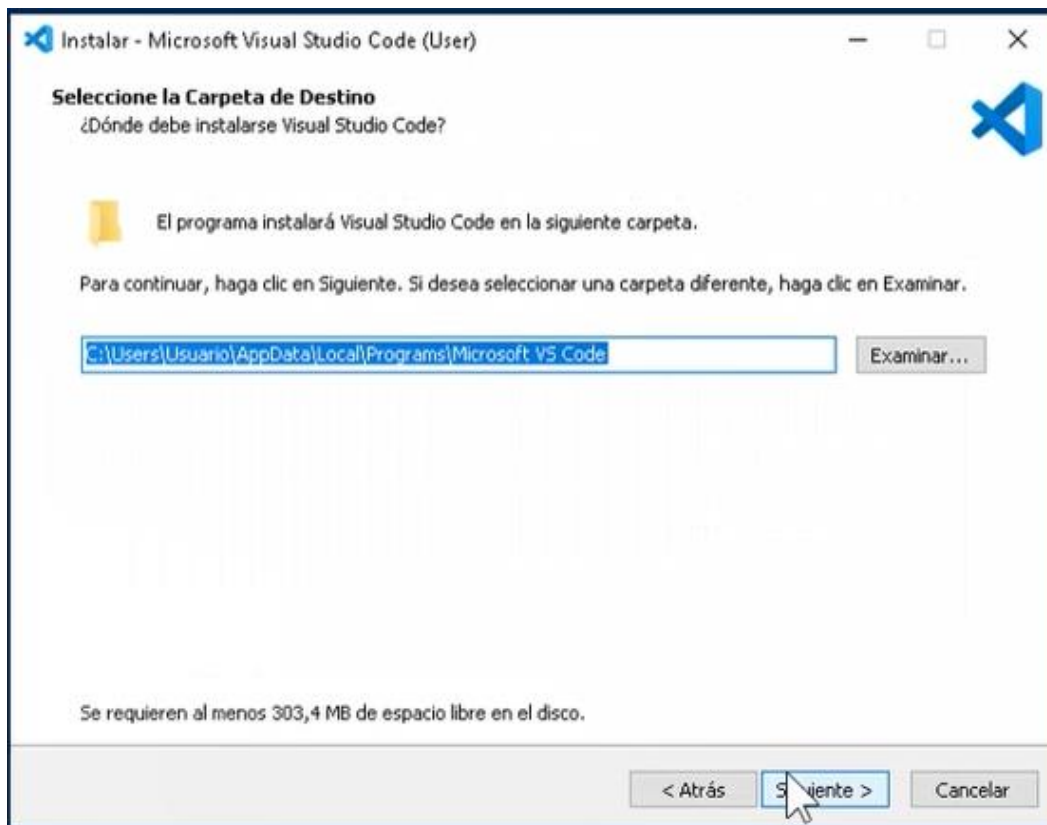




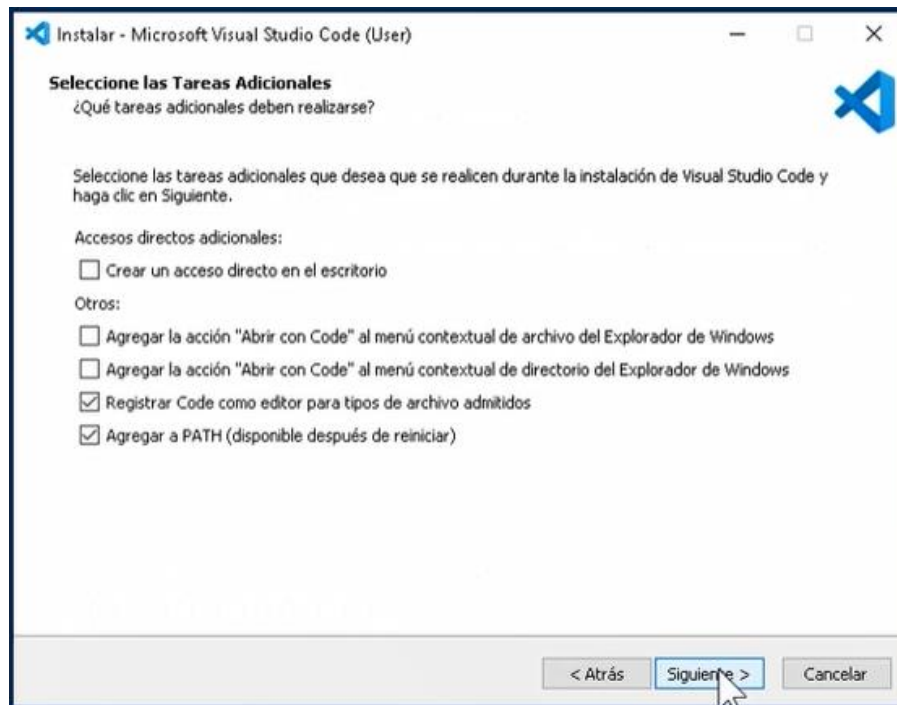
Ponemos el instalador \*.exe en la carpeta adecuada de nuestro PC para lanzar la instalación.



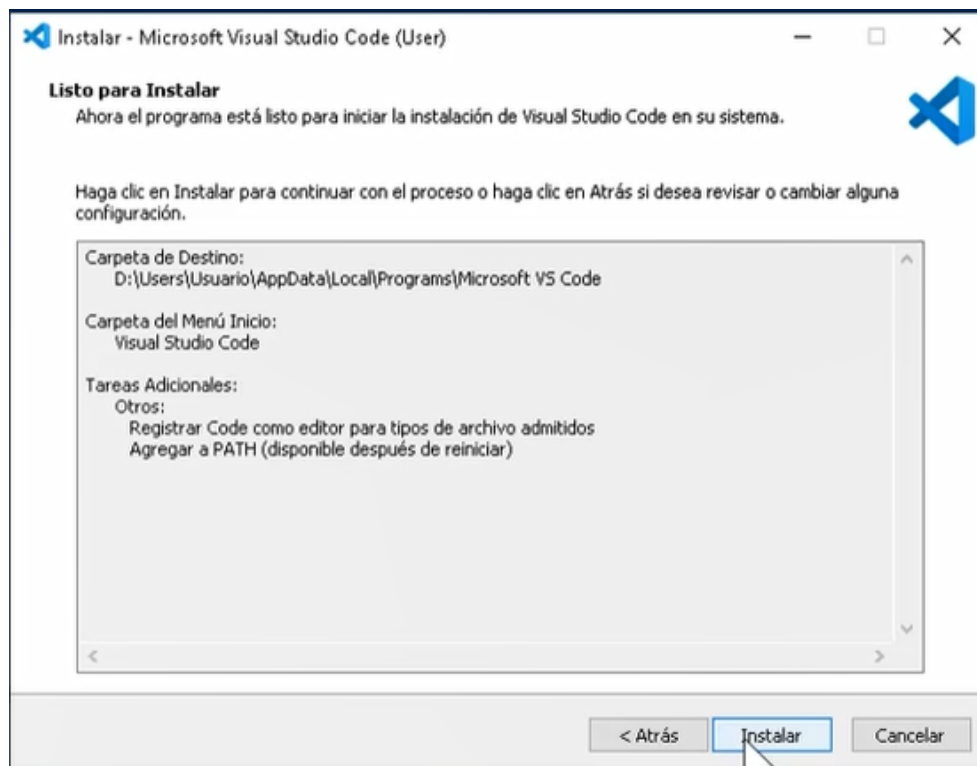
Aceptamos el acuerdo, tras leerlo. Pulsamos en [Siguiente] y nos sale esto:



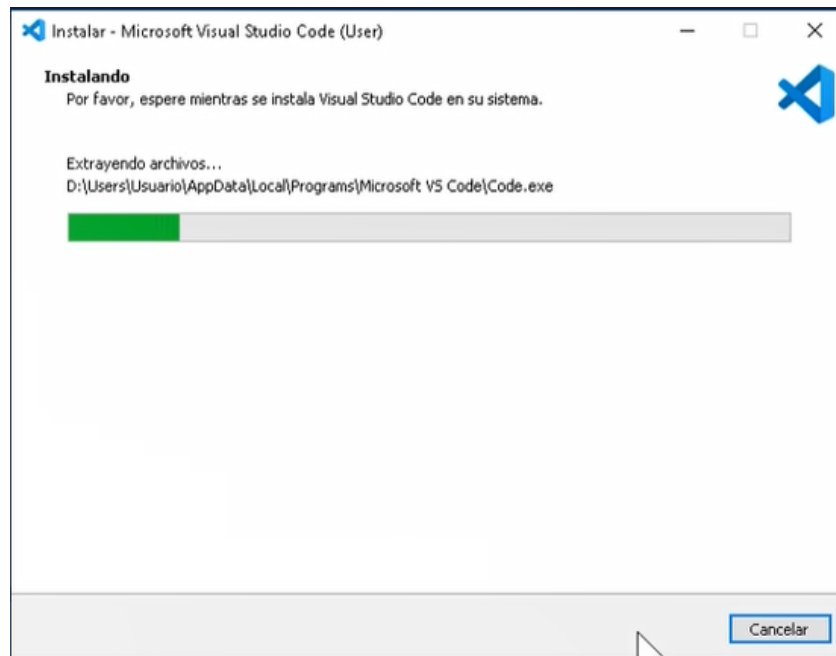
Elegimos la carpeta de destino por de la instalación. Pulsamos [Siguiente] y nos sale:



Pulsamos [Siguiente] y nos sale:



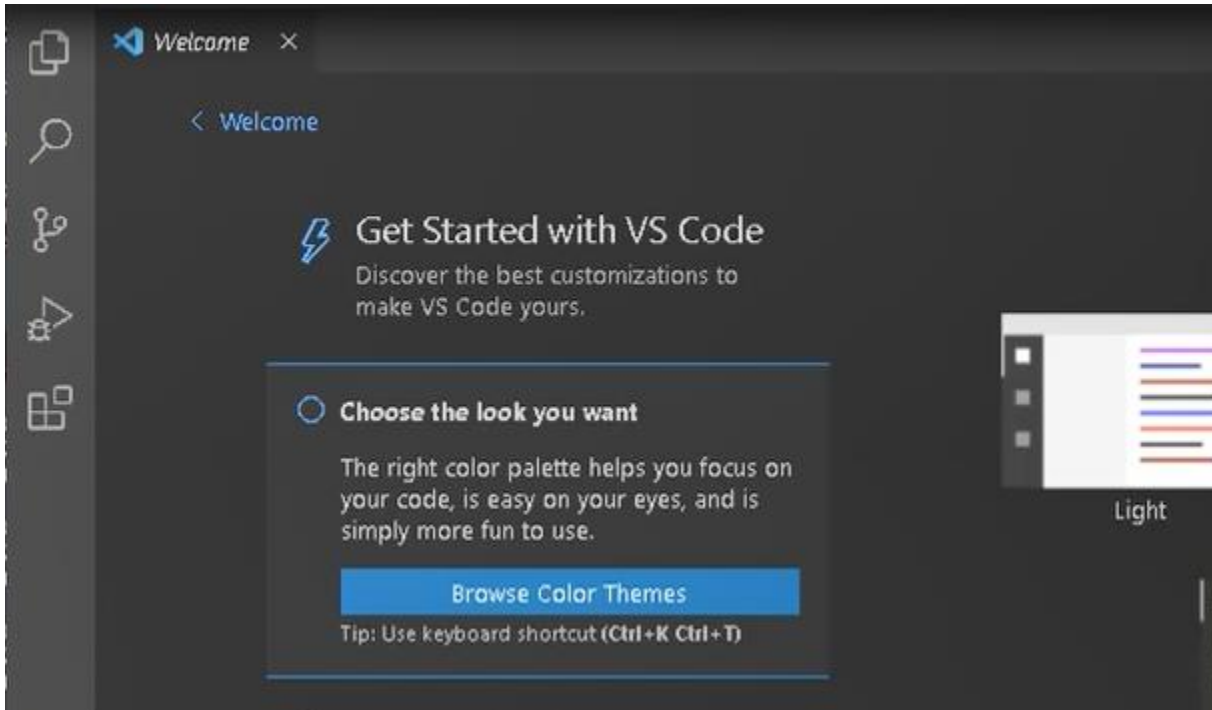
Comprobamos en el resumen que todo está bien. Pulsamos [Siguiente] y nos sale:



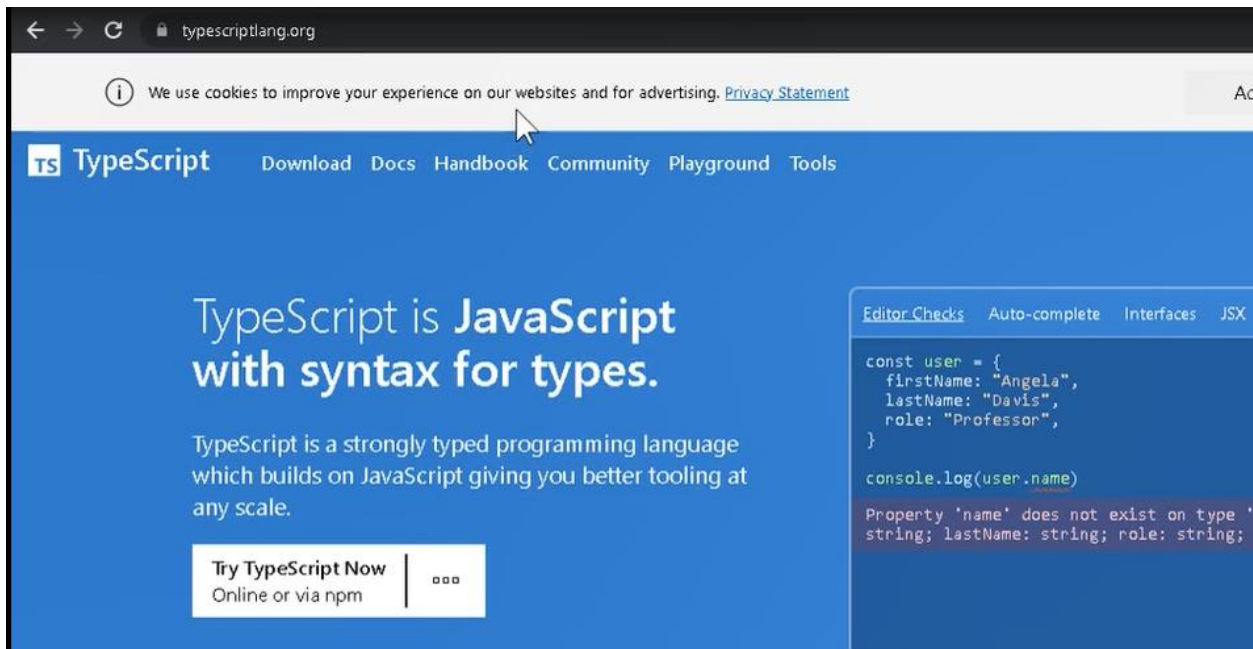
La instalación avanza hacia su terminación, y nos sale lo siguiente al llegar al final:



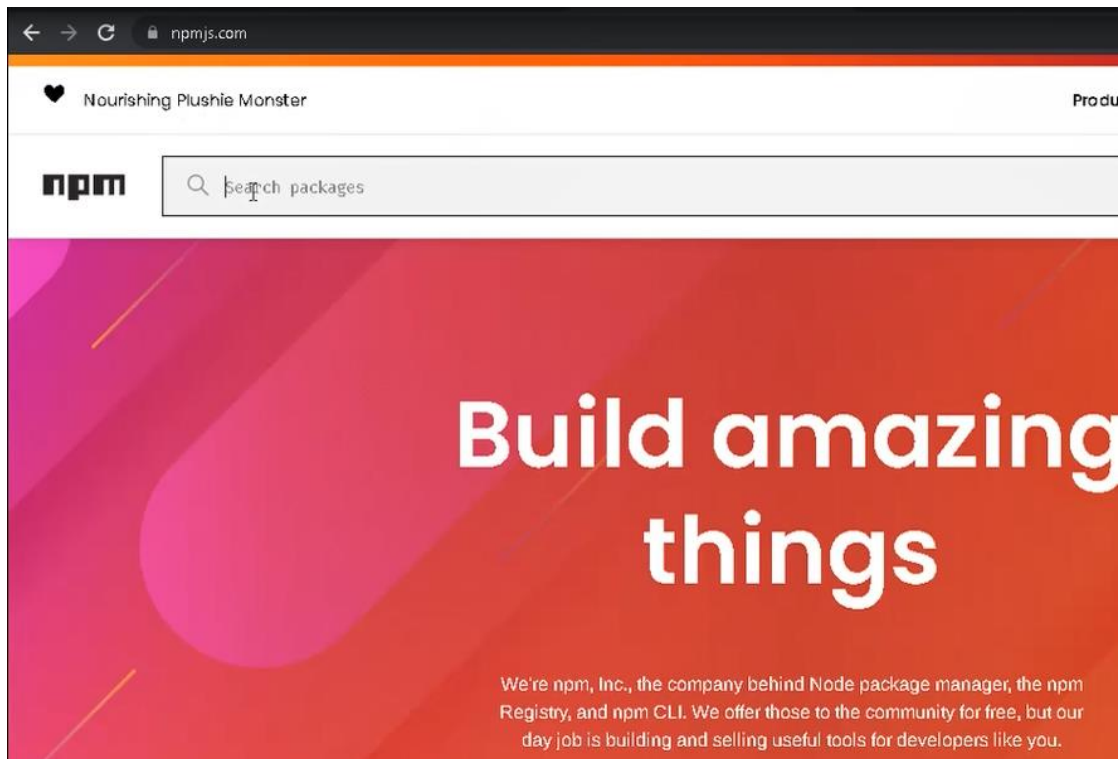
Al dejar marcada la opción “Ejecutar Visual Studio Code”, este se abre al dar a [Finalizar]



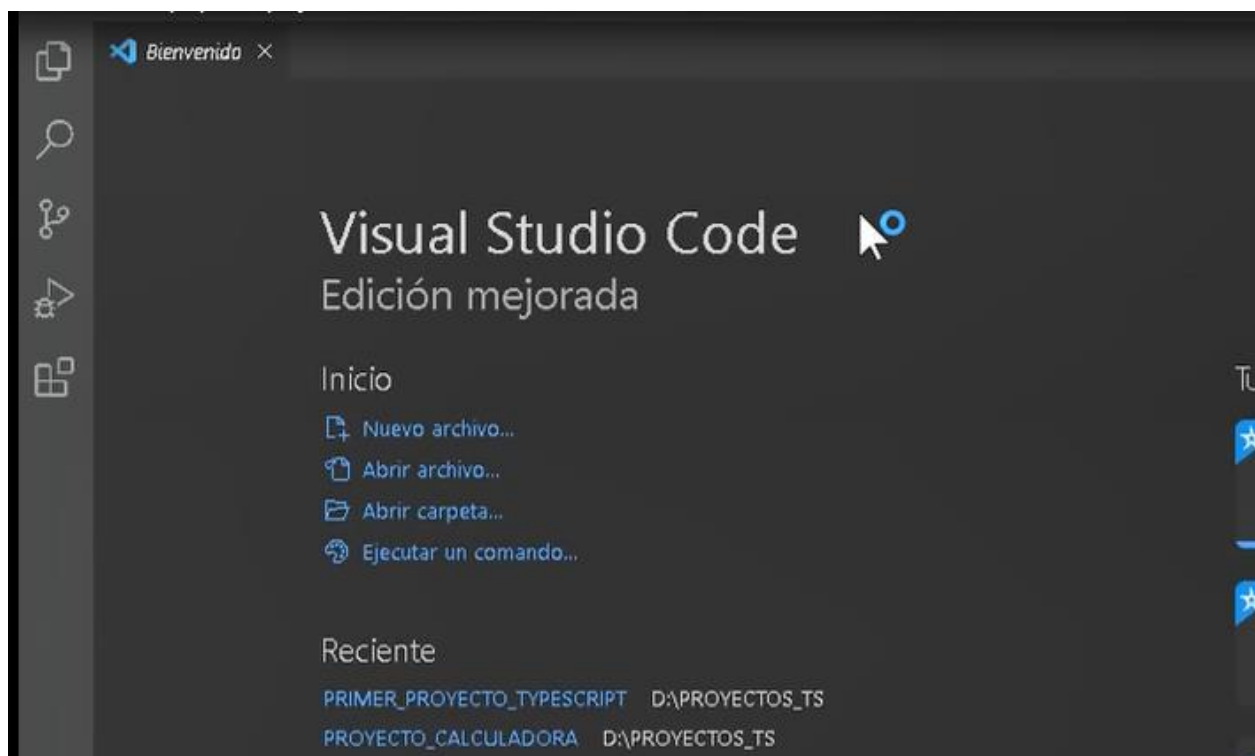
Ahora ya estamos listos para crear nuestro primer proyecto, para eso usaremos TypeScript empezando desde su página oficial:



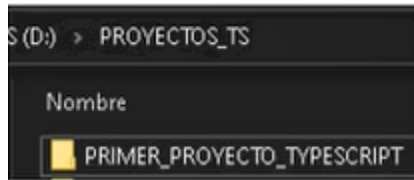
Y también usaremos los comandos **npm** que se encuentran en esta página oficial.



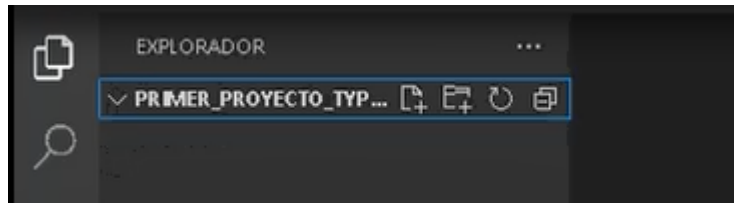
Con todo ello, procedemos a crear nuestro primer proyecto TypeScript. Abrimos VSC.



A continuación, creamos la carpeta de proyecto



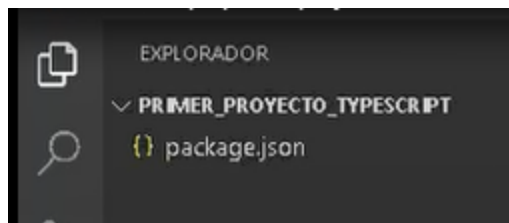
Arrastramos la carpeta a VSC y nos queda así:



En la terminal del proyecto se crea el proyecto node con: **npm init -y**



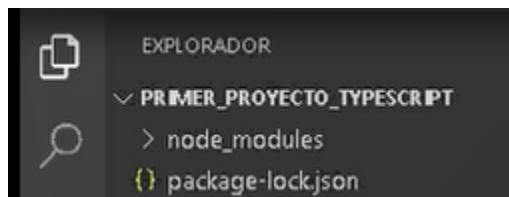
Se nos crea el **package.json**



Ahora escribimos en la terminal **npm install typescript -D**



Y nos crea esto:



A continuación escribimos en la terminal **tsc-init**

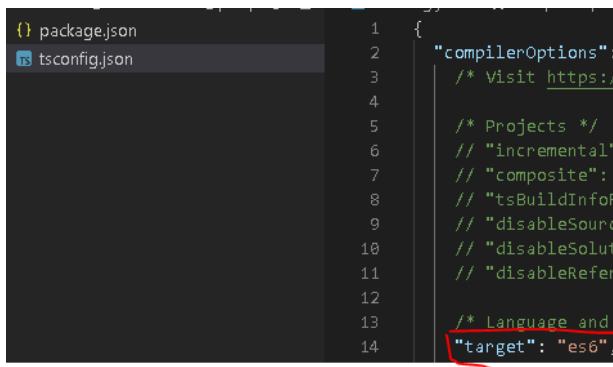
```
PS D:\PROYECTOS_TS\PRIMER_PROYECTO_TYPESCRIPT> tsc --init
message TS6071: Successfully created a tsconfig.json file.
```

Y nos crea el archivo **tsconfig.json**

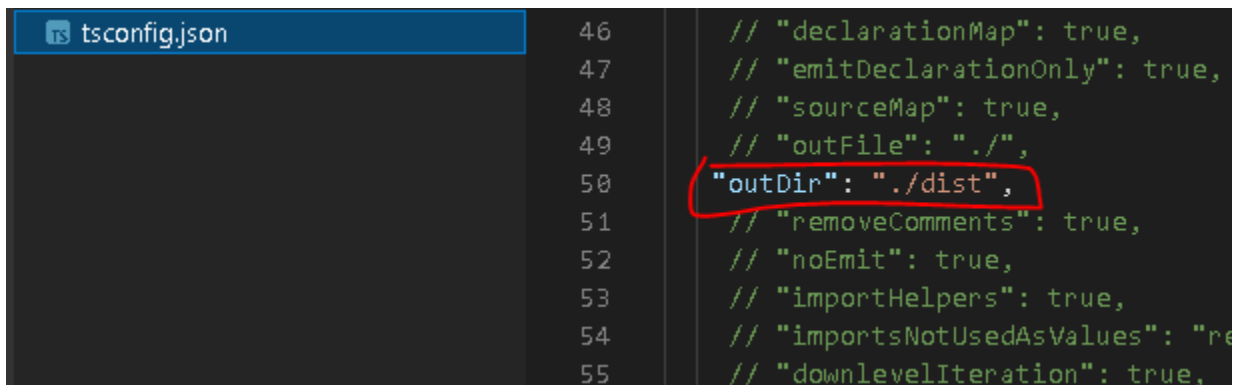


Vamos a dicho archivo para ver el contenido y hacemos los siguientes cambios:

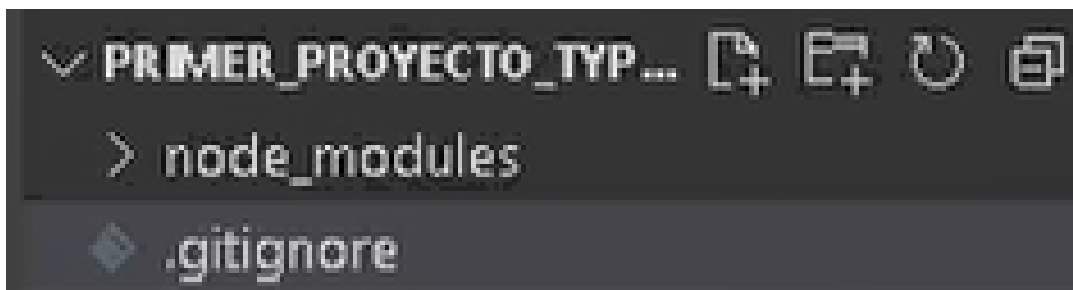
Cambiamos "target:" a "es6"



Cambiamos "outDir": "./dist",



Creamos un archivo **.gitignore**



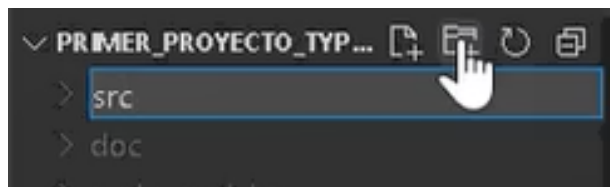
Y este es su contenido (para no subir a GitHub lo que ahí se especifica)



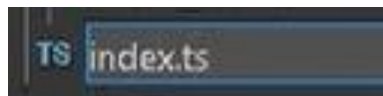
Creamos una carpeta **doc** para poner la documentación.



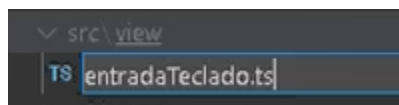
Creamos una carpeta **src** para poner los archivos con el código TS



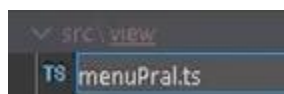
Dentro de **src** creamos el archivo **index.ts**



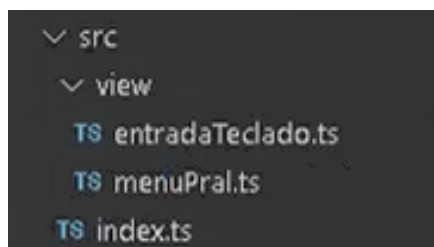
En **src** también creamos la carpeta **view** para poner dentro el archivo **entradaTeclado.ts**



Y en la misma carpeta ponemos otro archivo llamado **menuPral.ts**



Nos quedaría la siguiente estructura:





Ahora escribimos nuestro código en cada archivo \*.ts

Empezamos con el archivo **entradaTeclado.ts** (código que permite entrada de datos vía teclado)

```
src > view > TS entradaTeclado.ts > [x] leerTeclado
1  import readline from 'readline'
2  let readlineI: readline.Interface
3
4  let leeLinea = (prompt: string) => {
5      readlineI = readline.createInterface({
6          input: process.stdin,
7          output: process.stdout,
8      })
9      return new Promise<string>((resuelta: any, rechazada: any) => {
10         readlineI.question(`${prompt}: `, (cadenaEntrada: string) => {
11             resuelta(cadenaEntrada)
12         })
13     })
14 }
15
16 export let leerTeclado = async (prompt: string) => {
17     let valor: string
18     valor = await leeLinea(prompt)
19     readlineI.close()
20     return valor
21 }
```

Después el archivo **menuPral.ts** (código que importa datos vía teclado y los aplica al menú)

```
src > view > TS menuPral.ts > [x] menuPral
1  import { leerTeclado } from '../view/entradaTeclado'
2
3  export const menuPral = async () => {
4      let seleccionado: number
5      console.log('\n')
6      console.log('1.- ¿Es el número PRIMO?')
7      console.log('2.- ¿PAR o IMPAR?')
8      console.log('3.- Calcular superficie de un RECTÁNGULO')
9      console.log('4.- Comparar FECHAS')
10     console.log('0.- Salir')
11     seleccionado = parseInt(await leerTeclado('opción '))
12     return seleccionado
13 }
```

Y por último el archivo **index.ts** (que ejecuta las opciones del menú, según datos del teclado)

```
src> ts index.ts > [0] main
1 import { menuPral } from './view/menuPral'
2 import { leerTeclado } from './view/entradaTeclado'
3
4 const main = async () => {
5   let n: number
6   let n1: number, n2: number
7   let f1: string, f2: string
8   do {
9
10    n = await menuPral()
11    switch(n){
12      case 1:
13        console.log("Estoy en opción 1")
14        await primo()
15        break
16      case 2:
17        console.log("Estoy en opción 2")
18        n1 = parseInt( await leerTeclado('Dame un número para saber si es PAR o IMPAR'))
19        if (n1 % 2 == 0)
20          { console.log('Es PAR')}
21        else
22          { console.log('Es IMPAR')}
23        break
24      case 3:
25        console.log("Estoy en opción 3")
26        n1 = parseInt( await leerTeclado('Dame longitud de la base (en cms)'))
27        n2 = parseInt( await leerTeclado('Dame longitud de la altura (en cms)'))
28        console.log('La superficie del rectángulo (en cms cuadrados) es: ${superficie (n1, n2)}')
29        break
30      case 4:
31        console.log("Estoy en opción 4")
32        f1 = await leerTeclado('Dime la primera fecha (FECHA 1) para comparar (formato: YYYY-MM-DD)')
33        f2 = await leerTeclado('Dime la segunda fecha (FECHA 2) para comparar (formato: YYYY-MM-DD)')
34        console.log('fecha 1 en formato ISO es:', new Date(f1).toISOString());
35        console.log('fecha 2 en formato ISO es:', new Date(f2).toISOString());
36        if (f1 < f2)
37          { console.log('La FECHA 1 es anterior a la FECHA 2')}
38        else {
39          console.log('La FECHA 1 es posterior a la FECHA 2')
40        }
41        break
42      case 0:
43        break
44      default:
45        console.log("Opción incorrecta, introduzca solo las opciones del menú")
46        break
47    }
48  } while (n != 0)
49 }
50
51 const primo = async () => {
52   let n1: number
53   let i: number
54   n1 = parseInt( await leerTeclado('Dame el número para comprobar si es PRIMO o no'))
55   for (i=2; i<n1; i++)
56   {
57     if (n1 % i === 0)
58     {
59       console.log("FALSO, NO es primo")
60     }
61   }
62   else
63   {
64     console.log("VERDADERO, Sí es primo")
65   }
66   break
67 }
68
69 const par_impar = (n1: number): number => n1
70 const superficie = (n1: number, n2: number): number => n1 * n2
71 main()
```

Creamos un archivo **README.md** y ponemos el contenido.

```
① README.md > # Este es el github de **ANTONIO ÁVILA DELGADO** (2º ASIR) > ## En este proyecto usamos varias estrategias (siguiendo el modelo de Proyecto 003)
1 # Este es el github de **ANTONIO ÁVILA DELGADO** (2º ASIR)
2 ## En este proyecto usamos varias estrategias [siguiendo el modelo de Proyecto 003] para uso de funciones.
3 La que sigue MVC será la de la opción de SUPERFICIE de un rectángulo.
4 Hemos creado un proyecto TS con un menú de 4 opciones (ampliando las funcionalidades de Proyecto003):
5 * **Verificación de número primo (booleano)**
6
7 * **Comprobar número par o impar**
8
9 * **Cálculo geométrico área de un rectángulo**
10
11 * **Comparación de fechas**
```

Ahora se puede tener dos terminales abiertos.

En uno escribimos: **tsc -w**(para que compile ante cambios, es decir, en “watchmode”)

```
PROBLEMAS  SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS D:\PROYECTOS_TS\PRIMER_PROYECTO_TYPESCRIPT> tsc -w
```

Y nos sale esto:

```
PROBLEMAS  SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN

[20:01:36] Starting compilation in watch mode...

[20:01:37] Found 0 errors. Watching for file changes.
```

En otro terminal invocamos el ejecutable de JS que está en la carpeta **dist** usando **nodedist**

```
PROBLEMAS  SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS D:\PROYECTOS_TS\PRIMER_PROYECTO_TYPESCRIPT> node dist
```

Y aparece el menú, tal y como esperábamos.

```
PS D:\PROYECTOS_TS\PRIMER_PROYECTO_TYPESCRIPT> node dist

1.- ¿Es el número PRIMO?
2.- ¿PAR o IMPAR?
3.- Calcular superficie de un RECTÁNGULO
4.- Comparar FECHAS
0.- Salir
opción : █
```

Usamos la opción 1, y funciona.

```
opción : 1
Estoy en opción 1
Dame el número para comprobar si es PRIMO o no: 7
VERDADERO, Sí es primo
```

Usamos la opción 2, y funciona.

```
opción : 2
Estoy en opción 2
Dame un número para saber si es PAR o IMPAR: 18
Es PAR
```

Usamos la opción 3, y funciona.

```
opción : 3
Estoy en opción 3
Dame longitud de la base (en cms): 20
Dame longitud de la altura (en cms): 5
La superficie del rectángulo (en cms cuadrados) es: 100
```

Usamos la opción 4, y funciona.

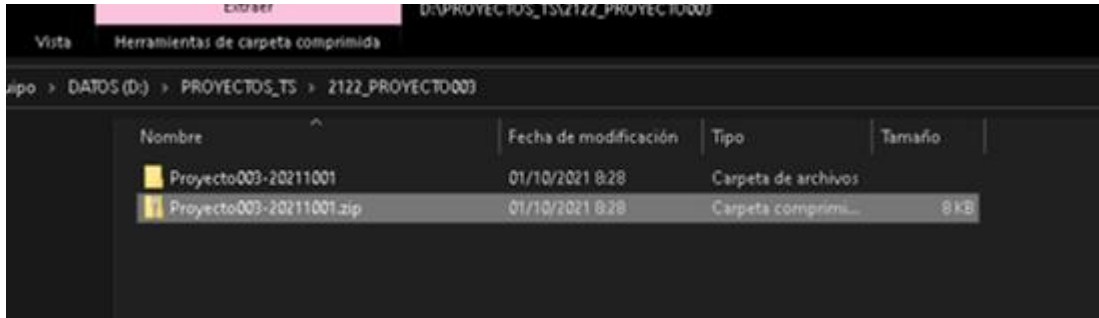
```
Estoy en opción 4
Dime la primera fecha (FECHA 1) para comparar (formato: YYYY-MM-DD): 2021-03-12
Dime la segunda fecha (FECHA 2) para comparar (formato: YYYY-MM-DD): 2021-10-07
fecha 1 en formato ISO es: 2021-03-12T00:00:00.000Z
fecha 2 en formato ISO es: 2021-10-07T00:00:00.000Z
La FECHA 1 es anterior a la FECHA 2
```

Por tanto, nuestro proyecto ha concluido con éxito.

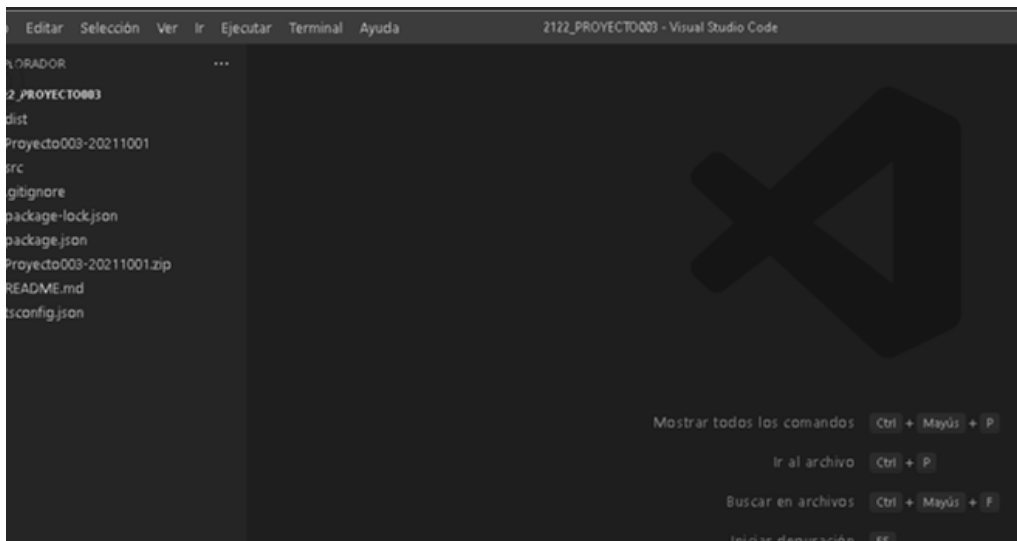
## IMPORTAR UN PROYECTO YA EXISTENTE

En este caso no vamos a crear un proyecto desde el principio, sino que vamos a importar uno ya existente llamado Proyecto003

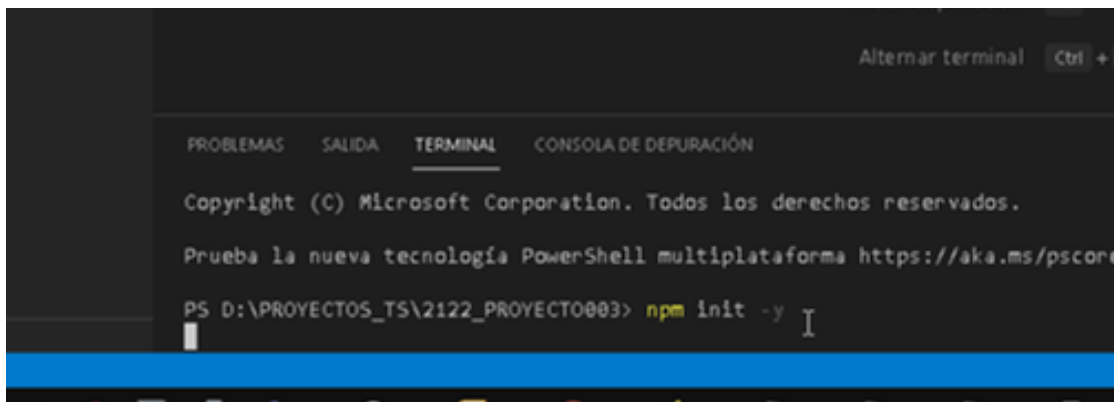
Recibimos el proyecto en formato \*.zip, y lo descomprimos



Primero arrastramos la carpeta a VSC



Lo iniciamos con **npm init -y**



Y después tenemos que instalarlo con **npm install** porque ya tiene un package.json

```
PS D:\PROYECTOS_TS\2122_PROYECTO003> npm install
```

Y se instalan los paquetes necesarios para que este proyecto ya existente que nos han dado pueda funcionar en nuestro equipo.

```
added 1 package from 43 contributors and audited 1 package in 2.182s
found 0 vulnerabilities
```

Lo que ya nos permite hacer el **tsc -w**

```
PROBLEMAS  SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN
[8:45:52] Starting compilation in watch mode...
[8:45:55] Found 0 errors. Watching for file changes.
```

Y ejecutarlo sin problemas con **node dist** y la ruta adecuada.

```
PS D:\PROYECTOS_TS\2122_PROYECTO003> node dist /pruebas/prueba

1.- Sumar
2.- Multiplicar
3.- Restar
0.- Salir
```

Por tanto la instalación de un proyecto que ya exista ha tenido éxito también.