

PROYECTO TYPESCRIPT
JUAN MANUEL CASTRO CABELLO
2ºASIR

TypeScript

Lo primero que vamos a realizar es la creación del archivo **package.json**, mediante **npm init** :

```
PS S:\Escritorio\INSTITUTO 2\bd\ProyectoTypeScriptJuanManuelCastroCabello> npm init
Wrote to S:\Escritorio\INSTITUTO 2\bd\ProyectoTypeScriptJuanManuelCastroCabello\package.json

{
  "main": "index.js",
  "doc": "doc",
},
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1"
},
"keywords": [],
"author": "",
"license": "ISC"
}
```


```
{ } package.json
```

Como veremos se nos habrá creado el archivo package.json.

Después instalaremos Typescript mediante `npm install typescript -D`, esto nos creará el archivo **package-lock.json**


```
PS S:\Escritorio\INSTITUTO 2\bd\ProyectoTypeScriptJuanManuelCastroCabello> npm install typescript -D
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN ProyectoTypeScriptJuanManuelCastroCabello@1.0.0 No description
npm WARN ProyectoTypeScriptJuanManuelCastroCabello@1.0.0 No repository field.

+ typescript@4.4.3
added 1 package from 1 contributor and audited 1 package in 13.821s
found 0 vulnerabilities
```

 package-lock.json

Acto seguido mediante el comando `npx tsc --init` iniciaremos typescript y nos creará el archivo de configuración de typescript **tsconfig.json**.

```
PS S:\Escritorio\INSTITUTO 2\bd\ProyectoTypeScriptJuanManuelCastroCabello> npx tsc --init
message TS6071: Successfully created a tsconfig.json file.
```

 tsconfig.json

En este archivo modificaremos los siguientes parámetros:

```
"target": "es6",
```

```
"outDir": "../dist",
```

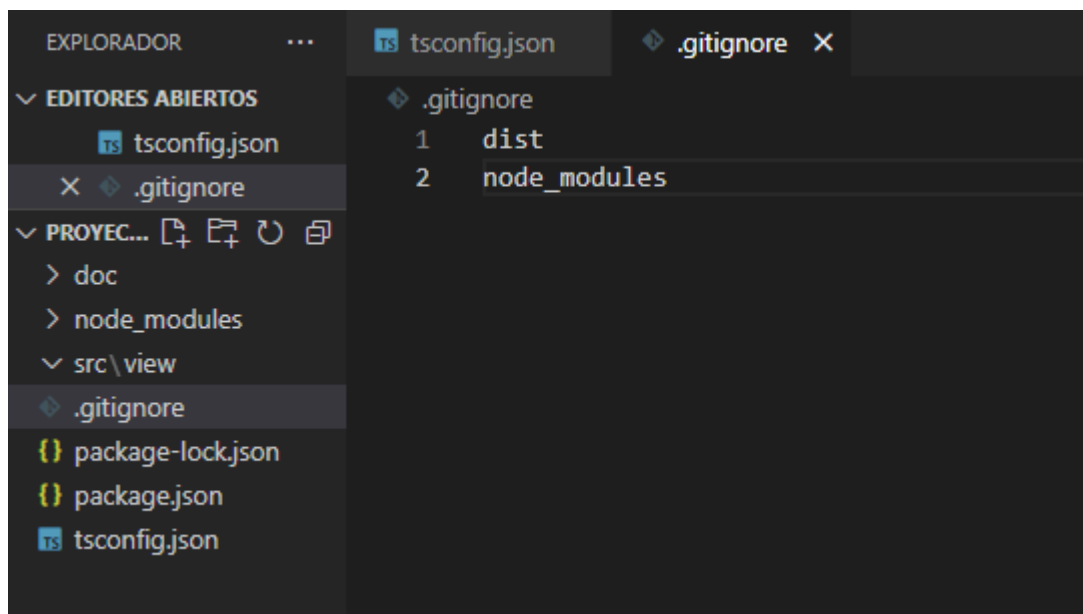
```
"exclude": [
  "node_modules"
]
```

Una vez realizado esto, crearemos la carpeta src, donde irá el index.ts y la carpeta view, donde encontraremos los archivos entradaTeclado.ts y menuPral.ts (donde colocaremos el menú principal de la práctica).

También crearemos el archivo .gitignore que nos servirá para ignorar los archivos que no queremos subir a nuestro repositorio, en dicho archivo vamos a poner la carpeta dist y node_modules .

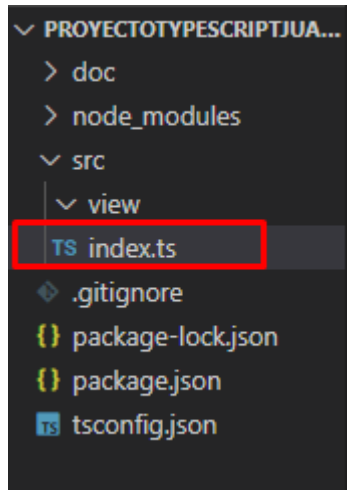
Creamos tambien la carpeta doc donde introduciremos esta documentación .

Dicho esto se nos quedará una estructura tal que así :

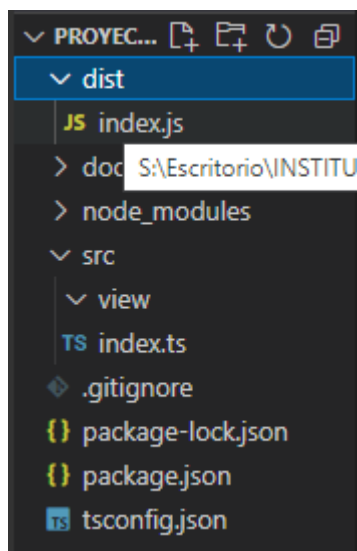
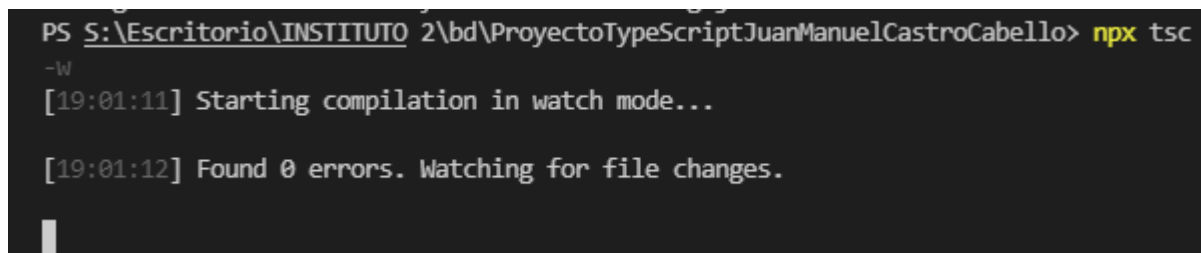


Ahora vamos a proceder con la explicación de la práctica.

Lo primero que vamos a crear es el archivo **index.ts** donde vamos a meter lo que va a realizar la práctica.



Después vamos a usar el comando **npx tsc -w** para que esté compilando mientras realizamos la práctica, esto lo introducirá en la carpeta dist.



Ahora vamos a ver el contenido de entradaTeclado.ts:

```
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 }
```

estos errores los vamos a solucionar con el comando:

npm install al @types/node

```
PS S:\Escritorio\INSTITUTO 2\bd\ProyectoTypeScriptJuanManuelCastroCabello> npm install @types/node
npm WARN ProyectoTypeScriptJuanManuelCastroCabello@1.0.0 No description
npm WARN ProyectoTypeScriptJuanManuelCastroCabello@1.0.0 No repository field.

+ @types/node@16.10.3
added 1 package from 42 contributors and audited 2 packages in 5.789s
found 0 vulnerabilities
```

Veamos el contenido de menuPral.ts, donde esta el menú principal de la práctica con las opciones disponibles:

```
1  import { leerTeclado } from '../view/entradaTeclado'
2
3  export const menuPral = async () => {
4      let n: number
5      console.log('\n')
6      console.log('1.-|-----NUMERO PRIMO-----|')
7      console.log('2.-|-----NUMERO PAR O IMPAR-----|')
8      console.log('3.-|-----FECHAS-----|')
9      console.log('0.-|-----SALIR-----|')
10     n = parseInt( await leerTeclado('opción: ') )
11     return n
12 }
```

Y el contenido de index.ts con la práctica en sí :

```
1  import { menuPral } from './view/menuPral'
2  import { leerTeclado } from './view/entradaTeclado'
3
4  const main = async () => {
5      let n: number
6      do {
7
8          n = await menuPral()
9          switch(n){
10             case 1:
11                 console.log("|-----OPCION 1-----|")
12                 await primo()
13                 break
14             case 2:
15                 console.log("|-----OPCION 2-----|")
16                 await parimpar()
17                 break
18             case 3:
19                 console.log("|-----OPCION 3-----|")
20                 await fechas()
21                 break
22             case 0:
23                 console.log('\n|-----ADIOS-----|')
24                 break
25             default:
26                 console.log("Opción incorrecta")
27                 break
28         }
29     }while (n != 0)
30 }
31
32
33
34
```




```
35  const primo = async () => {
36      let n1: number
37      let primo=true
38      n1 = parseInt( await leerTeclado('Dame un número'))
39      for (let x = 2; x < n1 / 2 ; x++) {
40          if (n1 % x == 0) {
41              primo=false
42          }
43      }
44      if (primo==true) {
45          console.log("El numero es primo")
46      } else {
47          console.log("El numero no es primo")
48      }
49  }
50  const parimpar = async () => {
51      let n1: number
52      n1 = parseInt( await leerTeclado('Dame un número'))
53      if (n1 % 2 == 0) {
54          console.log("El número es par");
55      } else {
56          console.log("El número es impar");
57      }
58  }
59  }
60  const fechas = async () => {
61      console.log ("Inserte fecha para el primer usuario (aaaa-mm-dd)")
62      let fecha1A= await(leerTeclado(' '))
63      console.log ("Inserte fecha para el segundo usuario (aaaa-mm-dd)")
64      let fecha2A= await(leerTeclado(' '))
65      let fecha1Buena= new Date(fecha1A)
66      let fecha2Buena = new Date(fecha2A)
67      if (fecha1Buena.getTime() < fecha2Buena.getTime()) {
68          console.log("El usuario 1 es mayor")
69      } else {
70          console.log ("El usuario 2 es mayor")
71      }
72  }
73  }
74  main()
```

Ahora vamos a proceder a subirlo al repositorio.

Primero iniciamos sesión en Github y creamos un nuevo repositorio.

Create a new repository

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

Owner *
 castrocabellojuanmanuel2


Repository name *


Proyecto003_JuanManuelCastroC ✓

Proyecto003_JuanManuelCastroCabello is available. to?

Great repository names are short and memorable.

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 el comando git init para iniciar el proceso de subida:

```
PS S:\Escritorio\INSTITUTO 2\bd\ProyectoTypeScriptJuanManuelCastroCabello> git init
Initialized empty Git repository in S:/Escritorio/INSTITUTO 2/bd/ProyectoTypeScriptJuanManuelCastroCabello/.git/
```

ponemos git add. para añadir los archivos y hacemos commit

```
PS S:\Escritorio\INSTITUTO 2\bd\ProyectoTypeScriptJuanManuelCastroCabello> git commit -m "first commit"
[master (root-commit) d2afbfb] first commit
7 files changed, 252 insertions(+)
create mode 100644 .gitignore
create mode 100644 package-lock.json
create mode 100644 package.json
create mode 100644 src/index.ts
create mode 100644 src/view/entradaTeclado.ts
create mode 100644 src/view/menuPral.ts
create mode 100644 tsconfig.json
```

ponemos el comando git branch y git remote para subirlo a nuestro repositorio.

```
PS S:\Escritorio\INSTITUTO 2\bd\ProyectoTypeScriptJuanManuelCastroCabello> git branch -M main
PS S:\Escritorio\INSTITUTO 2\bd\ProyectoTypeScriptJuanManuelCastroCabello> git remote add origin https://github.com/castrocabellojuanmanuel2/Proyecto003_JuanManuelCastroCabello.git
PS S:\Escritorio\INSTITUTO 2\bd\ProyectoTypeScriptJuanManuelCastroCabello>
```

y hacemos git push para lanzarlos

```
PS S:\Escritorio\INSTITUTO 2\bd\ProyectoTypeScriptJuanManuelCastroCabello> git push -u origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (11/11), 5.40 KiB | 1.35 MiB/s, done.
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/castrocabellojuanmanuel2/Proyecto003_JuanManuelCastroCabello.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```