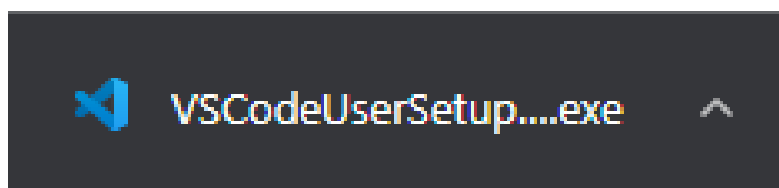
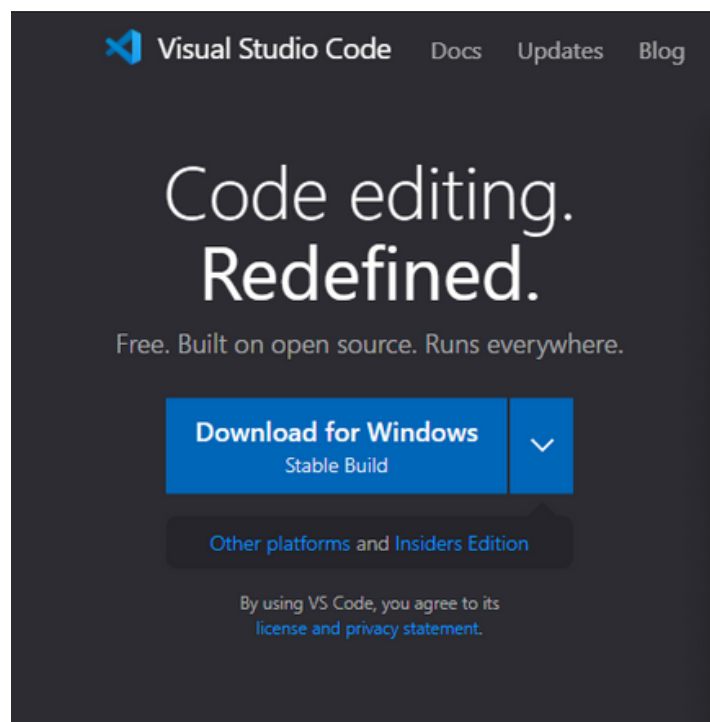




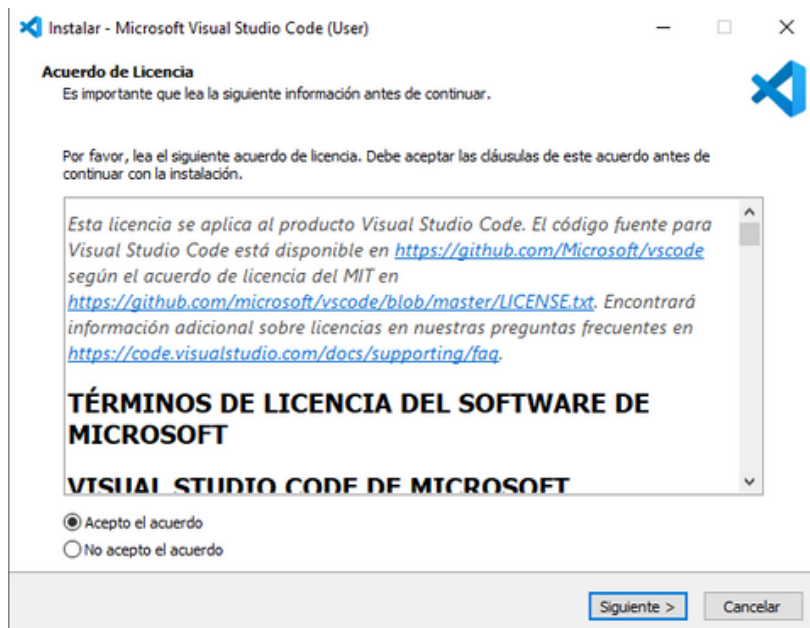
By Álvaro Maseri Apraiz

¿Cómo realizar un proyecto TypeScript?

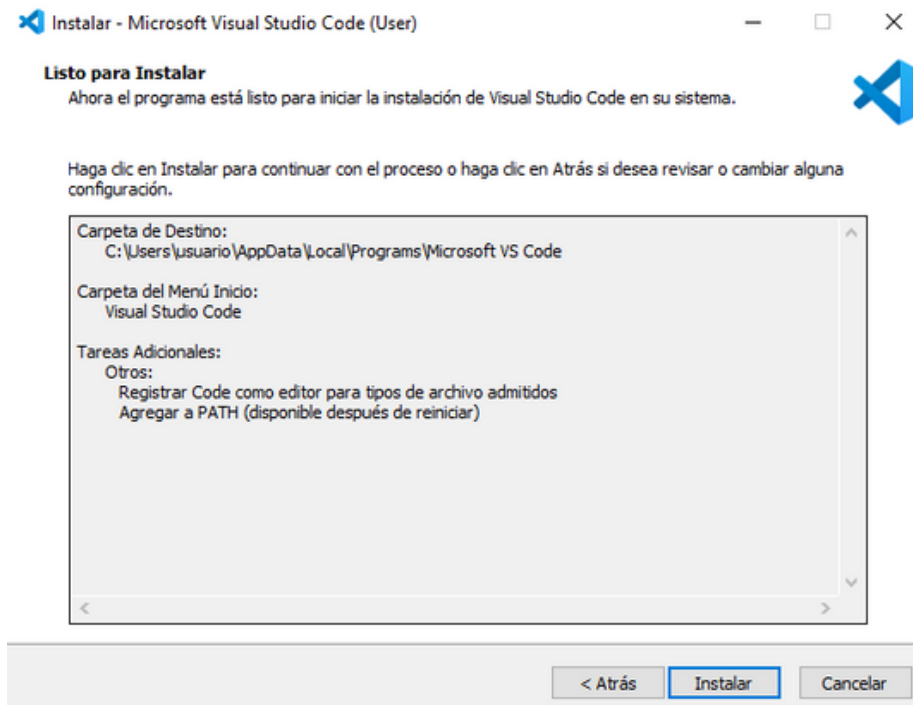
Comenzaremos instalando un editor de código fuente en este caso Visual Studio Code



Ejecutamos



Aceptamos, configuramos al gusto
e instalamos



Ahora instalaremos el Node.js



Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome.

New security releases to be made available October 12th, 2021

Descargar para Windows (x64)

14.18.0 LTS

Recomendado para la mayoría

16.10.0 Actual

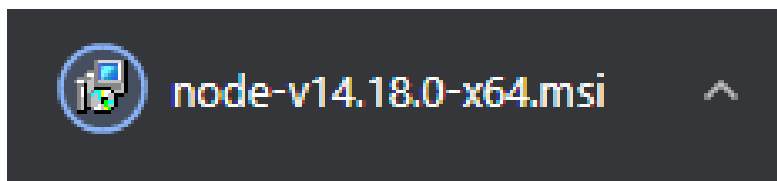
Últimas características

[Otras Descargas](#) | [Cambios](#) | [Documentación del API](#)

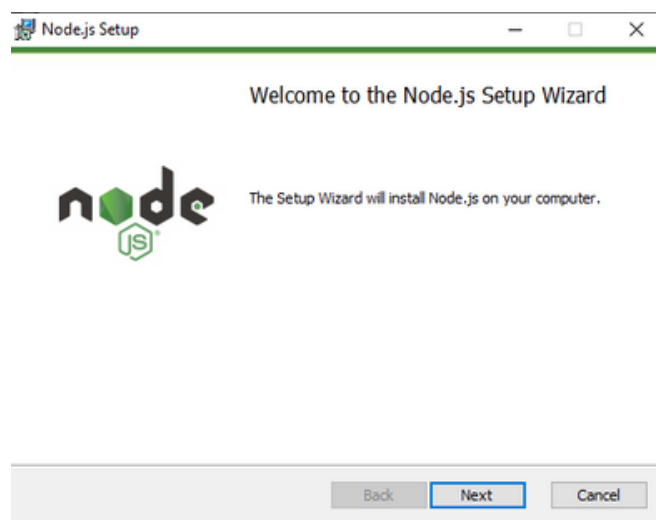
[Otras Descargas](#) | [Cambios](#) | [Documentación del API](#)

O eche un vistazo a la [Programa de soporte a largo plazo \(LTS\)](#)

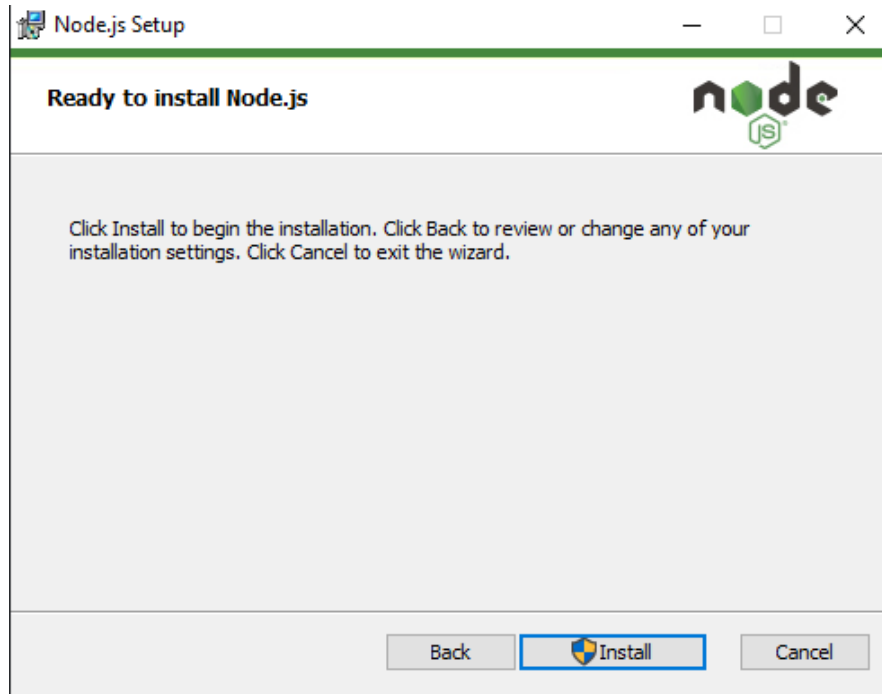
Versión 14.18 LTS



Ejecutamos



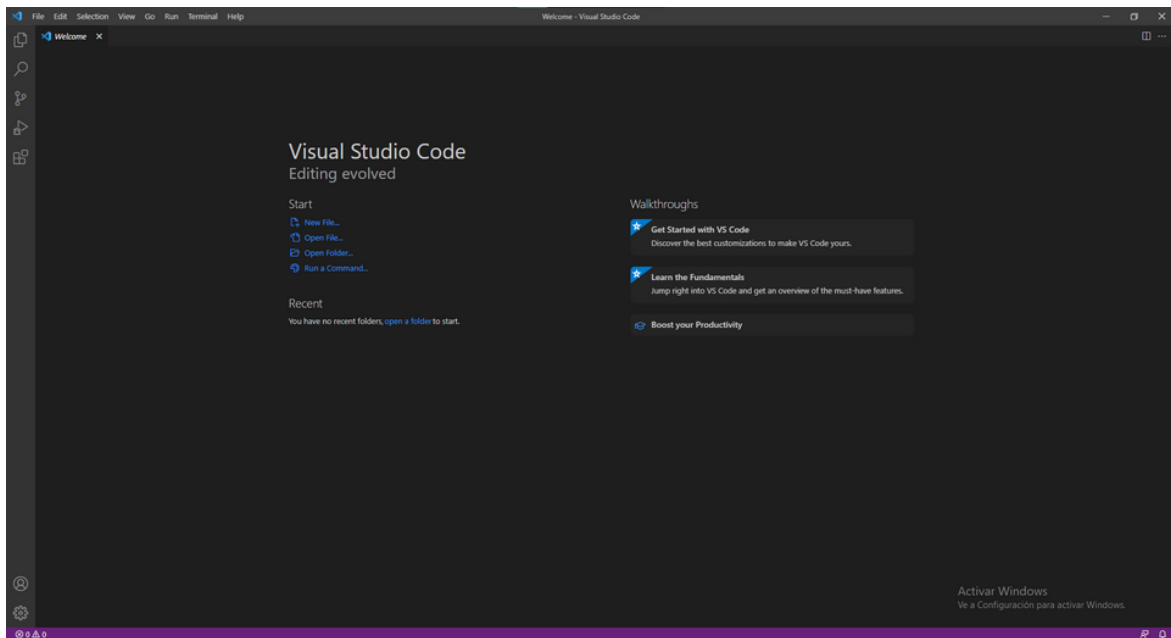
Aceptamos, configuramos al gusto
e instalamos



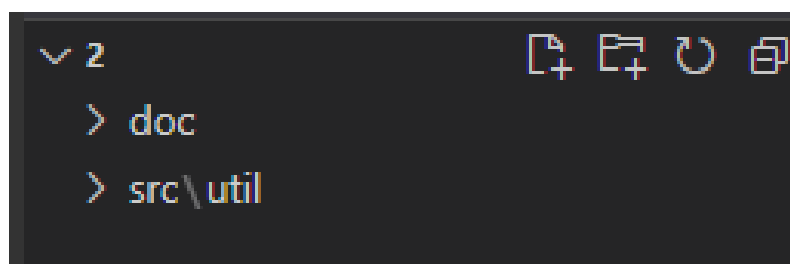
Ya procederemos a trabajar con el editor de código fuente
Abrimos Visual Studio Code



Primera visión del Visual Studio Code



Para trabajar en este entorno utilizando JavaScript la siguiente organización



Ahora descargaremos lo necesario para trabajar con JavaScript

```
npm install -g typescript
```

```
+ typescript@4.4.3  
updated 1 package in 0.951s
```

```
npm init -y
```

Iniciamos el siguiente paquete

```
{ } package.json
```

```
{ } package.json U x
{ } package.json > ...
5   "main": "index.js",
6   "directories": {
7     "doc": "doc"
8   },
9   "scripts": {
10    "test": "echo \"Error: no test specified\" && exit 1"
11  },
12  "keywords": [],
13  "author": "",
14  "license": "ISC"
15
16
```

Para que la política de ejecución no sea restrictiva

```
Set-executionPolicy unrestricted
```

entonces

```
tsc --init
```

Archivo de configuración

```
TS tsconfig.json
```

Donde
actualizamos la versión del lenguaje
java cambiando es5 por es6

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

Donde estarán nuestro JavaScripts

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

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

Instalamos

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

```
{  
  "package-lock.json"  
}
```

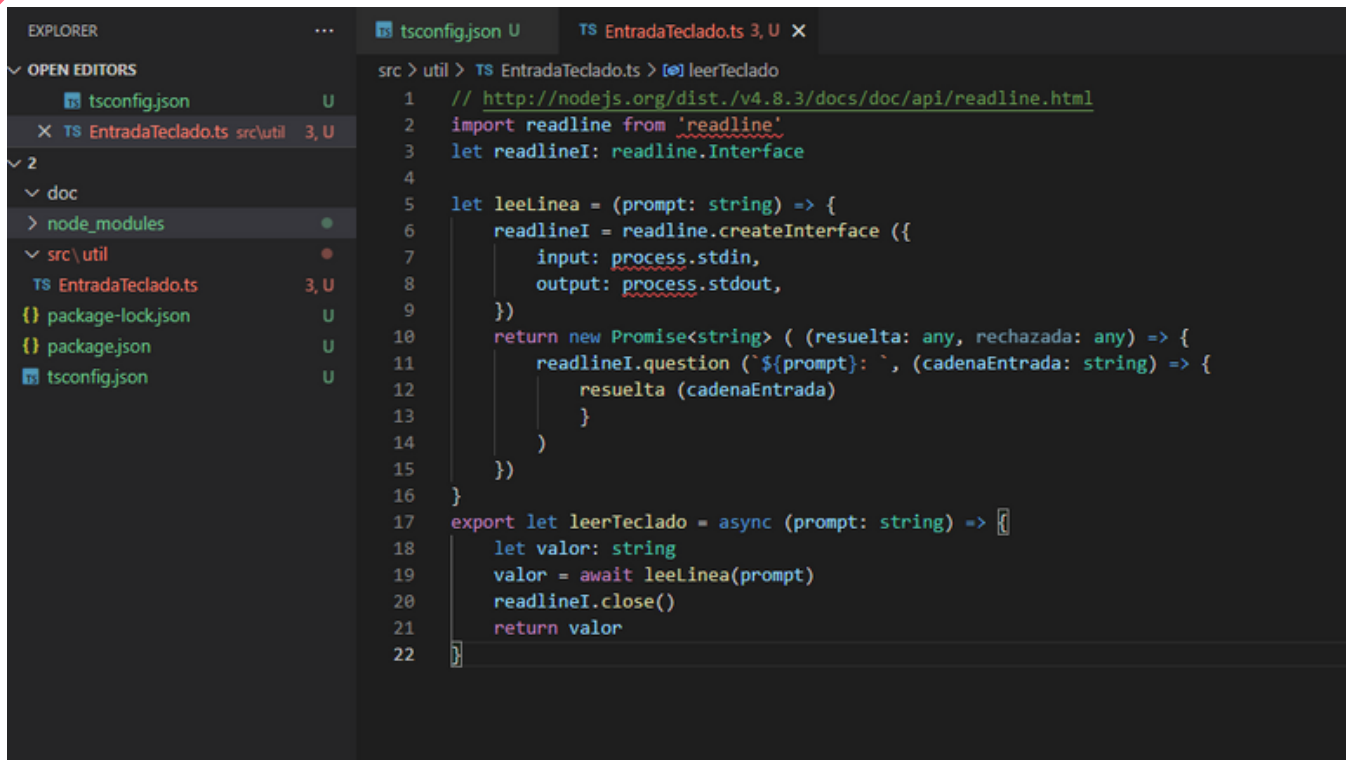
Para desinstalar

```
npm uninstall typescript
```

Con todo esto listo empezaremos un el proyecto

```
✓ doc  
> node_modules  
✓ src\util  
{ } package-lock.json U  
{ } package.json U  
TS tsconfig.json U
```

Primero añadiremos la entrada
de teclado
Dentro de util



The screenshot shows a VS Code editor with the Explorer sidebar on the left and a code editor on the right. The Explorer sidebar shows the project structure with 'src\util' expanded, containing 'EntradaTeclado.ts'. The code editor shows the implementation of 'leerTeclado' in 'EntradaTeclado.ts'. The code imports 'readline' from 'node:readline' and uses 'readline.createInterface' to create a readline interface. It then defines 'leerLinea' as a function that prompts the user and returns a Promise. Finally, 'leerTeclado' is an async function that calls 'leerLinea' and returns the result.

```
1 // http://nodejs.org/dist/v4.8.3/docs/doc/api/readline.html
2 import readline from 'readline'
3 let readlineI: readline.Interface
4
5 let leerLinea = (prompt: string) => {
6   readlineI = readline.createInterface ({
7     input: process.stdin,
8     output: process.stdout,
9   })
10   return new Promise<string> ( (resuelta: any, rechazada: any) => {
11     readlineI.question ( `${prompt}: `, (cadenaEntrada: string) => {
12       resuelta (cadenaEntrada)
13     }
14   )
15 })
16 }
17 export let leerTeclado = async (prompt: string) => {
18   let valor: string
19   valor = await leerLinea(prompt)
20   readlineI.close()
21   return valor
22 }
```

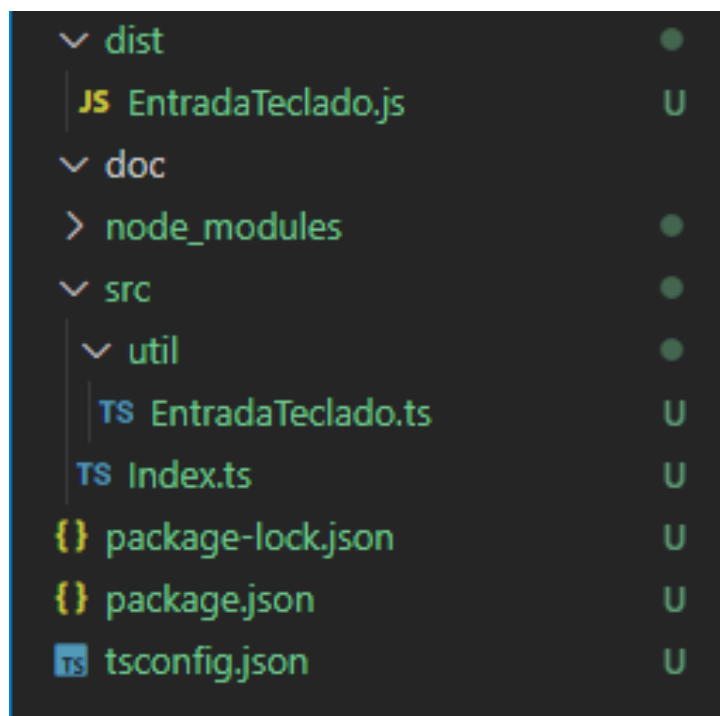
Para solucionar los errores ejecutamos

```
npm install @types/node
```

Instalando los nodos necesarios
Ahora crearemos el archivo Index.ts
dentro de src

```
TS Index.ts
```


Tendríamos la siguiente estructura



En el archivo Index.ts es donde trabajaremos aquí van unos ejemplos

Ejemplo1 --> Dame un numero y te digo si es primo o no:

```
import { leerTeclado } from "../util/EntradaTeclado";

let main = async () => {
  const num1 = parseInt (await leerTeclado('dame un numero'))
  let primo = 1
  if (num1 == 1 || num1 == 0){
    console.log('Este numero es primo');}
  else {
    for (var i = 2 ; i<num1 ; i++){
      if (num1%i==0){
        primo = 0
      }
    }
  }
  if (primo==1){
    console.log('su numero es primo')
  }
  else {
    console.log('su numero no es primo')
  }
}

main()
```

Con el comando

```
tsc -W
```

Compilamos el JavaScript el cual se encuentra en la carpeta dist tal y como configuramos anteriormente

```
▼ dist ●
  > util ●
  JS EntradaTeclado.js U
  JS Index.js U
```

Con el comando

```
node dist
```

Ejecutamos el JavaScript y nos comenta la solución

```
PS C:\Users\usuario\Desktop\PUNTA_DEL_VERDE\base de datos\1º trimestre\
Documentaciones\2> node dist
dame un numero: 97
su numero es primo
```

Ejemplo2 --> Compara tres números

```
let main = async () => {
  const A = parseInt (await leerTeclado('dame tres numeros distintos'))
  const B = parseInt (await leerTeclado('dame otro'))
  const C = parseInt (await leerTeclado('dame last one'))

  if (A>B && B>C) {
    console.log (`El numero ${A} es mayor que ${B}
    que a la vez es mayor que ${C}`)
  }
  else if (B>C && C>A) {
    console.log (`El numero ${B} es mayor que ${C}
    que a la vez es mayor que ${A}`)
  }
  else if (C>B && B>A) {
    console.log (`El numero ${C} es mayor que ${B}
    que a la vez es mayor que ${A}`)
  }
  else if (C>A && A>B) {
    console.log (`El numero ${C} es mayor que ${A}
    que es mayor que ${B}`)
  }
  else if (B>A && A>C) {
    console.log (`El numero ${B} es mayor que ${A}
    que a la vez es mayor que ${C}`)
  }
  else if (A>C && C>B) {
    console.log (`El numero ${B} es mayor que ${A}
    que a la vez es mayor que ${C}`)
  }

  else if (B==A && A==C) {
    console.log (`Los tres numero son iguales`)}
}

main()
```

node dist

```
PS C:\Users\usuario> node dist UNTA_DEL_VERDE\base de datos\1º trimetre\Documentaciones\2>
dame tres numeros distintos: 10
dame otro: 4
dame last one: 5
El numero 4 es mayor que 10
que a la vez es mayor que 5
```

Ejemplo3 --> Precio Skate

Le pasamos la marca y el ancho.

```
import { leerTeclado } from "../util/EntradaTeclado"

class Skate {
  private x: string
  private y: number
  private z: number
  private _o: number
  constructor(){
    this.x= ""
    this.y= 0
    this.z= 0
    this._o= 0
  }
  gama(x:string){
    if (x='family') {
      this.z = 0.4
    }
    else if (x='enjoy') {
      this.z = 0.10
    }
    else {
      this.z = 0.21
    }
  }
  ancho(y:number){
    if (y > 8.5) {
      this.y = 100
    }
    else if (y > 7.5) {
      this.y = 80
    }
  }
}
```

```
    else {
      this.y = 50
    }
  }
  precio(){
    this._o = this.y*this.z + this.y
  }
  get o (){
    return this._o
  }
}

let main = async () => {
  const gama1 = await leerTeclado('¿De que marca es tu skate?')
  const ancho1 = parseInt(await leerTeclado('¿Cual es el ancho?'))
  let S1 : Skate
  S1= new Skate ()
  S1.gama(gama1)
  S1.ancho(ancho1)
  S1.precio()
  console.log(`mamahuevo tu tabla vale: ${S1.o}`)
}

main()
```

node dist

```
¿De que marca es tu skate?: family
¿Cual es el ancho?: 10
mamahuevo tu tabla vale: 140
```

Ahora vamos a subirlo al repositorio de GitHub pero antes debemos crear este archivo

```
❖ .gitignore
```

Y dentro escribiremos

```
dist  
node_modules
```

Para subirlo a Git

```
$ git init
```

```
$ git add .
```

Añadimos los archivos

hacemos commit de los cambios

```
$ git commit -m subimos
[master (root-commit) c1c09d5] subimos
3 files changed, 154 insertions(+)
create mode 100644 2/.gitignore
create mode 100644 2/src/Index.ts
create mode 100644 2/src/util/EntradaTeclado.ts
```

añadimos el remoto

```
$ git add remote origin https://github.com/MASERIAPRAIZALVARO/1-JavaScripts.git
fatal: pathspec 'remote' did not match any files
```

```
$ git push -u origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 1.56 KiB | 1.56 MiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/MASERIAPRAIZALVARO/1-JavaScripts/pull/new/master
remote:
To https://github.com/MASERIAPRAIZALVARO/1-JavaScripts.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

y lo subimos al repositorio remoto

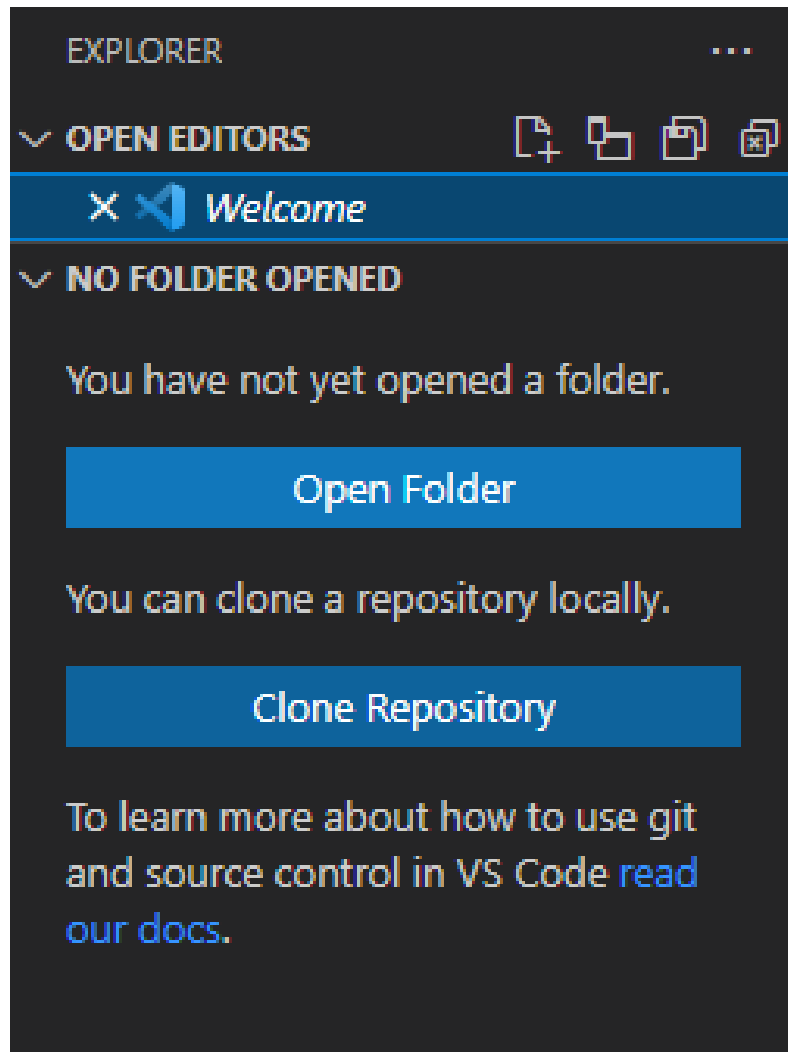
Para hacer la instalación de un proyecto ya existente utilizaremos el Git clone

```
usuario@DESKTOP-SQ7I4S0 MINGW64 ~/Desktop/proyectos/GBD/typescript1/2/clonacion
(master)
$ git clone https://github.com/MASERIAPRAIZALVARO/BasedeDatos.git
Cloning into 'BasedeDatos'...
remote: Enumerating objects: 502, done.
remote: Counting objects: 100% (502/502), done.
remote: Compressing objects: 100% (287/287), done.
Receiving objects: 94% (472/502), 4. remote: Total 502 (delta 199), reused 480 (
delta 177), pack-reused 0
Receiving objects: 100% (502/502), 5.56 MiB | 8.06 MiB/s, done.
Resolving deltas: 100% (199/199), done.
```

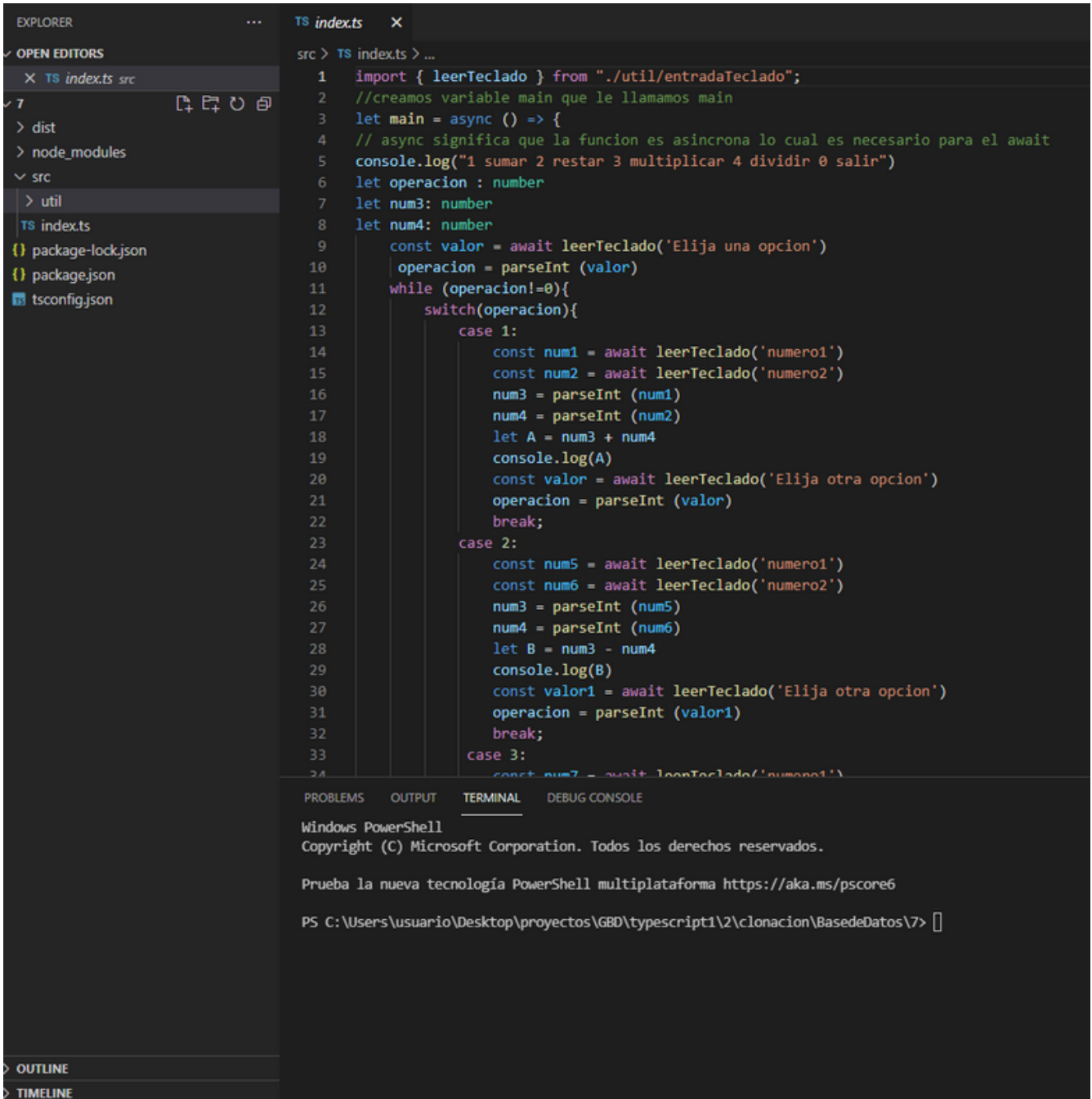
Clonación realizada!!!!

Este equipo > Escritorio > proyectos > GBD > typescript1 > 2 > clonacion > BasedeDatos				
Nombre	Fecha de modificación	Tipo	Tamaño	
.git	07/10/2021 23:43	Carpeta de archivos		
7	07/10/2021 23:43	Carpeta de archivos		

Ahora solo tenemos que entrar en el Visual Studio Code y abrir la carpeta deseada



Listo para su uso



The image shows a screenshot of the Visual Studio Code editor. On the left, the Explorer sidebar shows a project structure with folders 'dist', 'node_modules', and 'src'. Inside 'src', there is a folder 'util' containing 'TS index.ts'. The main editor area displays the content of 'TS index.ts', which is a TypeScript file. The code defines an asynchronous function 'main' that uses 'await' to call 'leerTeclado' (a function from './util/entradaTeclado'). It prompts the user to choose an operation (1: sum, 2: subtract, 3: multiply, 4: divide, 0: exit). It then uses a 'switch' statement to handle the chosen operation, calling 'leerTeclado' again to get numbers and performing the respective arithmetic operation. The bottom panel shows the 'TERMINAL' tab with a Windows PowerShell prompt, displaying copyright information and a URL for PowerShell multiplatform technology.

```
src > TS index.ts > ...
1  import { leerTeclado } from "../util/entradaTeclado";
2  //creamos variable main que le llamamos main
3  let main = async () => {
4  // async significa que la funcion es asincrona lo cual es necesario para el await
5  console.log("1 sumar 2 restar 3 multiplicar 4 dividir 0 salir")
6  let operacion : number
7  let num3: number
8  let num4: number
9      const valor = await leerTeclado('Elija una opcion')
10     operacion = parseInt (valor)
11     while (operacion!=0){
12         switch(operacion){
13             case 1:
14                 const num1 = await leerTeclado('numero1')
15                 const num2 = await leerTeclado('numero2')
16                 num3 = parseInt (num1)
17                 num4 = parseInt (num2)
18                 let A = num3 + num4
19                 console.log(A)
20                 const valor = await leerTeclado('Elija otra opcion')
21                 operacion = parseInt (valor)
22                 break;
23             case 2:
24                 const num5 = await leerTeclado('numero1')
25                 const num6 = await leerTeclado('numero2')
26                 num3 = parseInt (num5)
27                 num4 = parseInt (num6)
28                 let B = num3 - num4
29                 console.log(B)
30                 const valor1 = await leerTeclado('Elija otra opcion')
31                 operacion = parseInt (valor1)
32                 break;
33             case 3:
34                 const num7 = await leerTeclado('numero1')
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

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

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

PS C:\Users\usuario\Desktop\proyectos\GBD\typescript1\2\clonacion\BasedeDatos\7>

Cualquier duda consultar documentación de
Git alojada en este repositorio
<https://github.com/maseri420/eskorbuto.git>