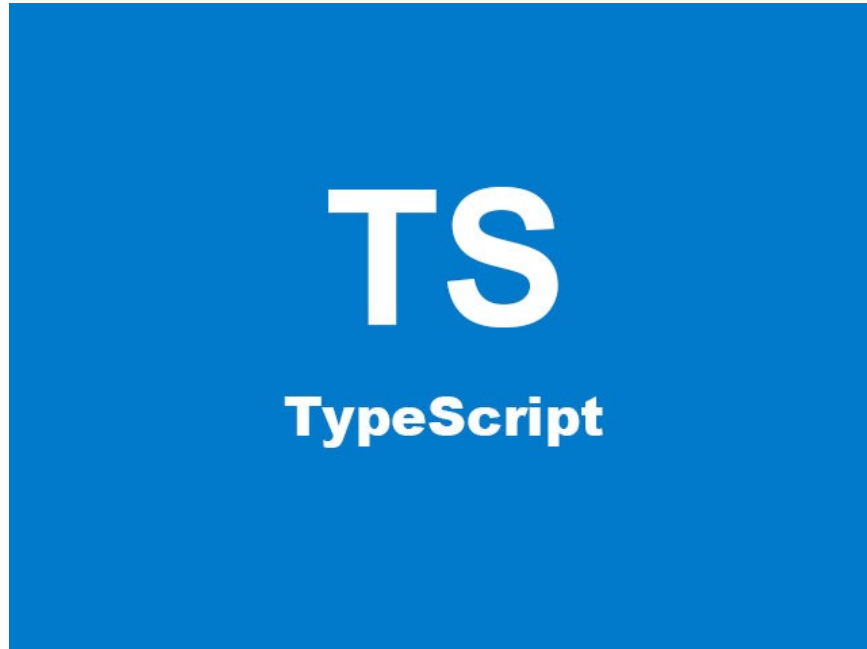


IES Punta Del Verde

TypeScript



Gonzalo Pazos Sardella

Índice.

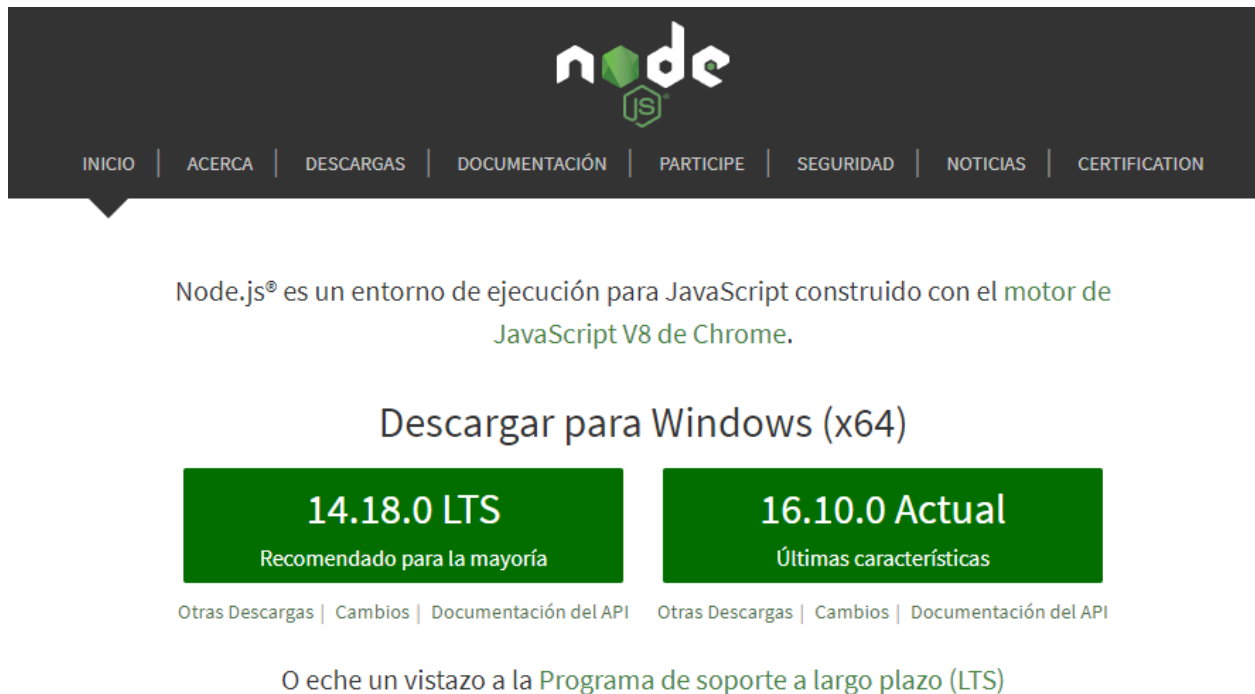
Instalación node js	3
1.1 Descarga node js	3
1.2 Configuración node js	4
1.3 Ver versión en PowerShell	7
2. Instalación TypeScript	7
2.1 Instalación PowerShell	7
3. Instalación Visual Studio Code	8
3.1 Descarga de Visual Studio Code	8
4. Creación proyecto	9
4.1 Primeros pasos	9
4.2 Código de los ejercicios	13
4.3 Git ignore y readme.	15
5. Subida de proyecto a GitHub	17
5.1 Git Clone / Download	17
5.2 Subida desde Visual Studio Code	17
6. Instalación de un proyecto	18
6.1 Git Clone / Download	18

1.Instalación node js

Link: <https://nodejs.org/es/>

1.1 Descarga node js

Nos iremos al link para acceder a la página web, para la instalación de node js. En mi caso descargare la versión 14.18.0 LTS.



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

Descargar para Windows (x64)

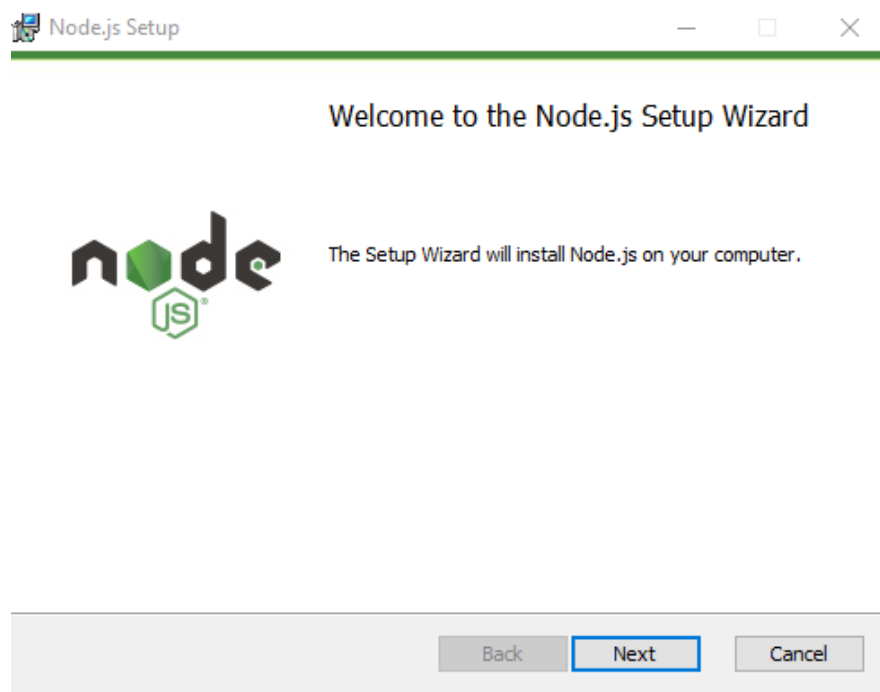
14.18.0 LTS	16.10.0 Actual
Recomendado para la mayoría	Últimas características

Otras Descargas | Cambios | Documentación del API Otras Descargas | Cambios | Documentación del API

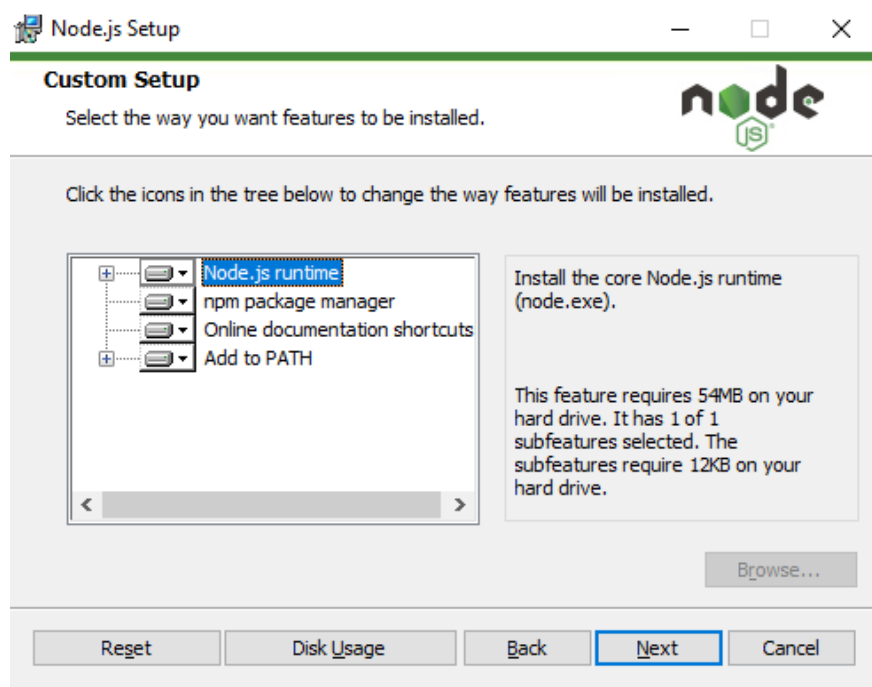
O eche un vistazo a la Programa de soporte a largo plazo (LTS)

1.2 Configuración node js

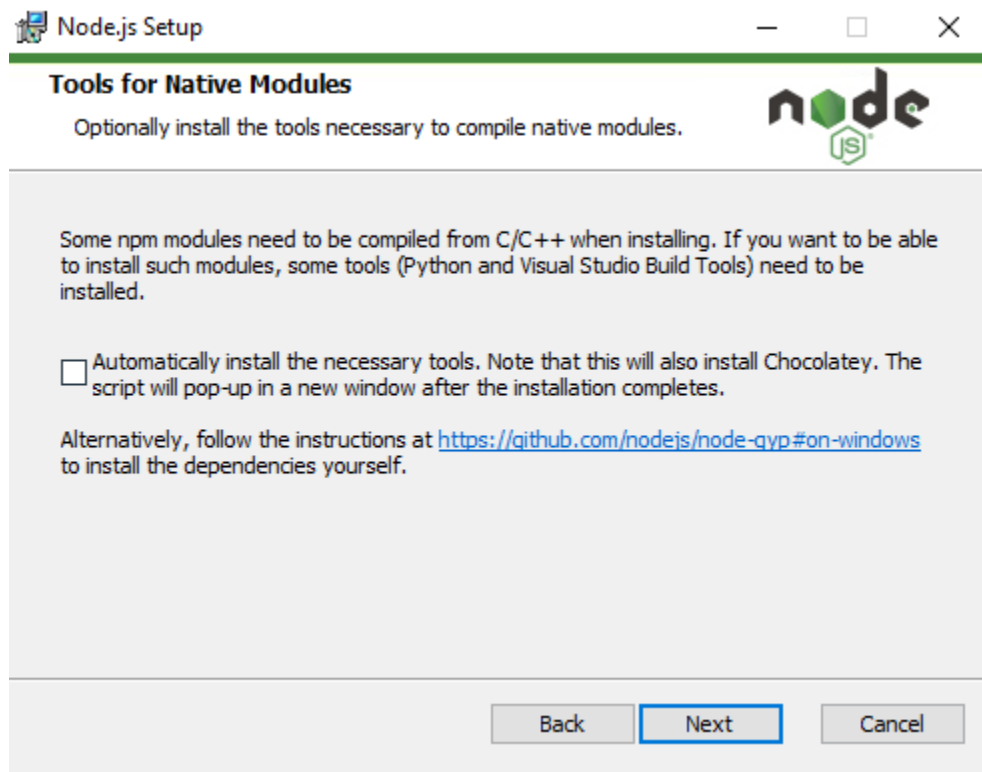
El primer paso es iniciar el proceso de configuración de node js.



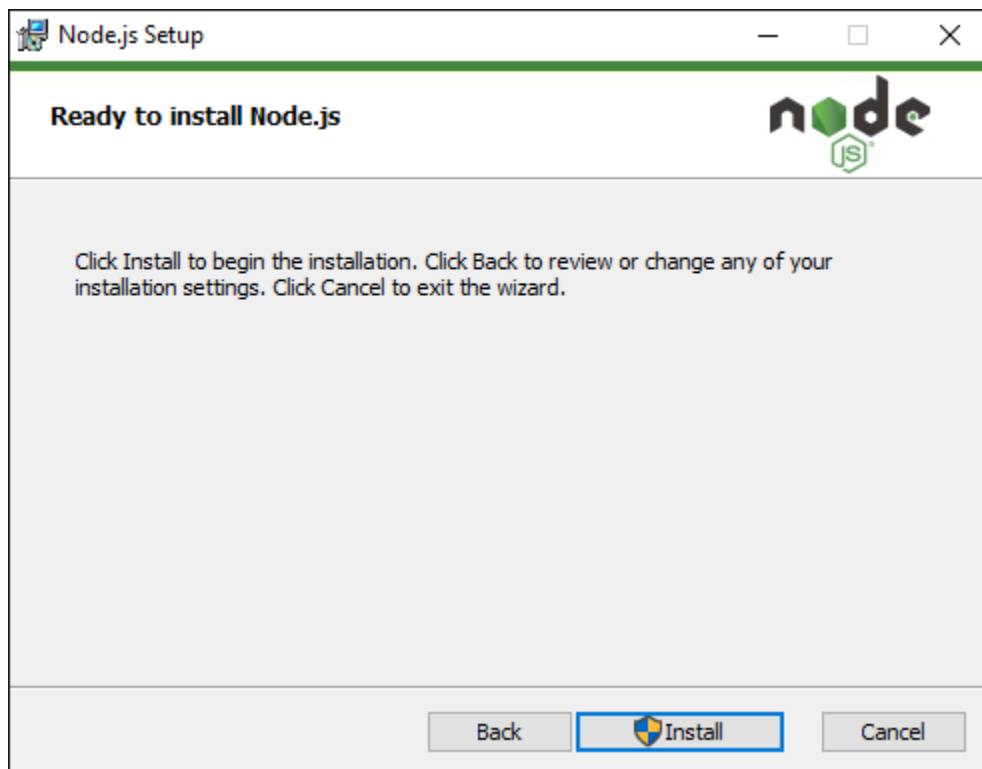
Dejaremos las características predeterminadas de la instalación.



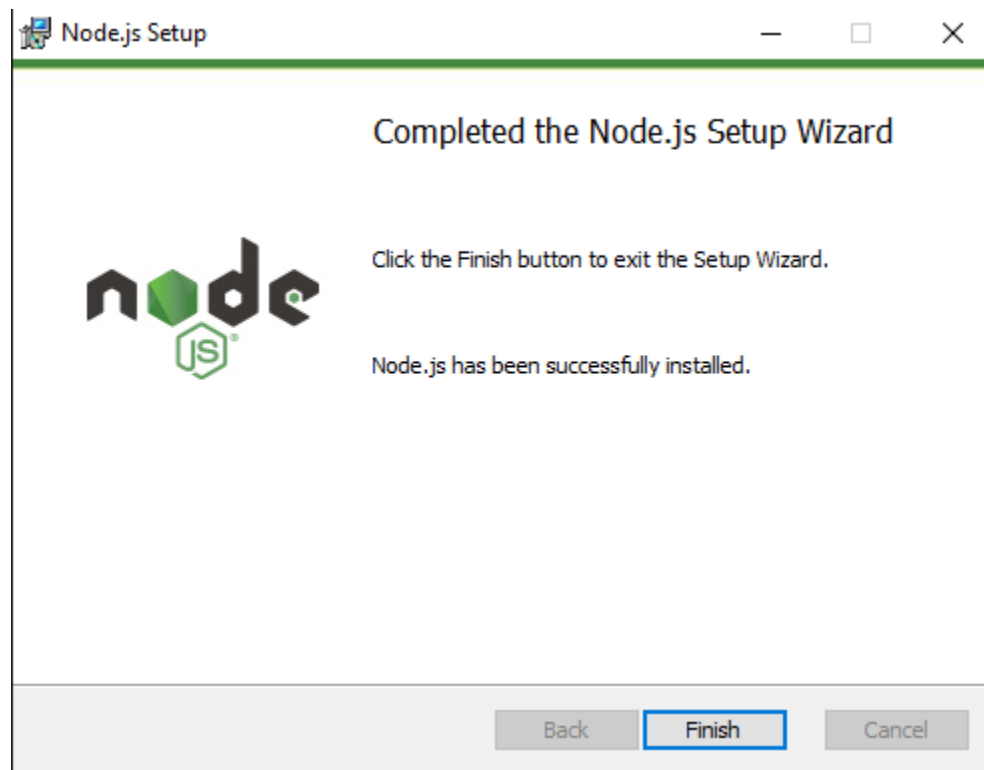
Aquí marcamos la casilla para que nos instale automáticamente las herramientas necesarias.



A continuación le daremos a Install para comenzar la instalación, con todas las configuraciones establecidas.



Una vez se termine de descargar y esta haya sido de forma correcta, le daremos a finish para cerrar la pestaña.



1.3 Ver versión en PowerShell

Para comprobar la versión que tenemos de node, podemos usar en nuestra powershell el comando **node -v**. El cual nos mostrará la versión que tenemos actualmente.

```
PS C:\WINDOWS\system32> node -v
v14.18.0
```

2. Instalación TypeScript

Link para TypeScript: <https://www.typescriptlang.org/>

2.1 Instalación PowerShell

Para instalar JavaScript en nuestro ordenador usaremos PowerShell. Una vez abierto usando el comando **npm install -g typescript**, instalaremos este automáticamente.

```
PS C:\WINDOWS\system32> npm install -g typescript
C:\Users\GONZIPOLIRIS\AppData\Roaming\npm\tsc -> C:\Users\GONZIPOLIRIS\AppData\Roaming\npm\node_modules\typescript\bin\tsc
C:\Users\GONZIPOLIRIS\AppData\Roaming\npm\tsserver -> C:\Users\GONZIPOLIRIS\AppData\Roaming\npm\node_modules\typescript\bin\tsserver
+ typescript@4.4.3
added 1 package from 1 contributor in 5.865s
```

Cuando intentamos comprobar la versión del JavaScript veremos que nos dará un error. Esto es a causa de las políticas de ejecución, para solventar este problema usaremos el comando **set-executionPolicy unrestricted**. Una vez ejecutado este comando nos dirá si lo queremos ejecutar, en nuestro caso pondremos **A**.

```
PS C:\WINDOWS\system32> set-executionPolicy unrestricted

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about_Execution_Policies help topic at https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): o
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A
```

Ya podemos ver la versión de JavaScript sin que nos salte algún error. Esto lo haremos con el comando **tsc -v**.

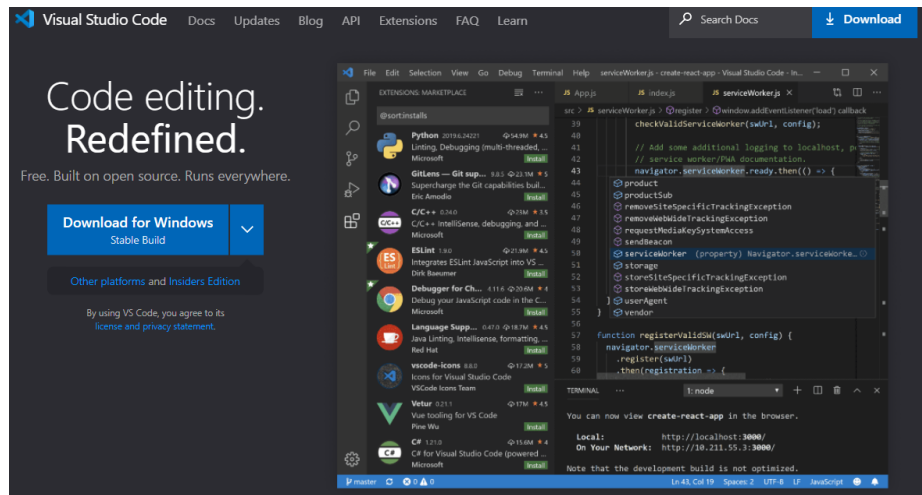
```
PS C:\WINDOWS\system32> tsc -v
Version 4.4.3
```

3. Instalación Visual Studio Code

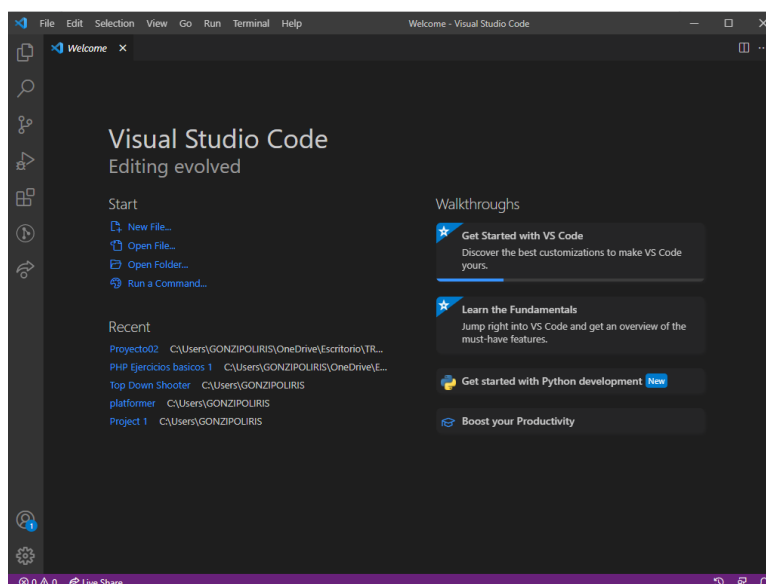
Link: <https://code.visualstudio.com/>

3.1 Descarga de Visual Studio Code

El primer paso es descargarnos el instalador en la página web de Visual Studio Code.



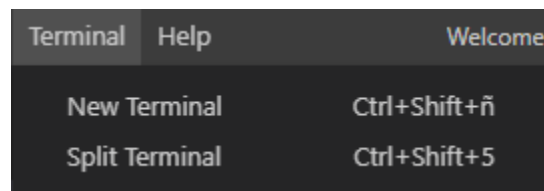
Una vez descargado, ejecutaremos el instalador y dejaremos todo por predeterminado, hasta finalizar la instalación.



4. Creación proyecto

4.1 Primeros pasos

Apenas abrimos Visual Studio, crearemos una carpeta donde tendremos el contenido de nuestro proyecto. A la vez crearemos una terminal, donde ejecutaremos comando necesarios para la creación del proyecto.

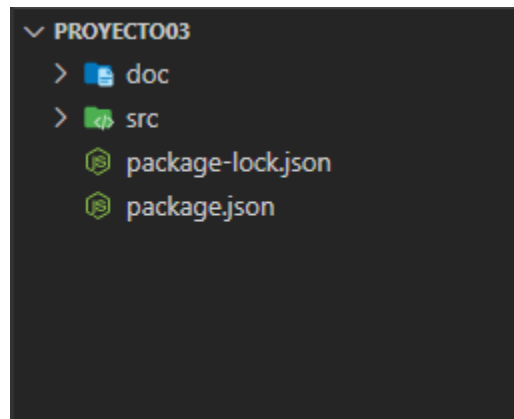


Una vez creada la terminal pondremos el comando **npm init -y**, esto hará que creamos los **package.json**.

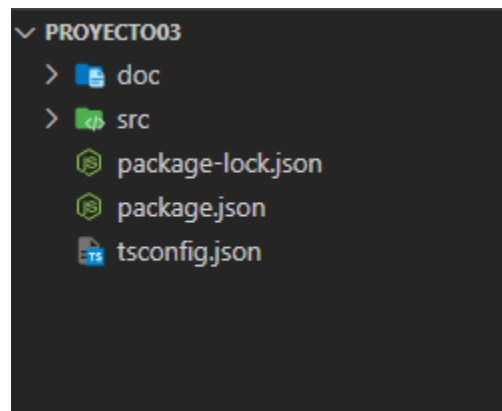
```
PS C:\Users\GONZIPOLIRIS\OneDrive\Escritorio\TRABAJOS PV\SGBD\Proyecto03> npm init -y
Wrote to C:\Users\GONZIPOLIRIS\OneDrive\Escritorio\TRABAJOS PV\SGBD\Proyecto03\package.json:

{
  "name": "Proyecto03",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "directories": {
    "doc": "doc"
  },
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Y en nuestra carpeta veremos los paquetes creados.



Con el comando **tsc --init**, esto nos creará el archivo **tsconfig.json**.



En este haremos modificaciones tales como actualizar la versión y poner la ruta a **./dist**. La target (versión), la cambiaremos de es5 a es6.

```

/* Language and Environment */
"target": "es6", /* Set the Java
// "lib": [], /* Specify a se
// "jsx": "preserve", /* Specify what
// "experimentalDecorators": true, /* Enable exper
// "emitDecoratorMetadata": true, /* Emit design-
// "jsxFactory": "", /* Specify the
// "jsxFragmentFactory": "", /* Specify the
// "jsxImportSource": "", /* Specify modu
// "reactNamespace": "", /* Specify the
// "noLib": true, /* Disable incl
// "useDefineForClassFields": true, /* Emit ECMAScr

```

Y el outDir lo pondremos como **./dist**.

```

/* Emit */
// "declaration": true,
// "declarationMap": true,
// "emitDeclarationOnly": true,
// "sourceMap": true,
// "outFile": "./",
"outDir": "./dist",
// "removeComments": true,
// "noEmit": true,
// "importHelpers": true,
// "importsNotUsedAsValues": "remove",
// "downlevelIteration": true,
// "sourceRoot": "",
// "mapRoot": ""

```

Una vez acabado todo esto pondremos en la terminal **tsc -w**, esto nos compilara los nuevos cambios, y en caso de tener un error avisarnos de dicho error.

```

PS C:\Users\GONZIPOLIRIS\OneDrive\Escritorio\TRABAJOS PV\SGBD\Proyecto03> tsc -w

```

Veremos como automáticamente se compila y nos muestra que tenemos 0 errores ya que no tenemos nada creado, que pueda dar error.

```
[8:54:13 PM] Starting compilation in watch mode...  
[8:54:14 PM] Found 0 errors. Watching for file changes.
```

4.2 Código de los ejercicios

Ejercicio 1:

El primer ejercicio consiste en la generación de números aleatorios a través del método **Math.floor**. Una vez recibido uno de los números aleatorios entre 0 y 100, con la función **If** nos compraba si el número es primo o no.

Funcionamiento: Si el número recibido es divisible entre dos, nos dará un mensaje “El número no es primo”, ya que los números primos son solo divisibles entre uno y su mismo número. Si este no es divisible entre dos a través de un **else** nos dará el mensaje “El número es primo”.

```
// Ejercicio 1

let a = Math.floor(Math.random() * (100)); // Declaramos $a a una funcion que nos permite generar numeros desde el 0 al 100
console.log("Numeros primos\n")

if(a%2==0) { // Hacemos un if para comprobar si el numero es primo o no
  console.log(`El número ${a} no es primo`);
}
else {
  console.log(`El número ${a} es primo`);
}
```

Ejercicio 2:

En este a través de la creación de una lista compuesta por números, con el método **sort()**, éste podrá ordenar la lista. El problema es que **sort** no funciona con números ya que los ordena pero de posicionamiento en la memoria. Para poder hacer que lo ordene usaremos una función.

Funcionamiento: Primero creamos una lista con sus respectivos valores. A la variable le aplicamos el método **sort** y a esta le agregamos una función que cogerá los valores de la lista ordenandolas de maneras ascendente con **a - b**, ya que está declarada que la izquierda es **a** y la derecha es **b**. La siguiente de menor a mayor cambiando las el orden con **b - a** es decir de derecha a izquierda.

```
// Ejercicio 2

console.log("\nOrdenar de mayor a menor una lista\n")

const numeros = [14, 4, 8 ,1]; // Creamos una lista con numeros
console.log(`La lista es: ${numeros}\n`)

console.log("Ordenada de mayor a menor:\n")
console.log(numeros.sort(function(a, b){return a - b})); // Sort nos permite ordenar, creando la función podemos ordenandolo de negativo a 0 a positivo
console.log("\nOrdenada de menor a mayor:\n")
console.log(numeros.sort(function(a, b){return b - a}));
```

Ejercicio 3.

Como último ejercicio, he hecho una función que cuenta las palabras que hay en un texto.

Funcionamiento: Primero he declarado una variable que contiene el texto. Nuevamente una función llamada **wordCount**, cuyo funcionamiento va a ser el de contar las palabras. Dentro de esta dentro del return meteremos el método **text.split**, este cumple el funcionamiento de quitarnos los espacios que separan el texto. Con el método **.length** cuenta cada palabra, restando uno a la función hacemos que esos espacios sean restados de la cuenta. Por último printeamos en pantalla el resultado.

```
// Ejercicio 3

console.log("Contar palabras en frase\n")

var frase = `¡Oh, memoria, enemiga mortal de mi descanso!`.
La virtud más es perseguida de los malos que amada de los buenos."
La ingratitude es hija de la soberbia."
La razón de la sinrazón que a mi razón se hace,
de tal manera mi razón enflaquece, que con razón me quejo de la vuestra fermosura.`;

function wordCount(text = '') { // Cuenta cada palabra
  return text.split(/\s+/).length - 1; // A cada espacio le resta ese espacio, para que estos no sean contados
};

console.log(`Texto: ${frase}\n`);
console.log(`La cantidad de palabras son: ${wordCount(frase)}`);
```

4.3 Git ignore y readme.

Por último agregamos los archivos **.gitignore** en el cual meteremos **dist** y **node_modules**.

```
.gitignore
1 dist
2 node_modules
```

Y crearemos también un readme con la información que queramos proporcionarle a los usuarios.

```
1 # Proyecto nodejs
2 # Proyecto Gonzalo Pazos Sardella
3 ## Ejercicios: Numeros primos, ordenar lista mayor a menor y menor a mayor y contar palabras en frase.
4
5 Contiene:
6   - doc : Contenido documentado en PDF.
7   - src : Carpeta con el codigo de los ejercicios
8   - .gitignore
9   - README.md información del proyecto
```

5. Subida de proyecto a GitHub

5.1 Git Clone / Download

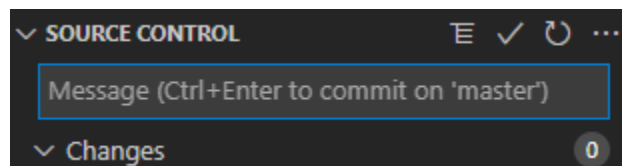
Para la subida he clonado el repositorio de GitHub para obtener una carpeta vacía donde copiar el contenido de mi carpeta de trabajo ahí.

```
PS C:\Users\GONZIPOLIRIS\OneDrive\Escritorio\TRABAJO PV\SGBD> git clone https://github.com/pazossardellagonzalo/Trabajo-TypeScript-GonzaloPazosSardella.git
Cloning into 'Trabajo-TypeScript-GonzaloPazosSardella'...
warning: You appear to have cloned an empty repository.
```









5.2 Subida desde Visual Studio Code

Una vez hecho eso desde **Visual Studio** en el apartado de **Source Control** añadiremos el contenido de toda la carpeta con **git add ..**

Una vez agregado todo el contenido haremos un commit. En **message** agregaremos un comentario para nuestro commit, una vez escrito le damos al tick para hacer el commit.



Para hacer el push le daremos a los tres puntitos, se nos abrirá una lista donde le daremos al pull, este nos subirá automáticamente a nuestro GitHub todo.

	pazossardellagonzalo nuevo cambio	0540bc7 14 minutes ago	 2 commits
	src	nuevo cambio	14 minutes ago
	.gitignore	ok	18 minutes ago
	README.md	ok	18 minutes ago
	package-lock.json	ok	18 minutes ago
	package.json	ok	18 minutes ago
	tsconfig.json	ok	18 minutes ago

6. Instalación de un proyecto

6.1 Git Clone / Download

Para obtener el archivo con todo su contenido podemos usar tanto un git clone referenciando el link del repositorio, o desde el mismo repositorio descargarlo.

Una vez lo tengamos, abriremos la carpeta en nuestro Visual Studio y crearemos una terminal. En esta usaremos el comando **npm install**. Nuevamente usaremos **npm install typescript -D**, para descargar typescript con dependencia a ese proyecto. A continuación cambiaremos la configuración del **tsconfig.json** si esta no lo esta. Por último en una nueva terminal pondremos **tsc -w**, para compilar el proyecto y saber si hay algún error por falta de cosas instaladas o de código.