

Primer Proyecto TypeScript

Antonio Ávila Delgado

2º ASIR A

ÍNDICE

- Introducción.....pág 3.
- Pasos.....pág 3

INTRODUCCIÓN

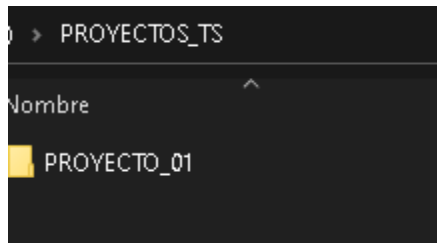
En este documento vamos a explicar, paso a paso, un caso práctico de uso de TypeScript

PASOS

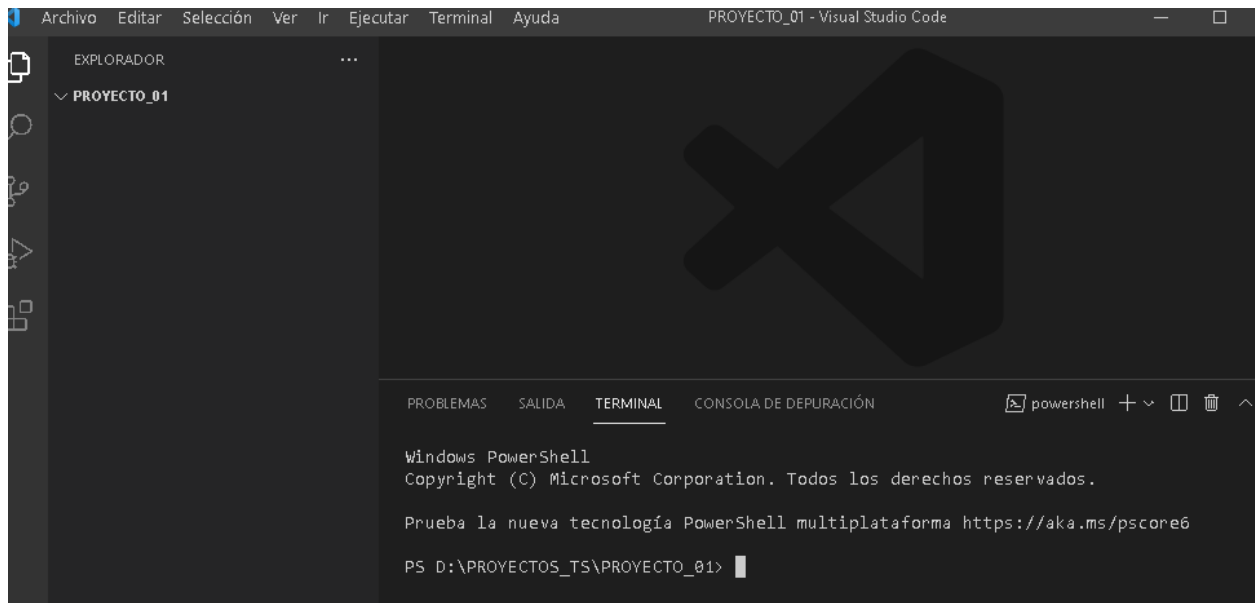
En primer lugar instalamos typescript a nivel global con: **npm install -g typescript**

```
PS C:\Users\Usuario> npm install -g typescript
C:\Users\Usuario\AppData\Roaming\npm\tsc -> C:\Users\Usuario\AppData\Roaming\npm\node_modules\typescript\bin\tsc
C:\Users\Usuario\AppData\Roaming\npm\tsserver -> C:\Users\Usuario\AppData\Roaming\npm\node_modules\typescript\bin\tsserver
+ typescript@4.4.3
updated 1 package in 3.943s
PS C:\Users\Usuario>
```

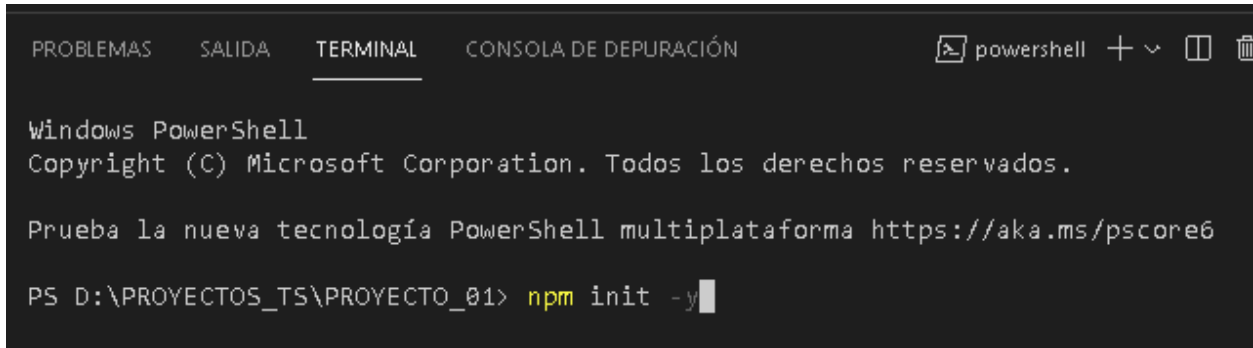
Creamos una carpeta de proyecto



La carpeta del proyecto y se arrastra a una nueva ventana de VSC.



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



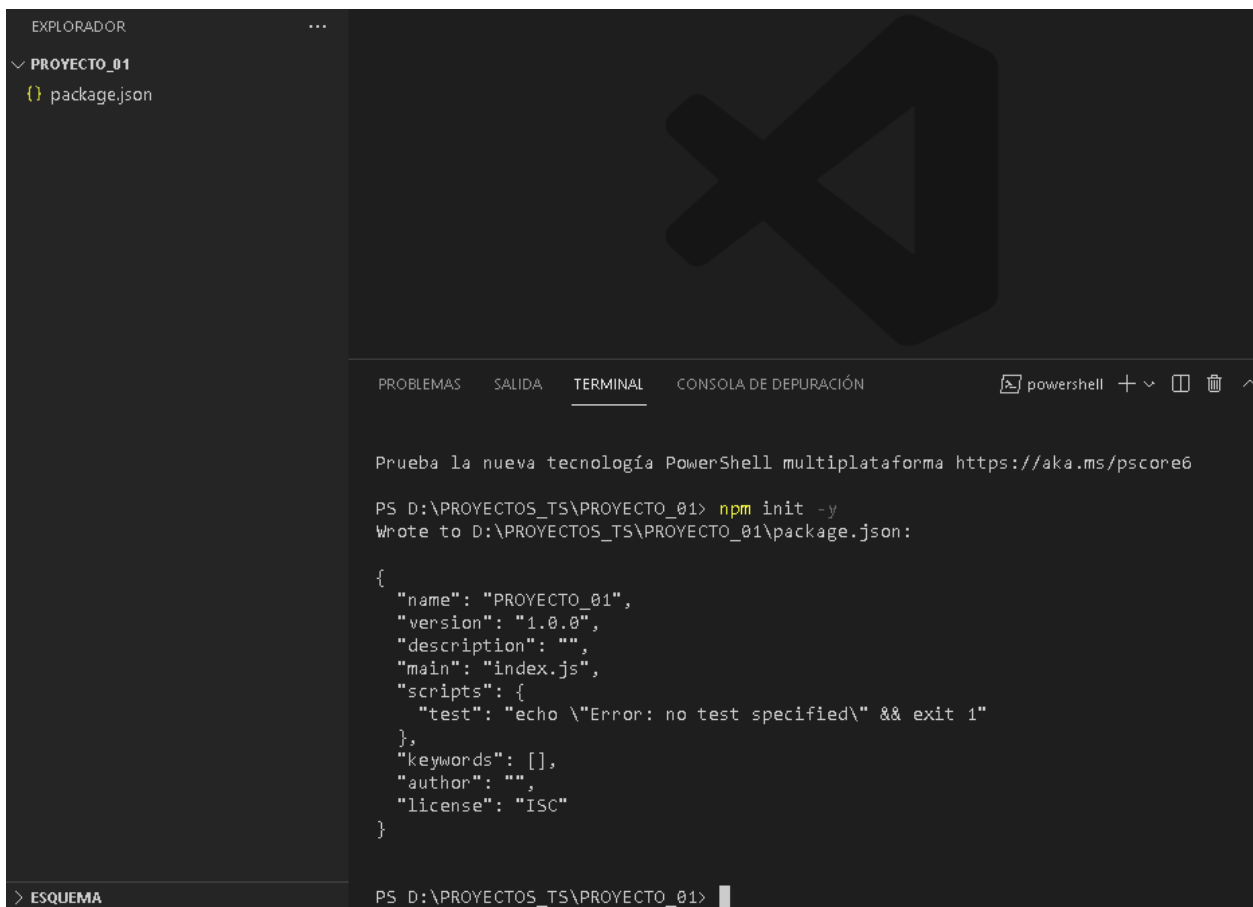
```
PROBLEMAS  SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN  powershell + v [icon] [icon]

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

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

PS D:\PROYECTOS_TS\PROYECTO_01> npm init -y
```

Y nos sale esto:



```
EXPLORADOR  ...
PROYECTO_01
  {} package.json

PROBLEMAS  SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN  powershell + v [icon] [icon]

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

PS D:\PROYECTOS_TS\PROYECTO_01> npm init -y
Wrote to D:\PROYECTOS_TS\PROYECTO_01\package.json:

{
  "name": "PROYECTO_01",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

> ESQUEMA  PS D:\PROYECTOS_TS\PROYECTO_01>
```

Es decir, crea el `package.json`

Ahora escribimos en la terminal `tsc --init`



```
PS D:\PROYECTOS_TS\PROYECTO_01> tsc --init
```

Y nos sale esto:

```
PS D:\PROYECTOS_TS\PROYECTO_01> tsc --init
message TS6071: Successfully created a tsconfig.json file.
PS D:\PROYECTOS_TS\PROYECTO_01>
```

Es decir, crea el archivo tsconfig.json

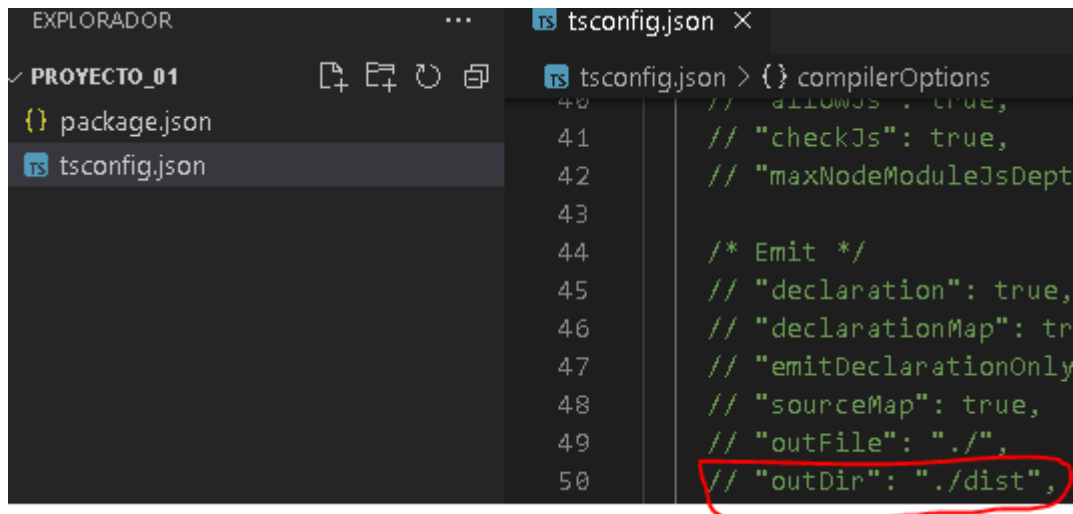
Vamos a dicho archivo para ver el contenido

```
1 {
2   "compilerOptions": {
3     /** Visit https://aka.ms/tsconfig.json to read more about this file */
4
5     /** Projects */
6     // "incremental": true,           /* Enable incremental compilation */
7     // "composite": true,            /* Enable constraints that allow a TypeScript proje
8     // "tsBuildInfoFile": "./",      /* Specify the folder for .tsbuildinfo incremental
9     // "disableSourceOfProjectReferenceRedirect": true, /* Disable preferring source files instead of decla
10    // "disableSolutionSearching": true, /* Opt a project out of multi-project reference che
11    // "disableReferencedProjectLoad": true, /* Reduce the number of projects loaded automaticall
12
13    /** Language and Environment */
14    "target": "es5",                 /* Set the JavaScript language version for emitted
15    // "lib": [],                    /* Specify a set of bundled library declaration fil
16    // "jsx": "preserve",            /* Specify what JSX code is generated. */
17    // "experimentalDecorators": true, /* Enable experimental support for TC39 stage 2 dra
18    // "emitDecoratorMetadata": true, /* Emit design-type metadata for decorated declarat
19    // "jsxFactory": "",              /* Specify the JSX factory function used when targe
20    // "jsxFragmentFactory": "",      /* Specify the JSX Fragment reference used for frag
21    // "jsxImportSource": "",        /* Specify module specifier used to import the JSX
22    // "reactNamespace": "",         /* Specify the object invoked for 'createElement'.
23    // "noLib": true,                /* Disable including any library files, including t
24    // "useDefineForClassFields": true, /* Emit ECMAScript-standard-compliant class fields.
```

Cambiamos "target:" a "es6"

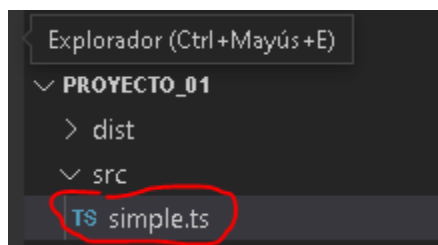
```
1 {
2   "compilerOptions": {
3     /** Visit https://aka.ms/tsconfig.json to read more about this file */
4
5     /** Projects */
6     // "incremental"
7     // "composite":
8     // "tsBuildInfoF
9     // "disableSourc
10    // "disableSolut
11    // "disableRefer
12
13    /** Language and
14    "target": "es6",
```

Cambiamos "outDir": "./dist",



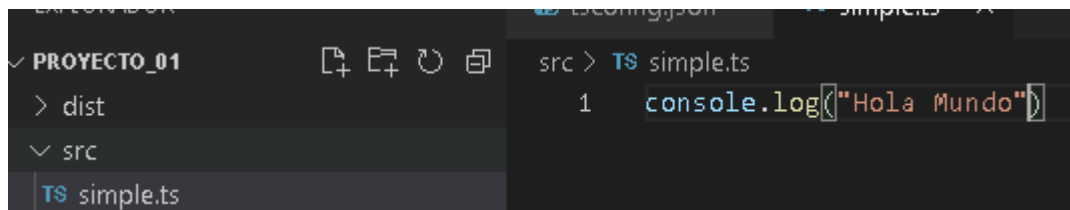
```
tsconfig.json
40 // "allowJs": true,
41 // "checkJs": true,
42 // "maxNodeModuleJsDepth": 1,
43
44 /* Emit */
45 // "declaration": true,
46 // "declarationMap": true,
47 // "emitDeclarationOnly": true,
48 // "sourceMap": true,
49 // "outFile": "./",
50 // "outDir": "./dist",
```

Creamos una carpeta src y un primer archivo por el ejemplo con el nombre: simple.ts



```
Explorador (Ctrl+Mayús+E)
PROYECTO_01
  dist
  src
  TS simple.ts
```

Y dicho archivo tendrá el contenido console.log("Hola Mundo")



```
src > TS simple.ts
1 console.log("Hola Mundo")
```

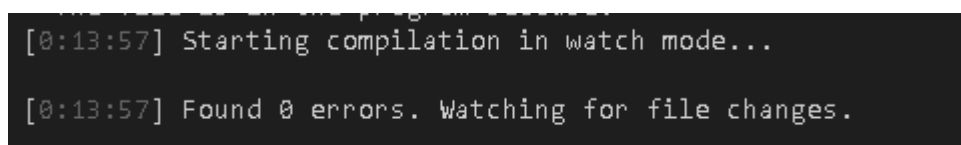
Ahora se puede tener dos terminales abiertos.

En uno escribimos: tsc -w



```
PROBLEMAS SALIDA TERMINAL CONSOLA DE DEPURACIÓN
PS D:\PROYECTOS_TS\PROYECTO_01> tsc -w
```

Y nos sale esto:



```
[0:13:57] Starting compilation in watch mode...
[0:13:57] Found 0 errors. Watching for file changes.
```

En el otro terminal invocamos el ejecutable de JavaScript que se encuentra en la carpeta dist:
>node dist/simple.js

```
PS D:\PROYECTOS_TS\PROYECTO_01> node dist/simple.js
Hola Mundo
PS D:\PROYECTOS_TS\PROYECTO_01> █
```

Con el package.json creado quedaría:

```
PS D:\PROYECTOS_TS\PROYECTO_01> node dist/index.js
Hola Mundo
PS D:\PROYECTOS_TS\PROYECTO_01> █
```