

PRIMER PROYECTO GITHUB

1º Instalación

Lo primero que debemos hacer es asegurarnos de que tenemos todas las herramientas necesarias. Para ello debemos instalar los siguientes programas utilizando la configuración por defecto:

-Visual Studio Code (Donde manipularemos el código):

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

-Node.js (Para ejecutar y trabajar con JavaScript):

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

-Git (Para crear nuestro repositorio en GitHub):

<https://git-scm.com/>

-TypeScript (Para crear nuestro código TypeScript): para instalarlo, previa instalación de Node.js , debemos introducir el comando `npm install -g typescript` en nuestra consola.

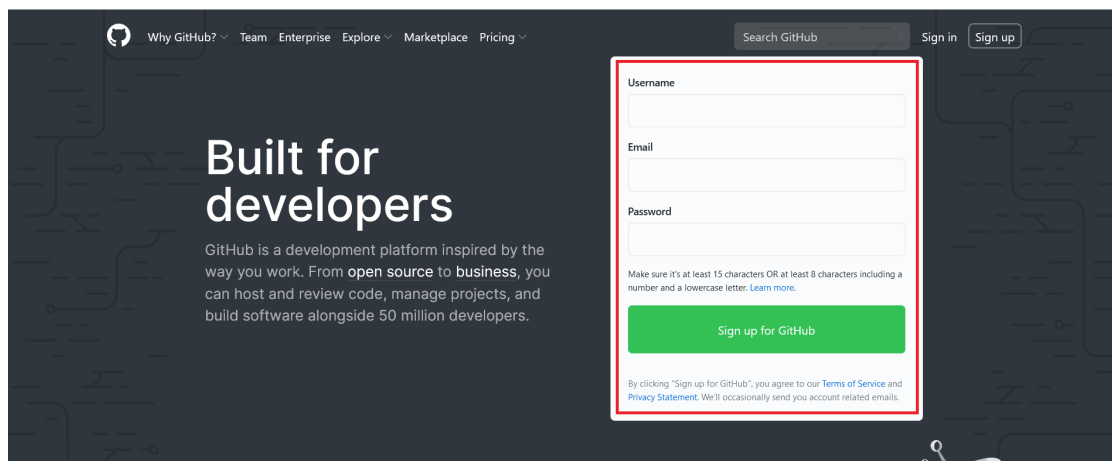
<https://www.npmjs.com/package/typescript>

Es posible que nos de un error en la instalación porque no tengamos activada la ejecución de scripts en nuestro equipo, para activarla debemos introducir el comando `Set-ExecutionPolicy Unrestricted`

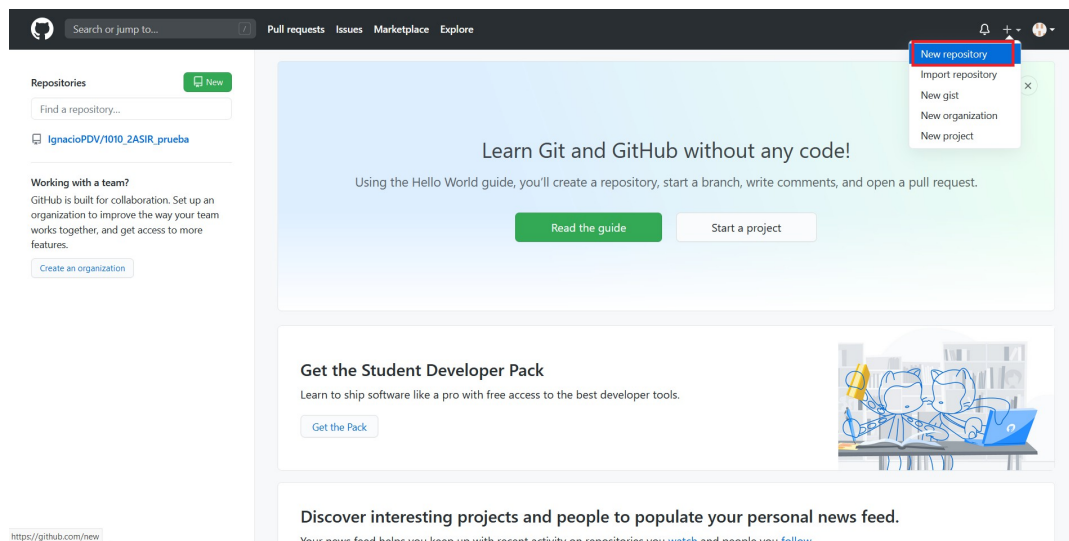
2º Creación de proyecto

Prevía creación de proyecto debemos crearnos una cuenta en github

<https://github.com/>



Una vez creada nuestra cuenta y verificada debemos crear nuestro proyecto



Le daremos un nombre, una posible descripción, la visibilidad del proyecto (si es público o privado) y si queremos añadir un archivo README, gitignore o elegir una licencia

Create a new repository

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

Owner *

Repository name *

IgnacioPDV

/

i200929_PrimerosPasosTypeScript

Great repository names are short and memorable. Need inspiration? How about [expert-festa?](#)

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.](#)

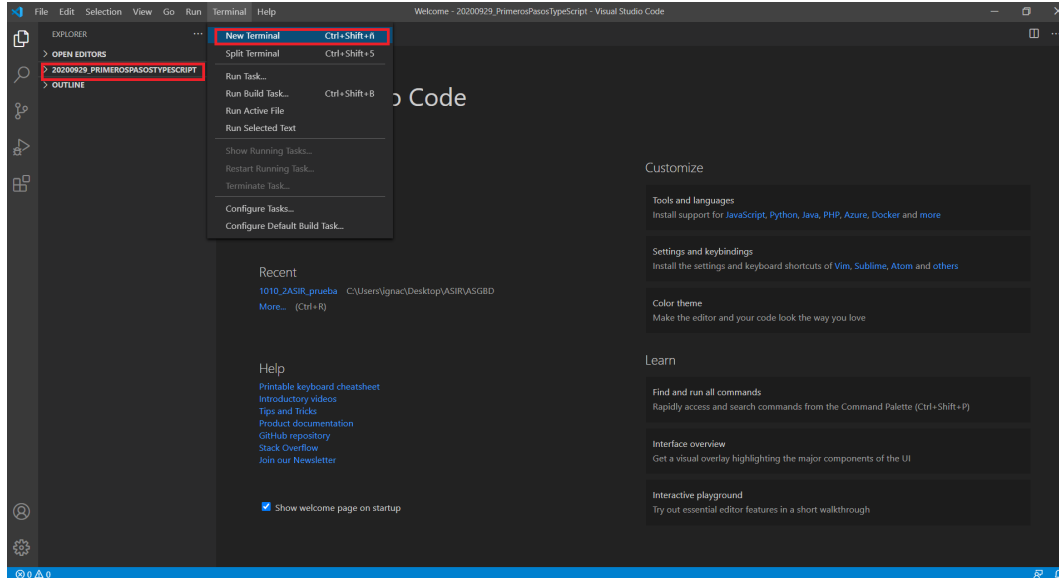
☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

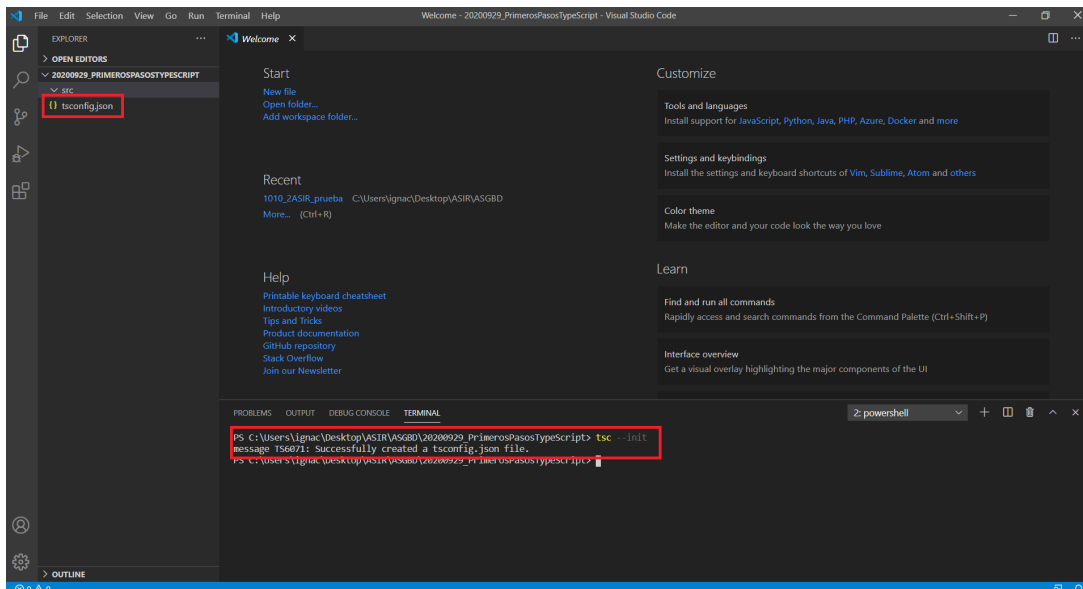
Create repository

3º Empezar a trabajar

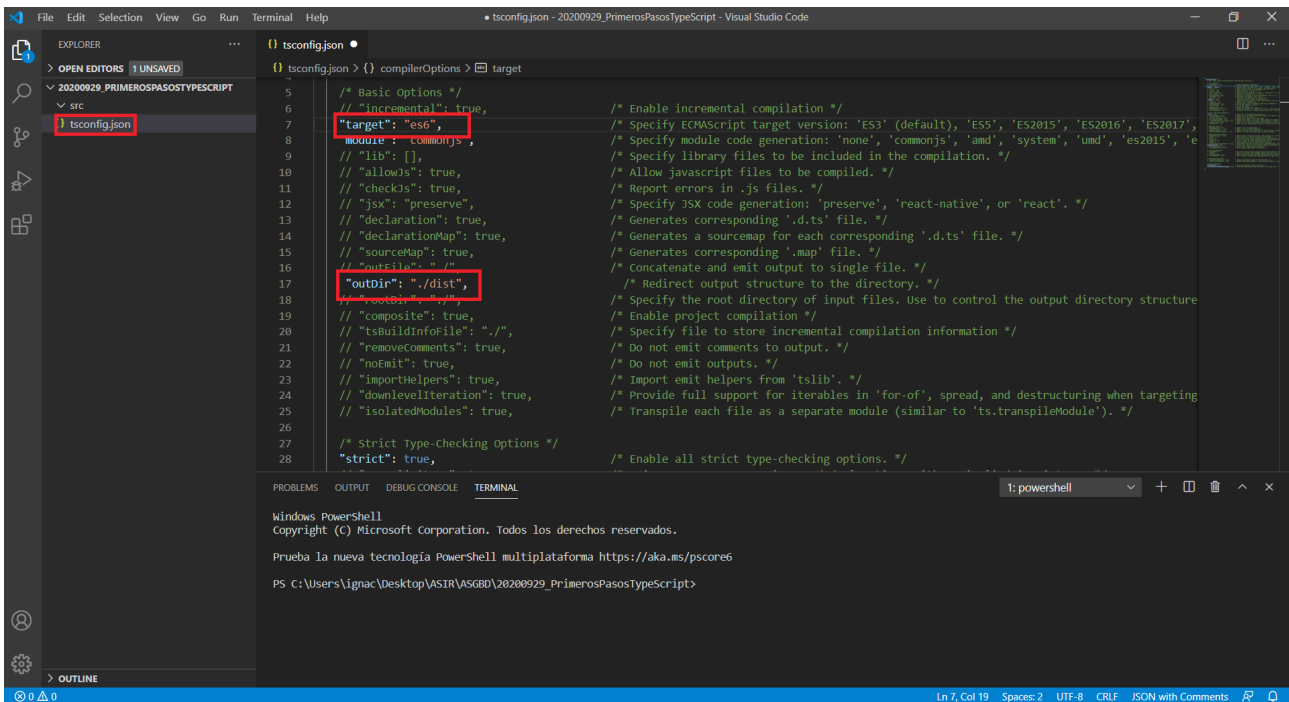
Para empezar a trabajar crearemos una carpeta (con el mismo nombre que le dimos al proyecto para facilitar el trabajo) y la abriremos desde Visual Studio Code. Dentro de esta carpeta crearemos otra llamada src y abriremos una terminal.



Ahora iniciaremos nuestro proyecto usando el comando `tsc --init` que nos generará un archivo llamado `tsconfig.json`, donde está la configuración de nuestro proyecto.



Accederemos al archivo de configuración y descomentaremos la opción “outDir” para que nos cree un directorio donde se compilará nuestro código TypeScript y le daremos de nombre dist. También cambiaremos la versión de JavaScript a “es6”.



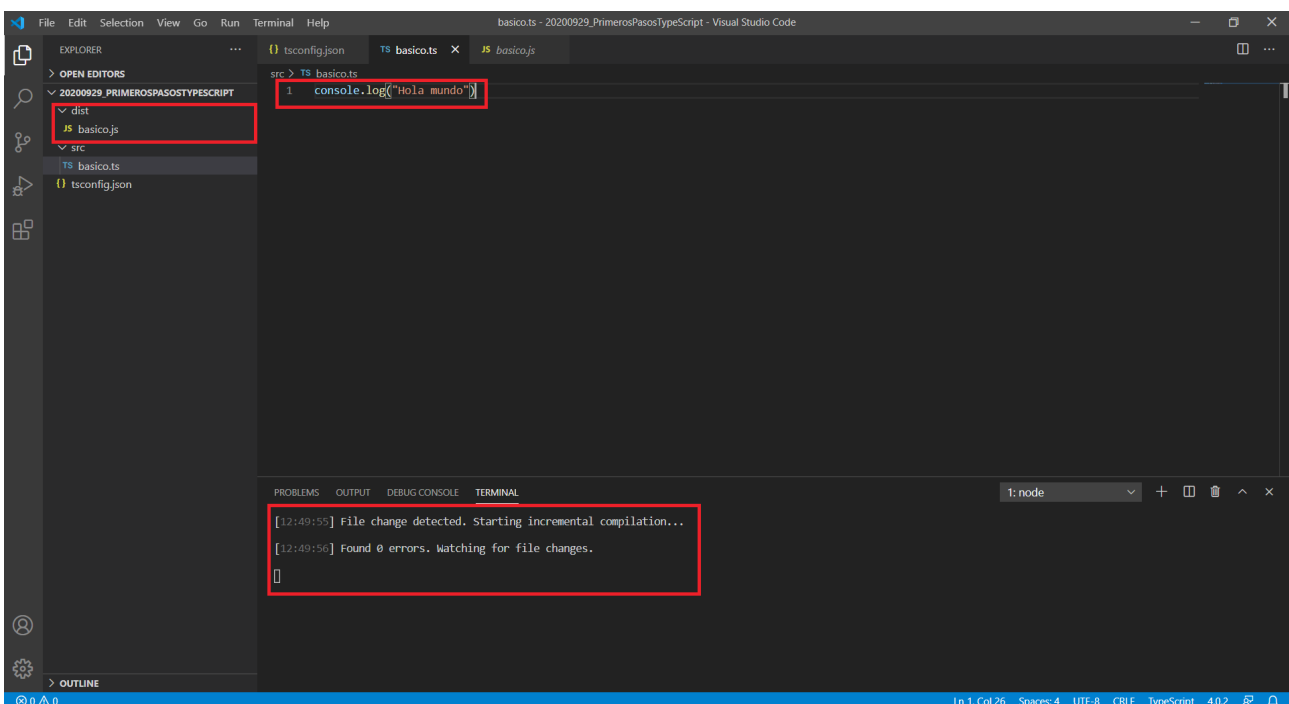
```
tsconfig.json
{
  "compilerOptions": {
    "target": "es6",
    "module": "commonjs",
    "lib": [],
    "allowJs": true,
    "checkJs": true,
    "jsx": "preserve",
    "declaration": true,
    "sourceMap": true,
    "outDir": "./dist",
    "strict": true
  }
}
```

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

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

PS C:\Users\lignac\Desktop\ASIR\ASG80\20200929_PrimerosPasosTypeScript>

Crearemos un archivo básico en TypeScript y en la consola ejecutamos el comando `tsc -w` para que cada vez que guardemos se compile automáticamente y se cree el archivo js. Y ya podemos empezar a hacer nuestro código.



```
basic.ts
1 console.log("Hola mundo");
```

dist
basic.js

terminal

```
[12:49:55] File change detected. Starting incremental compilation...
[12:49:56] Found 0 errors. Watching for file changes.
```

4º Sincronizar nuestro repositorio

Una vez queramos sincronizar nuestro repositorio nos posicionaremos desde la consola en nuestra carpeta del proyecto y haremos lo siguiente:

-Lo primero que haremos será crear un archivo llamado *.gitignore* que sirve para no incluir lo que elijamos, dentro escribiremos *dist*.

Ahora usaremos estos comandos:

-*git init* (para iniciar git)

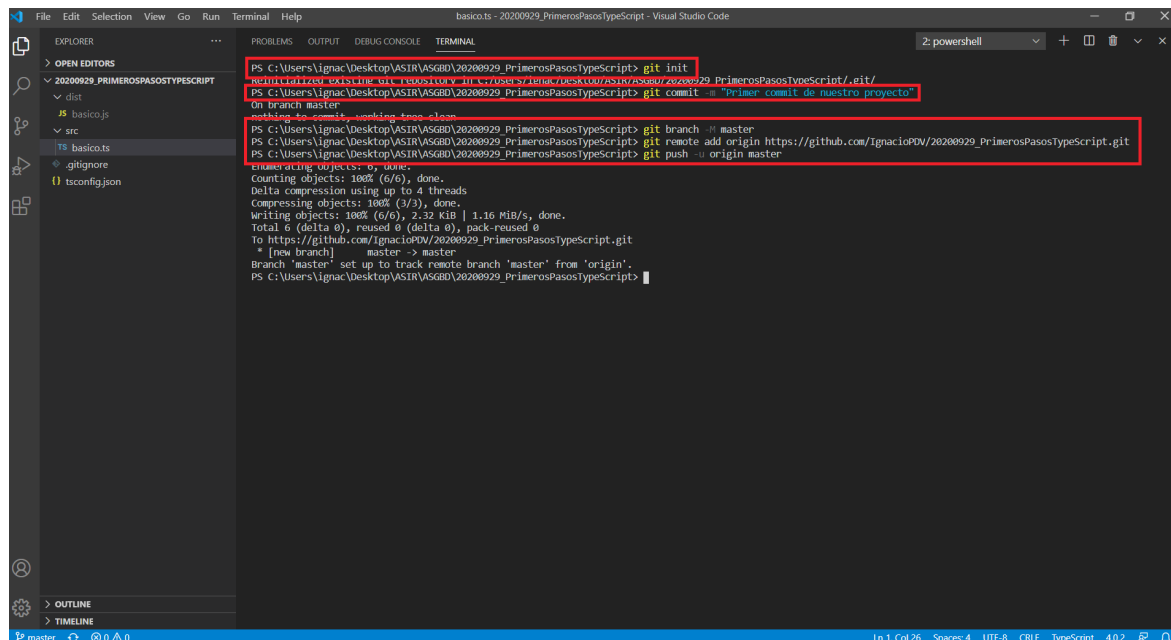
-*git commit -m ""* (hacemos el commit y ponemos un mensaje)

-*git branch -M master* (declaramos el master)

-*git remote add origin*

https://github.com/IgnacioPDV/20200929_PrimerosPasosTypeScript.git (declaramos el origen)

-*git push -u origin master* (y por último subimos el contenido)



```
PS C:\Users\Ignacio\Desktop\ASIR\ASIGBD\20200929_PrimerosPasosTypeScript> git init
Initialized empty Git repository in C:\Users\Ignacio\Desktop\ASIR\ASIGBD\20200929_PrimerosPasosTypeScript\.git\
PS C:\Users\Ignacio\Desktop\ASIR\ASIGBD\20200929_PrimerosPasosTypeScript> git commit -m "Primer commit de nuestro proyecto"
On branch master
nothing to commit, working tree clean
PS C:\Users\Ignacio\Desktop\ASIR\ASIGBD\20200929_PrimerosPasosTypeScript> git branch -M master
PS C:\Users\Ignacio\Desktop\ASIR\ASIGBD\20200929_PrimerosPasosTypeScript> git remote add origin https://github.com/IgnacioPDV/20200929_PrimerosPasosTypeScript.git
PS C:\Users\Ignacio\Desktop\ASIR\ASIGBD\20200929_PrimerosPasosTypeScript> git push -u origin master
Enumerating objects: 0, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/3), done.
Writing objects: 100% (6/6), 2.32 KiB | 1.16 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/IgnacioPDV/20200929_PrimerosPasosTypeScript.git
 * [new branch] master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
PS C:\Users\Ignacio\Desktop\ASIR\ASIGBD\20200929_PrimerosPasosTypeScript>
```

Las siguientes veces que queramos hacer un push usaremos los siguientes comando desde la carpeta de nuestro proyecto:

-*git add .* (el punto para añadir todo)

-*git commit -m ""*

-*git push -u origin master*