



# PRIMER PROYECTO DE GIT HUB

DANIEL MUÑOZ RODRÍGUEZ

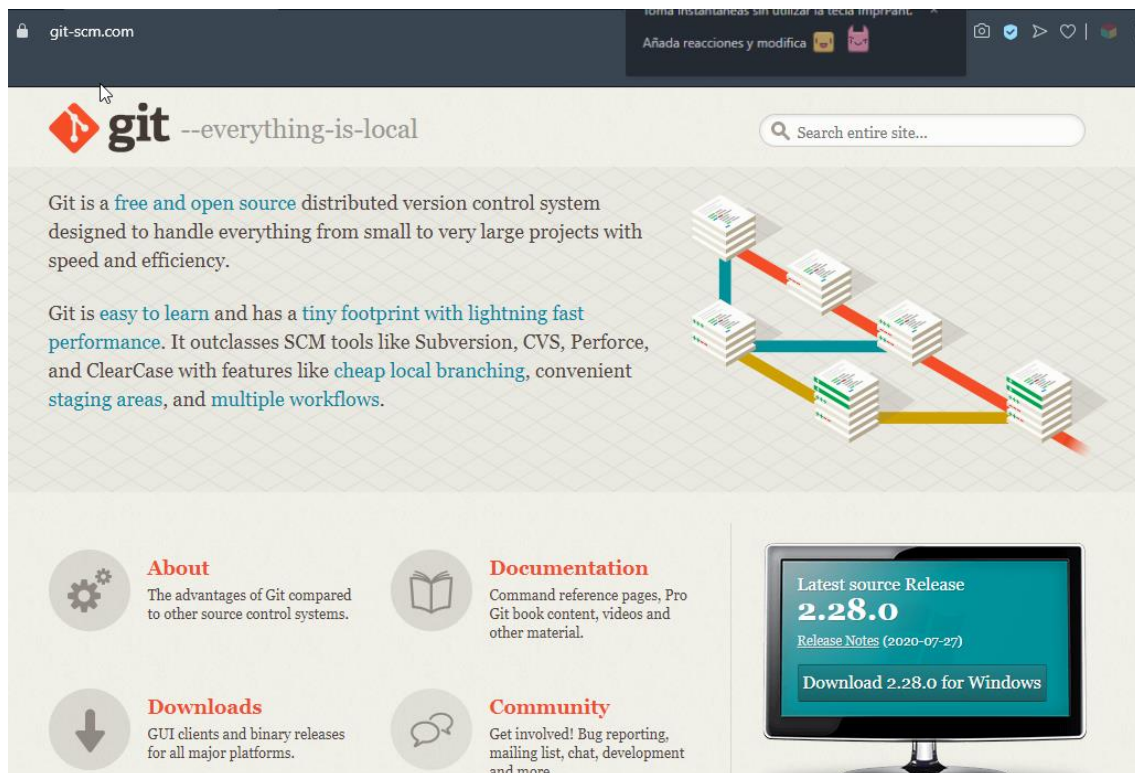


## Contenido

1.	Instalación de programas .....	2
2.	Creamos el repositorio .....	10
3.	Creamos el proyecto .....	11
4.	Sincronización del trabajo con el repositorio .....	13

## 1. Instalación de programas

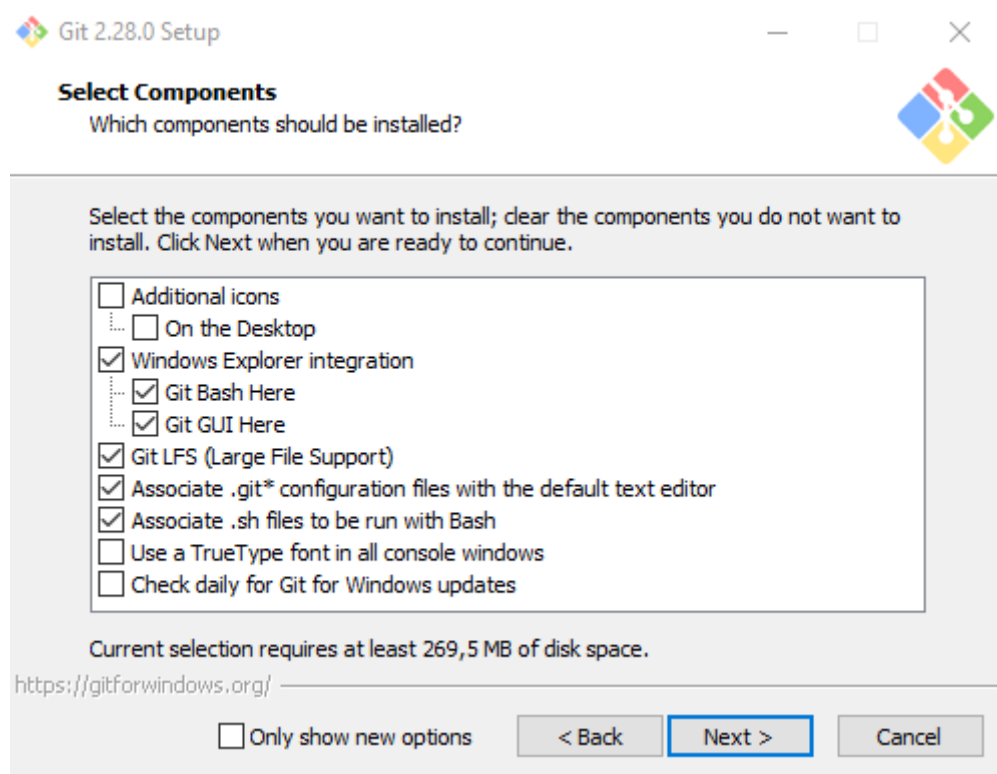
Lo primero que debemos de hacer es instalar git, para ello nos vamos a la siguiente página web: [enlace](https://git-scm.com) . Y descargamos el instalador de git.



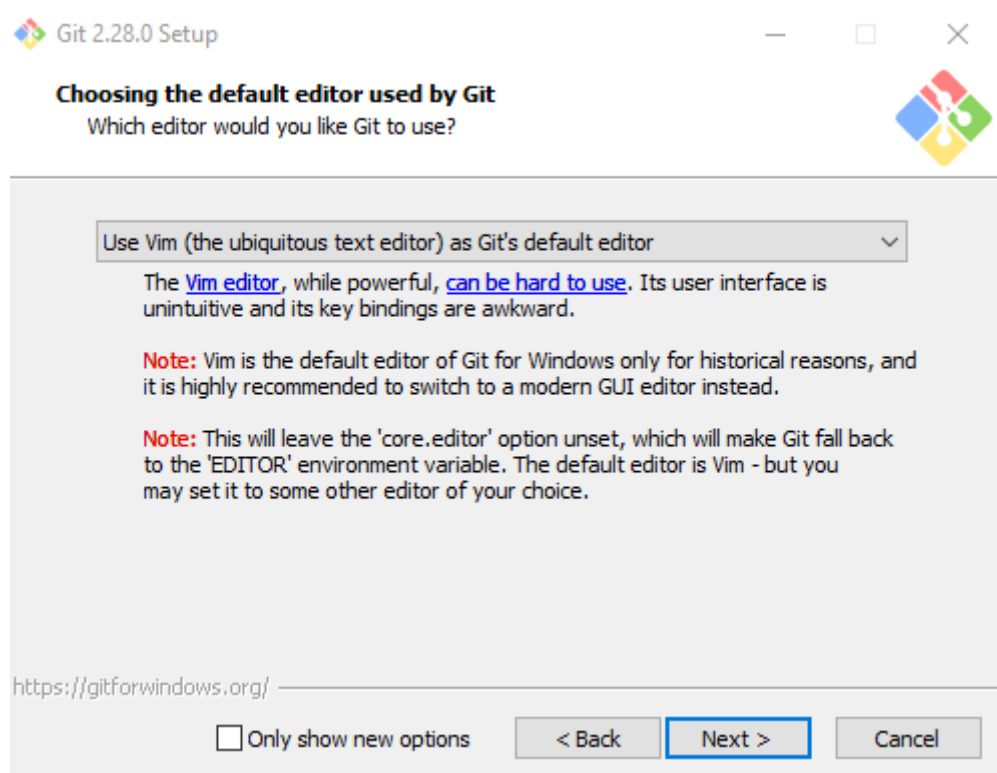
Una vez que lo tengamos descargado procedemos a instalarlo. Abrimos el instalador y le damos a siguiente.



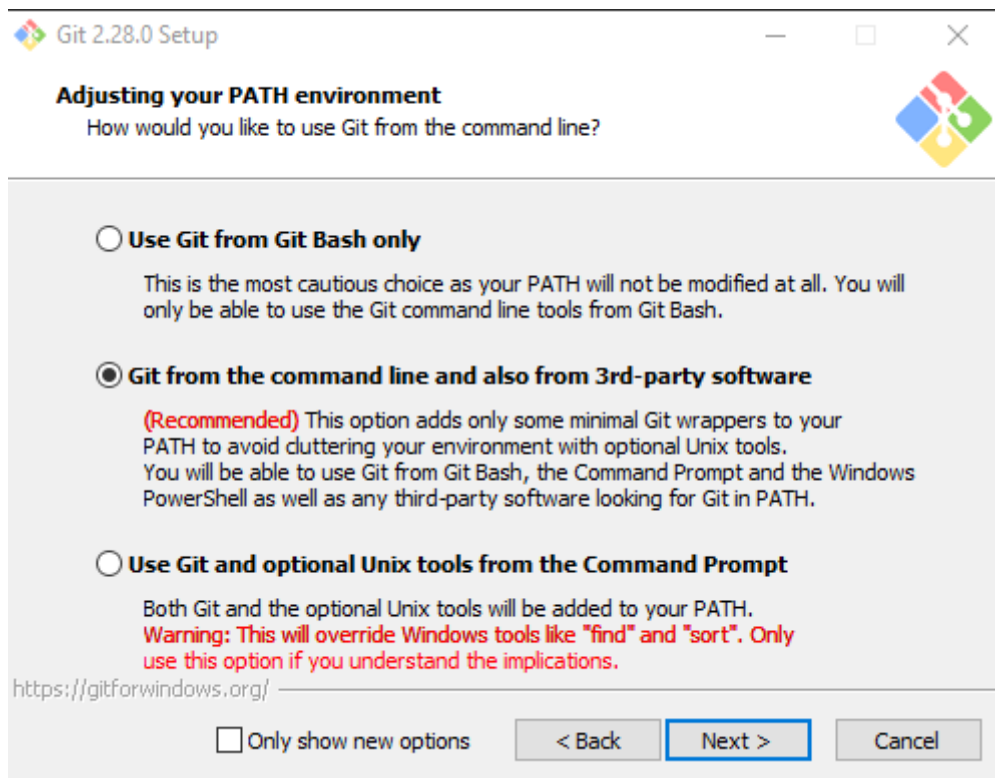
Dejamos marcadas las casillas por defecto.



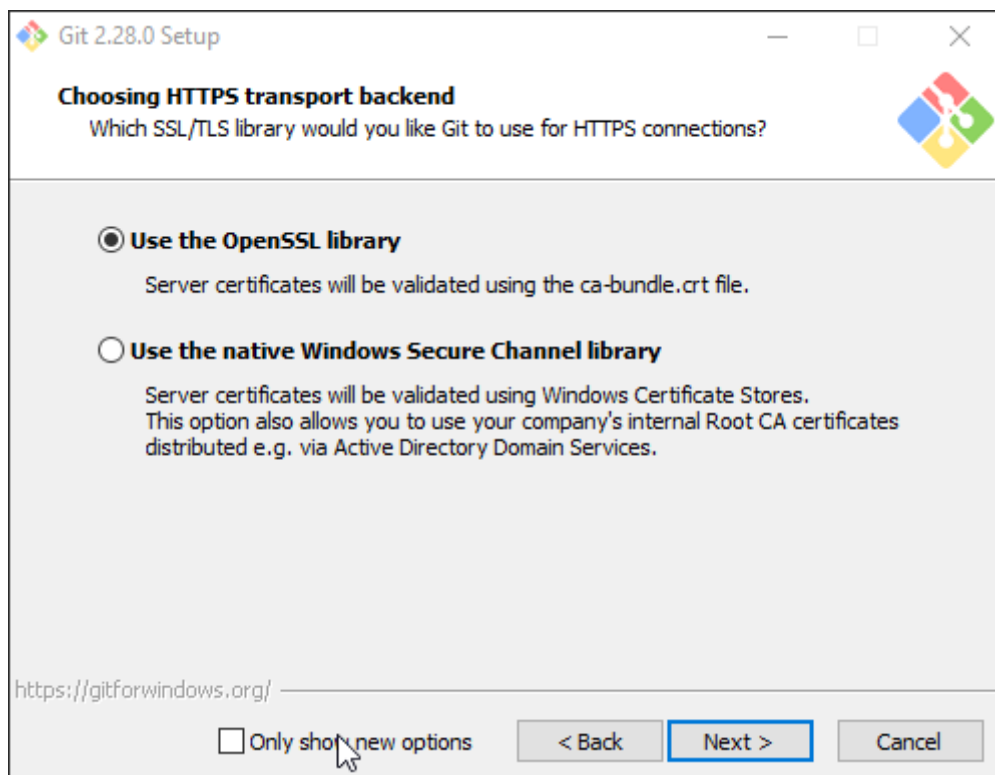
Dejamos por defecto la opción que nos viene indicada.



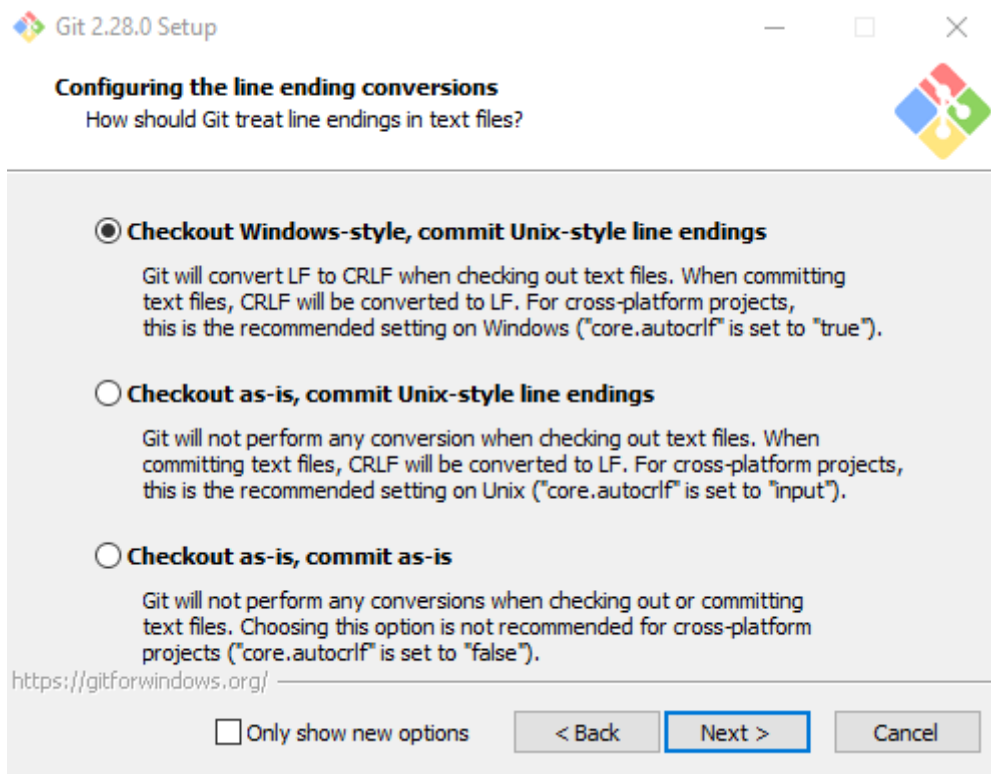
Marcamos la segunda casilla que es la recomendada.



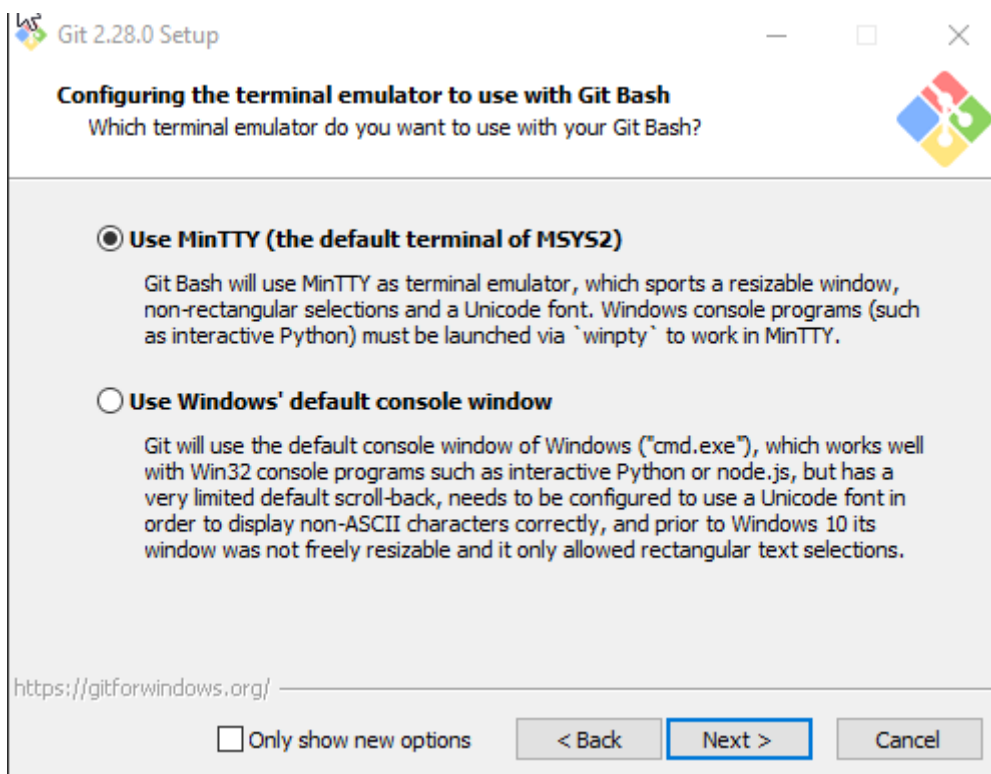
Marcamos la casilla de OpenSSL Library



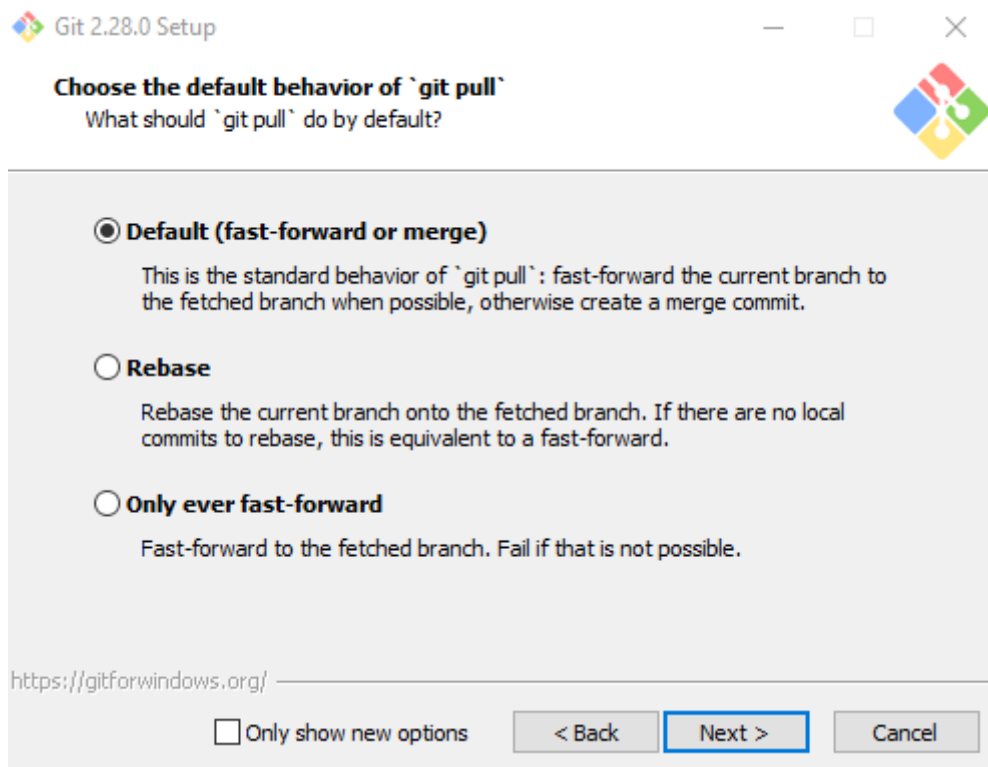
En esta ocasión marcamos la primera casilla.



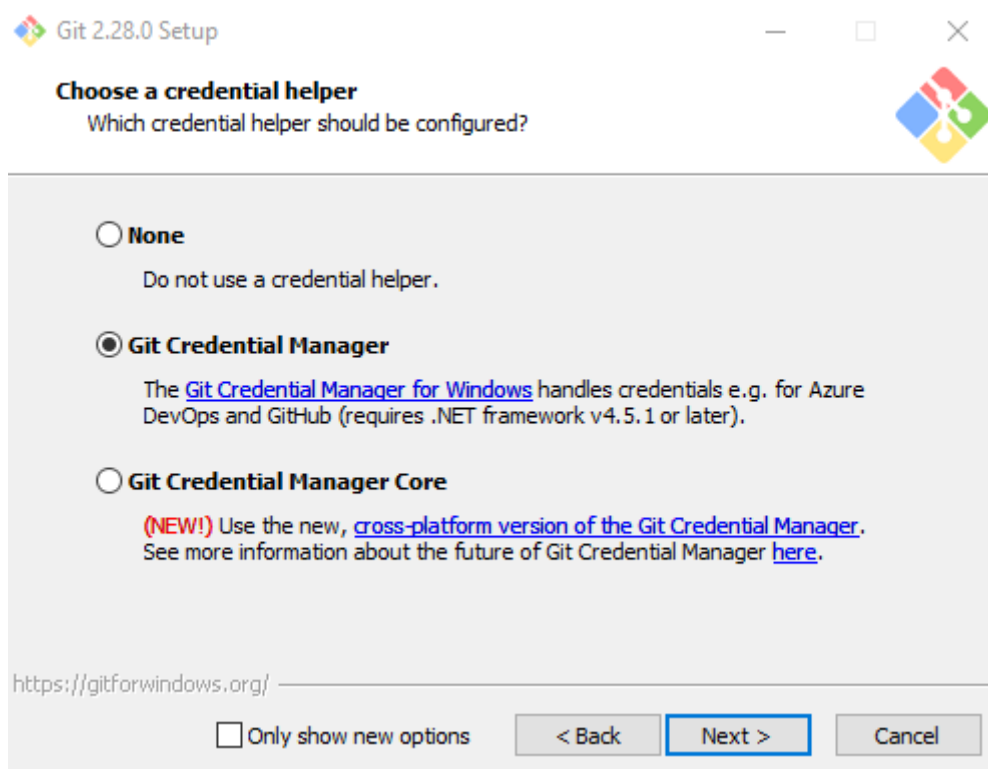
Dejamos seleccionada la primera casilla.



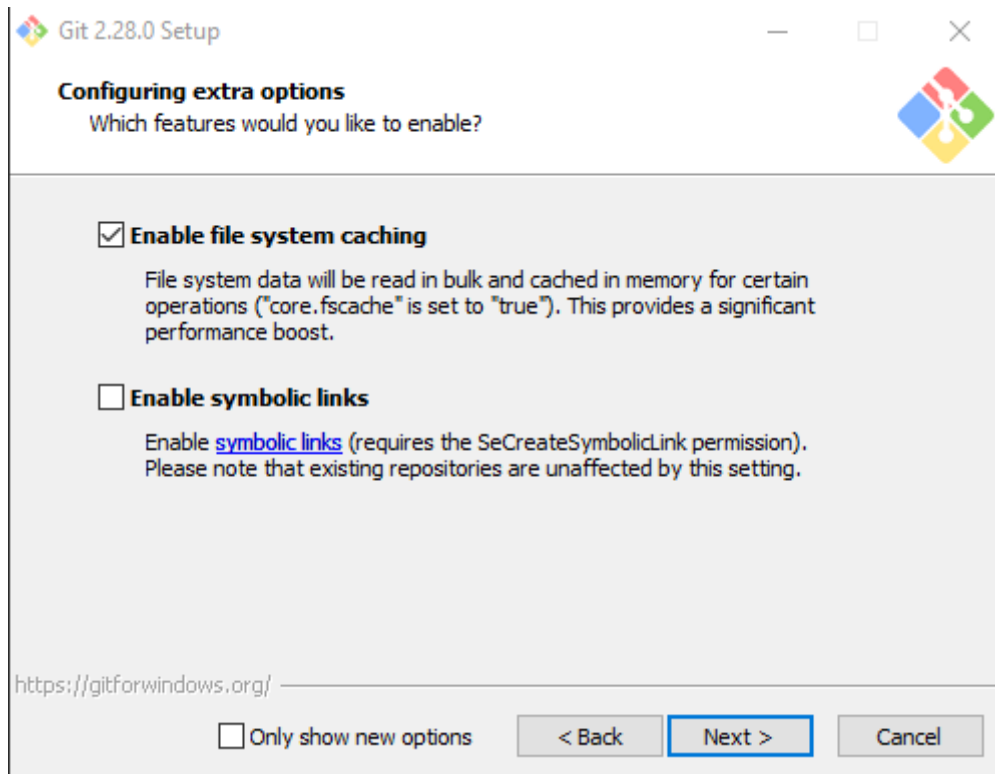
Seleccionamos la casilla de por defecto



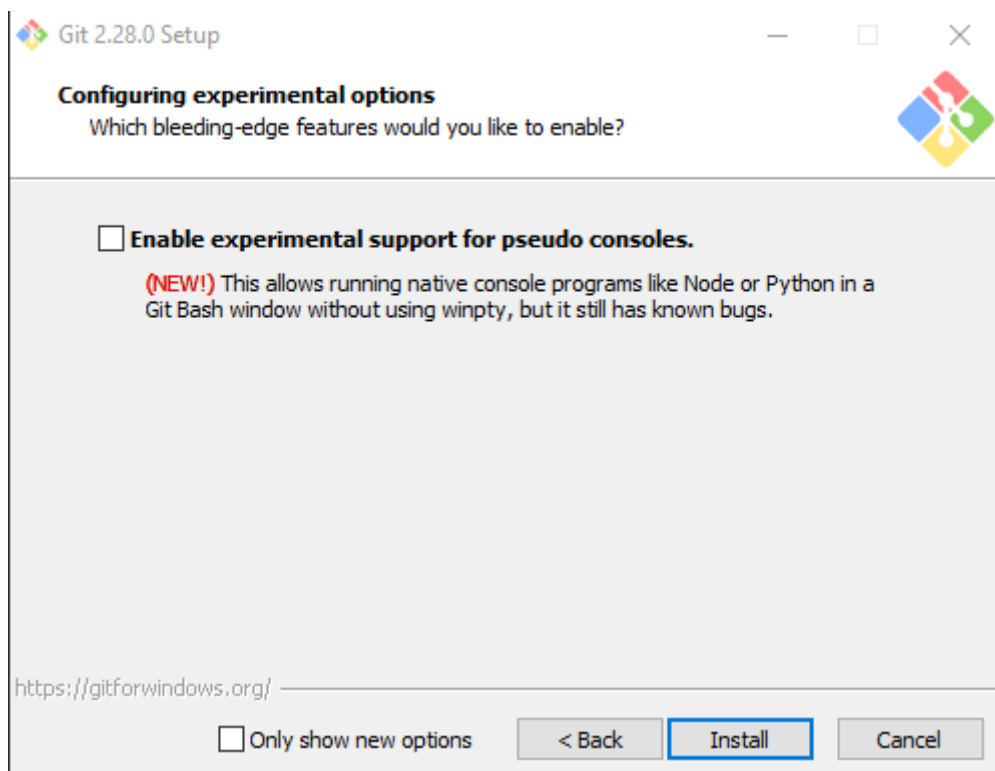
Dejamos marcada la segunda casilla



Marcamos la primera casilla



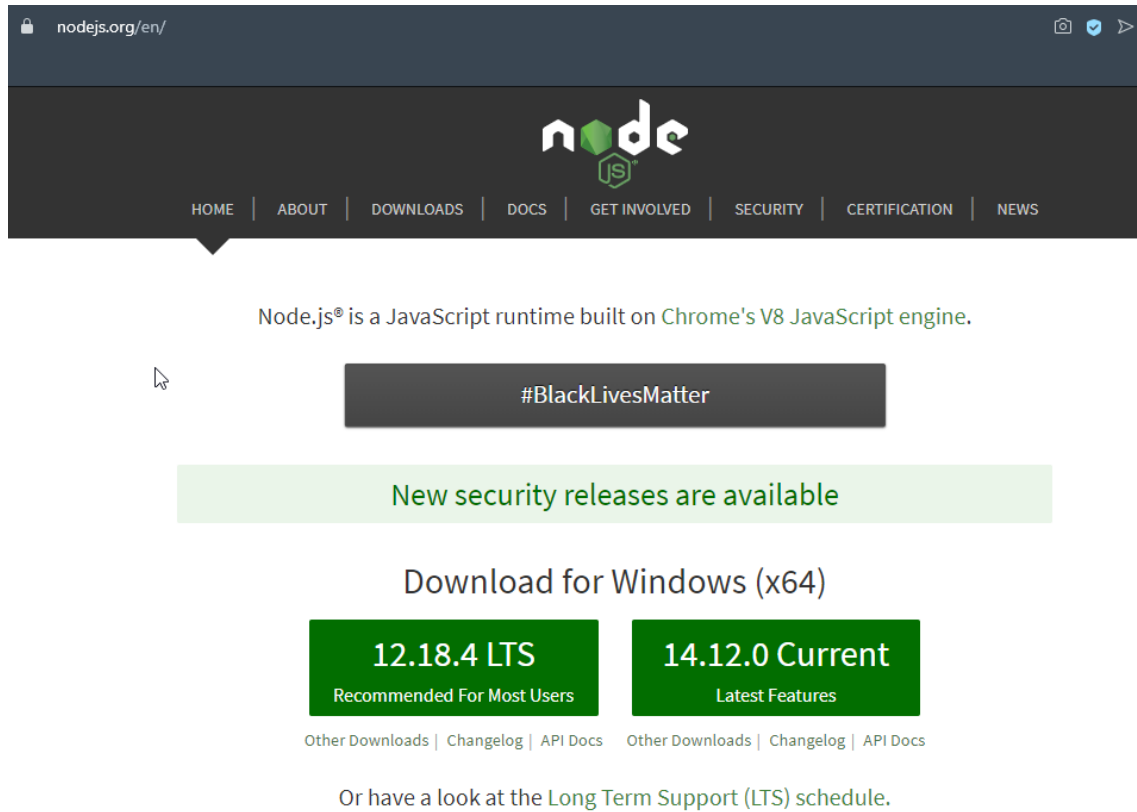
En esta ventana no marcamos nada y le damos a instalar.



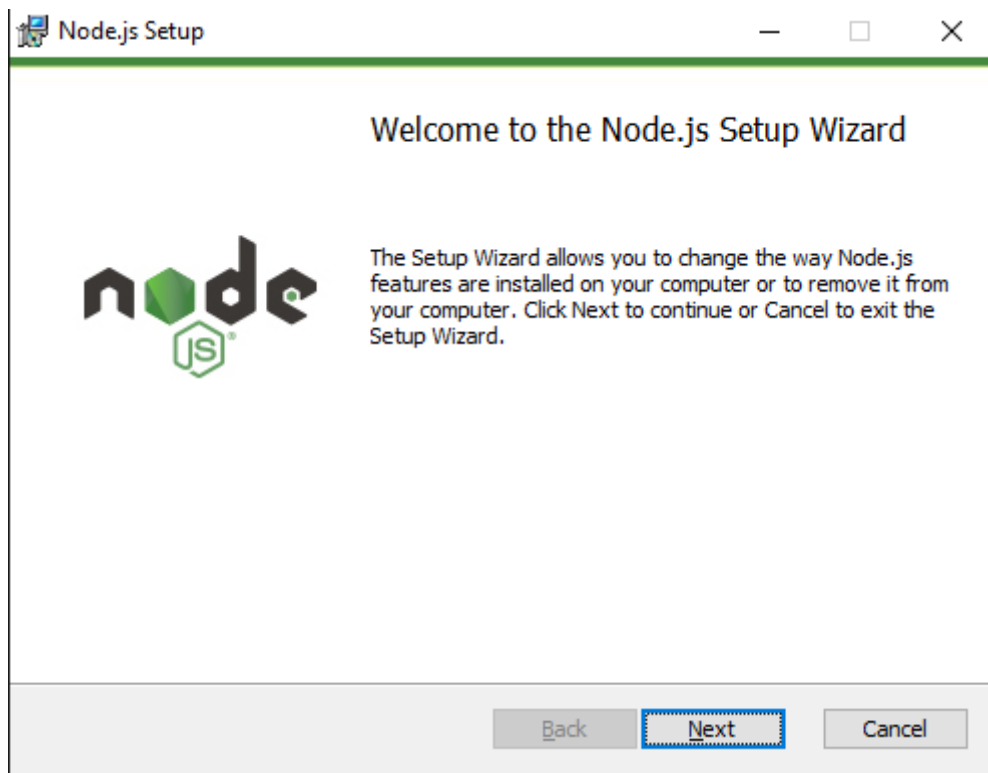
Cuando termine ya tendremos instalado git. Para finalizar la instalación tenemos que ir a la powershell e introducir el comando "git --init" y ya tendríamos instalado completamente.



Ahora procedemos a instalar nodejs, para ello descargamos el instalador, para ello nos vamos a la siguiente página web: [enlace](https://nodejs.org/en/) . Y descargamos la versión recomendada, en mi caso es la 12.18.4 LTS



Una vez que lo tengamos descargado nos ponemos a instalarlo. La primera pestaña le damos a siguiente.




Dejamos las opciones que vienen marcadas por defecto, y cuando termine la instalación.

## 2. Creamos el repositorio

Una vez que lo tengamos instalado, nos vamos a la powershell e introducimos el siguiente comando “npm install -g typescript” (este comando lo encontramos en la página de TypeScript [enlace](#) ). Y con esto ya tendremos instalado typescript.

Una vez tengamos estos programas instalados, creamos un repositorio en nuestro perfil de github.

Owner \*

 munozrodriguezdaniel ▾

Repository name \*

290920\_primer\_proyecto ✓

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

### 3. Creamos el proyecto

Creamos una carpeta que sea igual que nuestro repositorio, dentro de dicha carpeta creamos una carpeta que se llama src que será la que contendrá el fichero tsconfig.tsc que generaremos mediante el comando tsc -init.

Dentro del archivo realizaremos dos modificaciones, la primera será cambiar el valor de target a “es6” y la otra es descomentar la línea de outDir y ponerle el valor de “./dist”

Ahora dentro de la carpeta src crearemos un fichero de TypeScript

Ahora en la terminal ejecutamos el comando “tsc -w” para que cuando guardemos los datos la consola automáticamente lo compile y cree el fichero js.

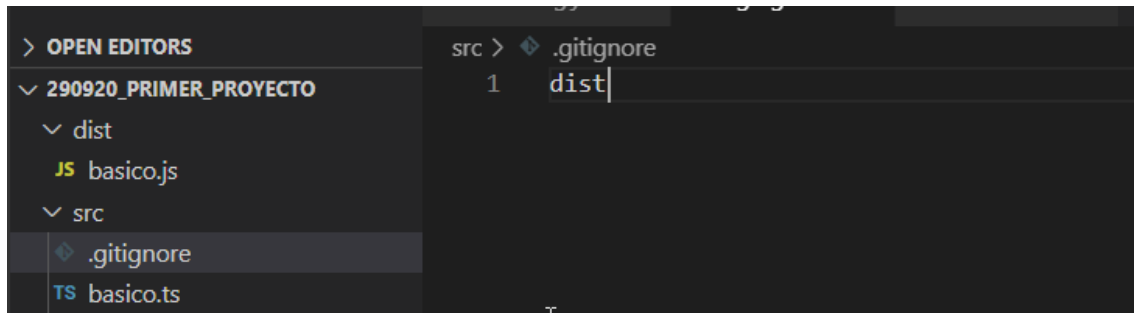
Y como vemos nos crea la carpeta con el fichero en js

▼ dist

JS basico.js

#### 4. Sincronización del trabajo con el repositorio

Ahora vamos a sincronizar nuestro trabajo con el repositorio, lo primero que debemos de hacer es crear un archivo que se llame gitignore para que cuando sincronizemos se ignore la carpeta dist.



Ahora en la terminal introducimos los siguientes comandos:

- git init
- git commit -m "Primer commit" (hacemos el commit y ponemos un mensaje)
- git branch -M master
- git remote add origin [https://github.com/munozrodriguezdaniel/290920\\_primer\\_proyecto.git](https://github.com/munozrodriguezdaniel/290920_primer_proyecto.git) (aquí declaramos donde vamos a sincronizar nuestro trabajo)
- git push -u origin master (y por último subimos el contenido)

```
S C:\Users\usuario\Desktop\290920_primer_proyecto> git init
Initialized existing Git repository in C:/Users/usuario/Desktop/290920_primer_proyecto/.git/
S C:\Users\usuario\Desktop\290920_primer_proyecto> git commit -m "Primer commit"
[master (root-commit) 42f27f4] Primer commit
3 files changed, 71 insertions(+)
create mode 100644 src/.gitignore
create mode 100644 src/basico.ts
create mode 100644 tsconfig.json
S C:\Users\usuario\Desktop\290920_primer_proyecto> git branch -M master
S C:\Users\usuario\Desktop\290920_primer_proyecto> git remote add origin https://github.com/munozrodriguezdaniel/290920_primer_proyecto.git
S C:\Users\usuario\Desktop\290920_primer_proyecto> git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), 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/munozrodriguezdaniel/290920_primer_proyecto.git
* [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
S C:\Users\usuario\Desktop\290920_primer_proyecto>
```

Las siguientes veces que queramos hacer un push no habrá que volver a usar todos los comandos anteriores solo los siguientes:

- git add .
- git commit -m ""
- git push -u origin master