



git

PRÁCTICA EVALUABLE 7

Jose Juan Navarro Giner

Instalación de Git	2
Subida del proyecto mediante Git	3

Instalación de Git

Git es un software que nos permite manejar el uso de un control de versiones. Nuestros proyectos son subidos a un **repositorio**, el cual vamos actualizando mediante sentencias '**commit**' y teniendo así un registro de anteriores sentencias del mismo tipo.

Para la instalación de **Git** en nuestro sistema Windows, en primer lugar, simplemente buscamos 'Git' en Google, accediendo después a la opción de '**Download for Windows**', tal y como se muestra en la Fig. 1.

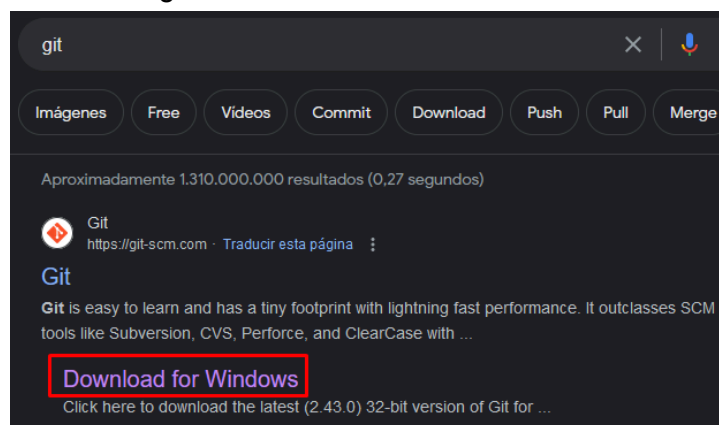


Fig. 1 - Instalación de Git I

Una vez dentro, descargamos la versión de 64 bits, tal y como se muestra en la Fig. 2.

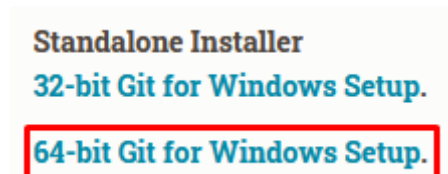


Fig. 2 - Instalación de Git II

Una vez hayamos abierto el instalador y habiendo otorgado los permisos de administrador, simplemente haremos clic en '**Siguiente**'. Git comenzará a instalarse en nuestro equipo sin problemas, como se ve en la Fig. 3.

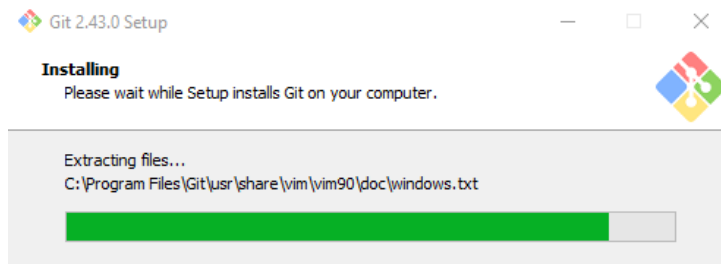


Fig. 3 - Instalación de Git III

Al acabar simplemente damos en finalizar y Git se habrá instalado exitosamente en nuestro sistema.

Subida del proyecto mediante Git

Habiendo creado un repositorio en **GitHub** de nuestro proyecto, vamos a subir nuestro proyecto a dicho repositorio mediante el uso de los comandos que ofrece Git. En primer lugar, utilizamos el comando que se muestra en la Fig. 4, el cual sirve para **inicializar un repositorio de Git vacío**.

```
C:\Users\josej>cd C:\Users\josej\Documents\ejemplo_JSDOC  
C:\Users\josej\Documents\ejemplo_JSDOC>git init  
Initialized empty Git repository in C:/Users/josej/Documents/ejemplo_JSDOC/.git/  
C:\Users\josej\Documents\ejemplo_JSDOC>
```

Fig. 4 - Inicializar repositorio Git

Habiendo creado el repositorio, escribimos el comando de la Fig. 5. Este comando sirve para poder **subir todos los archivos de una carpeta al repositorio**.

```
git add --all
```

Fig. 5 - Subida de ficheros

Una vez que el repositorio tenga cargados los ficheros que queremos subir, ejecutamos el comando de la Fig. 6, con la cual realizaremos nuestra **primera modificación** en nuestro repositorio. Le indicamos un **mensaje** entre comillas dobles, seguido de la instrucción **-m**, para indicar que queremos otorgarle un mensaje al commit.

```
>git commit -m "primer commit"
```

Fig. 6 - Primer commit

Ahora, mediante el comando mostrado de la Fig. 7, creamos la **rama Main**, que será la **rama principal de nuestro repositorio**.

```
>git branch -M main
```

Fig.7 - Creación de rama main

Ahora vamos a subirlo a nuestro repositorio remoto. Con el comando de la Fig. 8, le indicamos **a qué repositorio se subirá**, indicando su origen, es decir, la dirección del repositorio.

```
git remote add origin https://github.com/Joseju002/JSDoc.git
```

Fig. 8 - Subida a un repositorio remoto

Finalmente, mediante el comando de la Fig. 9, le indicamos **a qué rama se subirán los cambios**.

```
git push -u origin main
```

Fig. 9 - Subida a la rama main

Si vamos a la página de nuestro repositorio en GitHub, vemos como todos los ficheros de la carpeta se han subido correctamente (Fig. 10). Se nos indicará, además, el **nombre de la instrucción commit** con la que se subieron los ficheros. En mi caso, el mensaje del commit fue 'primer commit', por lo que se muestra como tal.

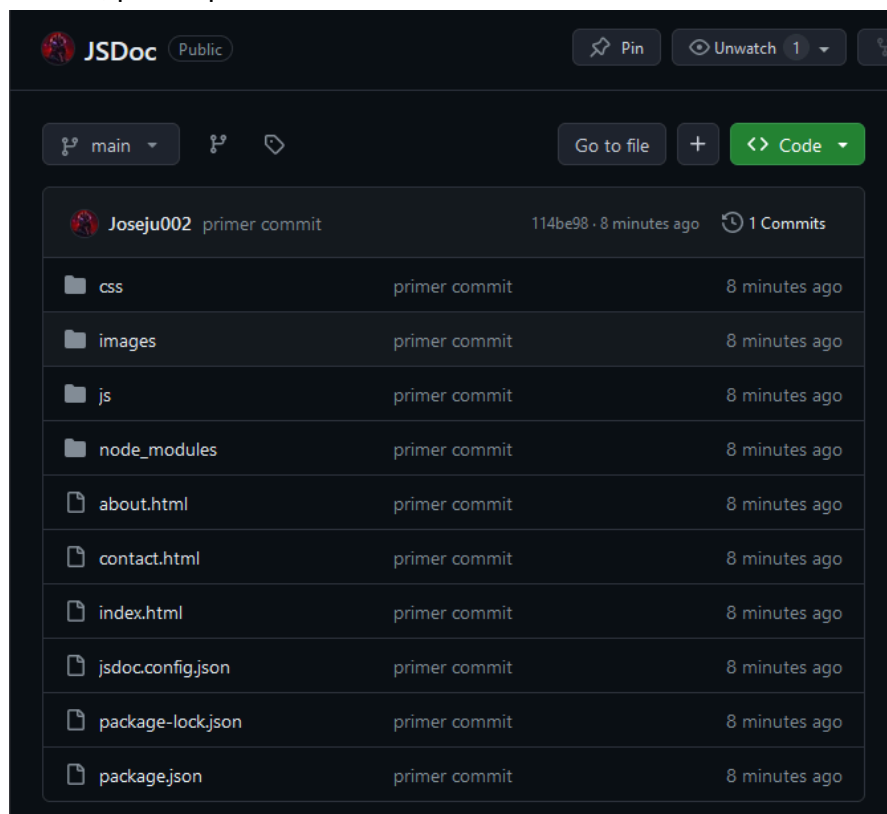


Fig. 10 - Repositorio en GitHub

Digamos que ahora queremos subir nuestro proyecto con ciertas modificaciones a una **rama distinta** de nuestro repositorio de Git. En primer lugar, podemos mirar las ramas que tenemos creadas en nuestro repositorio mediante el comando de la Fig. 11.

```
C:\Users\josej\Documents\ejemplo_JSDOC>git branch
* main
```

Fig. 11 - Ramas del repositorio

Como vemos, solo aparece la **rama main**, ya que simplemente hemos creado esa. Vamos a proceder y vamos a crear una nueva rama.

Mediante el comando que se muestra en la Fig. 12, podemos crear una **nueva rama**, seguido del nombre de la rama.

```
C:\Users\josej\Documents\ejemplo_JSDOC>git branch JSDoc  
C:\Users\josej\Documents\ejemplo_JSDOC>git branch  
JSDoc  
* main
```

Fig. 12 - Creación de rama secundaria

Como vemos también en la Fig. 12, mediante un **asterisco (*)** se nos indica que nos encontramos actualmente en la rama main.

Digamos que queremos movernos a la rama **JSDoc**, la rama que acabamos de crear, para trabajar en ella. Para ello, utilizamos el comando de la Fig. 13, en la cual pasamos de una rama a la que le estamos indicando. Nos indicará que hemos cambiado de rama en la consola.

```
C:\Users\josej\Documents\ejemplo_JSDOC>git checkout JSDoc  
Switched to branch 'JSDoc'  
C:\Users\josej\Documents\ejemplo_JSDOC>git branch  
* JSDoc  
main
```

Fig. 13 - Cambio de rama

Como también podemos ver en la Fig. 13, vemos que el asterisco ahora se encuentra en la rama **JSDoc**.

Ya hemos generado la documentación **JSDoc** de nuestro proyecto, de modo que vamos a subir dicho **fichero** generado en la rama que hemos creado anteriormente. Para eso, volvemos a hacer el comando **git add**. En este caso, simplemente voy a subir la carpeta **out** (que es en la que se ha generado la documentación) mediante el comando de la Fig. 14.

```
>git add out
```

Fig. 14 - Subida de ficheros en otra rama

Vamos a hacer también su respectiva instrucción **commit**, como se ve en el comando de la Fig. 15.

```
git commit -m "documentacion"
```

Fig. 15 - Commit en la segunda rama

Finalmente, mediante el comando de la Fig. 16, subimos a la rama los **cambios**.

```
git push origin JSDoc
```

Fig. 16 - Subida a la rama JSDoc

Si vamos a nuestro repositorio subido a GitHub (Fig. 17), vemos que ahora se nos indica que existen **dos ramas**. Si vamos a la rama **JSDoc**, en la cual hemos realizado el commit para la subida de la documentación, vemos que dicho fichero o carpeta se ha subido en **dicha sentencia commit**.

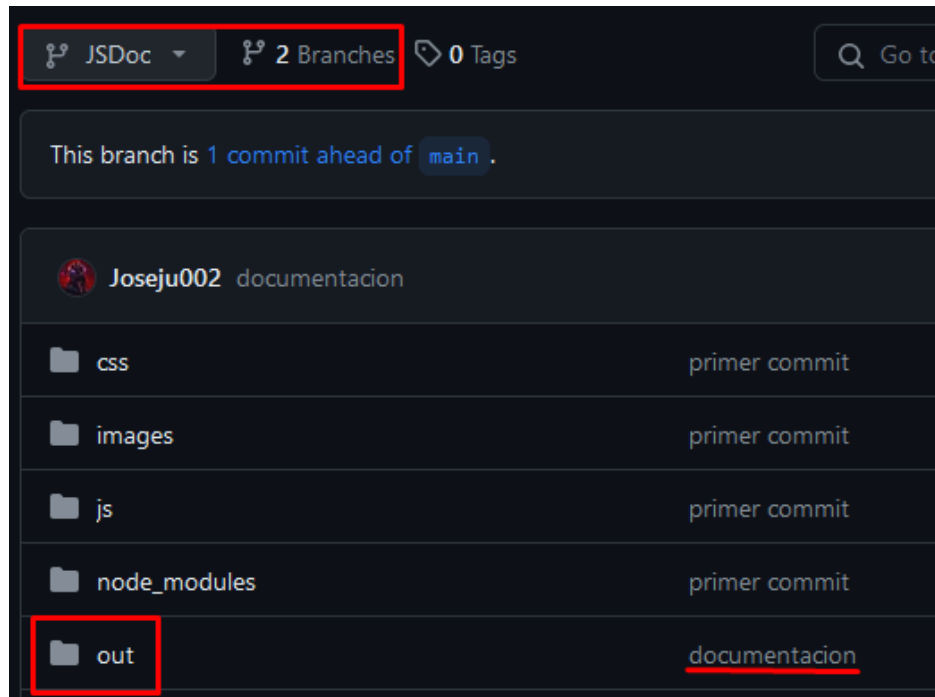


Fig. 17 - Repositorio de GitHub actualizado