

MANUAL GITHUB

Emilio Sánchez
SGBD 2ª ASIR

Índice Github

| | | |
|----|-----------------------------|----------|
| 1. | Introducción | página 2 |
| 2. | Descarga e instalación..... | página 3 |
| 3. | Configuración..... | página 5 |
| 4. | Comandos..... | página 6 |
| 5. | Primer commit | página 7 |
| | a. Añadir un README..... | página 8 |

Introducción

Vamos a ver la instalación de Github, Git y la configuración de sus carpetas.

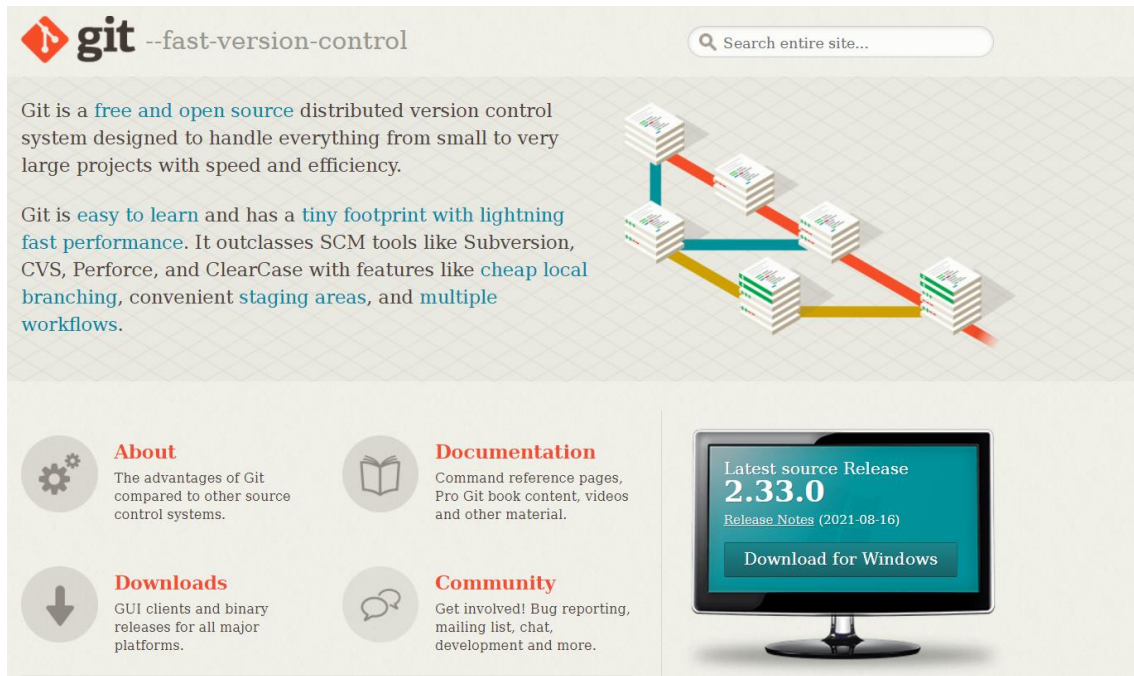
Git es un software de control de versiones, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.



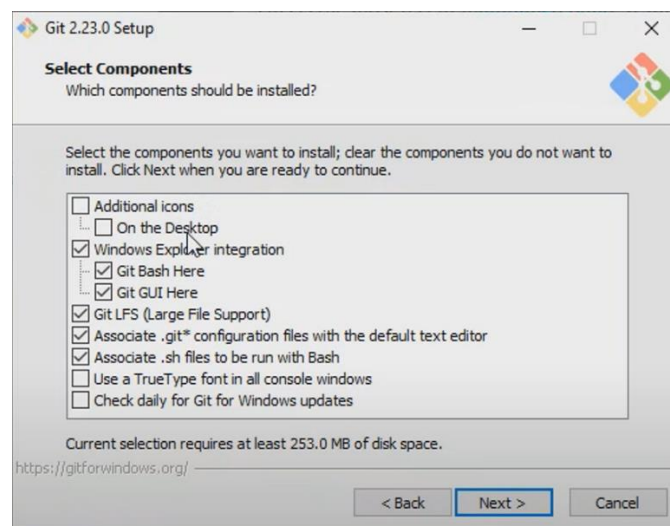
GitHub es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

Descarga e instalación

Primero procederemos a descargar Git, ya que nos hará falta para el funcionamiento de Github. Accederemos a la página oficial de Git desde [aquí](#) o buscándolo en cualquier buscador.



Instalaremos el ejecutable clicando en Next, sin cambiar ninguna otra opción.



Para comprobar si hemos realizado correctamente la instalación abriremos la **PowerShell**. Allí escribiremos **Git**, para comprobar si está instalado.

```
PS C:\Users\emili> git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      [--super-prefix=<path>] [--config-env=<name>=<envvar>]
      <command> [<args>]

These are common Git commands used in various situations:
```

```
PS C:\Users\emili> git --version
git version 2.33.0.windows.2
```


A continuación, procederemos a instalar Github. Iremos a la página de Github Desktop, tan solo pulsando [aquí](#).



Configuración

Tras crearnos una cuenta en [Github](#), procederemos a crear una carpeta que queramos subir a Github. En la página de Github crearemos la otra carpeta donde se subirá la que esté en nuestro ordenador. En este caso llamaremos **20-09-2021_practica1**.



| | |
|---|--------------------------|
| Owner * | Repository name * |
|  SanchezGarciaEmilio ▾ | / 20-09-2021_practica1 ✓ |

A continuación crearemos unas credenciales para tener nuestro ordenador enlazado a nuestra cuenta de Github.

```
PS C:\Users\emili> git config --global user.email "esangar2211@g.educaand.es"
```

```
PS C:\Users\emili> git config --global user.name "SanchezGarciaEmilio"
```

Podemos luego comprobar que las credenciales se han guardado desde el **Administrador de credenciales**.

[git:https://github.com](https://github.com)

Fecha de modificación: 07/03/2021



Comandos

Git contiene varios comandos para su uso. Entre los más comunes y los que usaremos será:

- **git init:** inicia git creando un fichero para poder subir los ficheros a Github.
- **git add:** añade archivos o ficheros para subirlos. Añadiendo un punto al final añadirá todos los ficheros. Se puede usar un **.gitignore** para añadir todos menos algunos ficheros.
- **git status:** comprueba el estado de los ficheros. Si están subidos, desactualizados o sin subir.
- **git commit:** añade ficheros o archivos para prepararlos antes de subirlos.
- **git push:** se suben los archivos añadidos con add y nombrados con un commit.

Primer Commit

En Visual Studio Code crearemos la carpeta con sus archivos. Crearemos un archivo **.md** para poder subir algo a Github. Pulsando **ctrl+shit+ñ** abriremos el terminal, allí meteremos los siguientes comandos.

Iniciaremos con **git init**.

```
PS C:\miscosas\01-Estudios\01-SegAsir\SGBD\Primer_trimestre\20-09-2021_Practica1> git init
Initialized empty Git repository in C:/miscosas/01-Estudios/01-SegAsir/SGBD/Primer_trimestre/20-09-2021_Practica1/.git/
```

Añadiremos **git add .** para añadir todos los archivos que tengamos en la carpeta, sino a continuación de **git add** pondremos el nombre de los archivos. Podemos chequear continuamente el estado de nuestros archivos con **git status**. Con **git commit -m ""** prepararemos nuestros archivos para subirlos, dentro de las comillas pondremos un motivo del commit que estamos haciendo.

```
PS C:\miscosas\01-Estudios\01-SegAsir\SGBD\Primer_trimestre\20-09-2021_Practica1> git commit -m "first commit"
[master (root-commit) 8f6b78c] first commit
1 file changed, 15 insertions(+)
create mode 100644 README.md
PS C:\miscosas\01-Estudios\01-SegAsir\SGBD\Primer_trimestre\20-09-2021_Practica1> git status
On branch master
nothing to commit, working tree clean
```

Tendremos que enlazar el archivo con el repositorio que hemos creado. Lo haremos con el siguiente comando, usando el link del repositorio.

```
PS C:\miscosas\01-Estudios\01-SegAsir\SGBD\Primer_trimestre\20-09-2021_Practica1> git remote add origin https://github.com/SanchezGarciaEmilio/20-09-2021_practica1.git
```

Finalmente añadiremos un **git push** para subir todos los archivos que hallamos pasado previamente por el commit.

```
PS C:\miscosas\01-Estudios\01-SegAsir\SGBD\Primer_trimestre\20-09-2021_Practica1> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 581 bytes | 290.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/SanchezGarciaEmilio/20-09-2021_practica1.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Si alguien con acceso al repositorio sube algo que queremos tener en nuestro ordenador actualizado, debemos poner en el terminal **git pull**.

Si queremos descargar una carpeta subida a Github, deberemos clonarla.



Podremos abrirlo con Github desktop, descargarlo como un Zip, pero si queremos clonarlo deberemos copiar el enlace que aparece al clicar en Code, y abriendo el terminal (CMD) en la carpeta donde deseamos clonarlo, pondremos **git clone** y a continuación el enlace.

Añadir un README

Al momento de la creación podemos añadir un README al archivo, tan solo clickando en él cuando creamos el repositorio (también se puede crear un archivo al que nombremos README con la misma función).

Initialize this repository with:

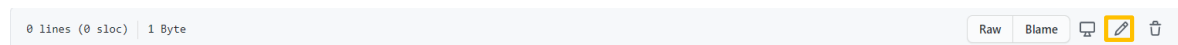
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

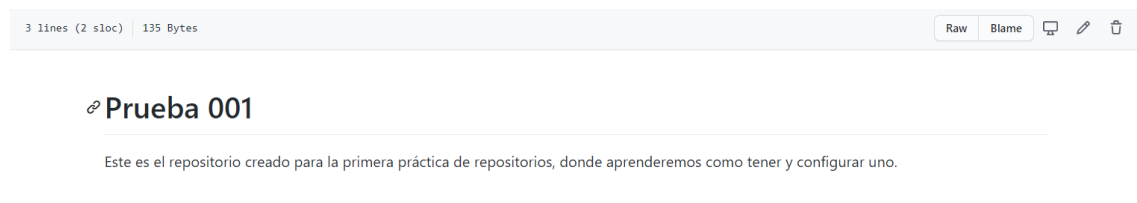
☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

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

Desde el propio Github podremos editar nuestro README, tan solo clicando en el archivo y luego en editar. Una vez dentro pondremos lo que deseamos, en este caso una explicación de que tratará nuestro repositorio.



Podremos hasta añadir un # delante de nuestro texto para ponerlo a modo de título o un + si queremos poner listas. Nos podría quedar algo así:



Finalmente nos quedaría un repositorio tal que [así](#).