<codoacodo/>

Desarrollo Fullstack

BA

Les damos la bienvenida

Vamos a comenzar a grabar la clase

GIT

Control de Versiones

GIT

Esta herramienta creada por *Linus Torvalds* nos permite llevar un **versionado** de nuestros proyectos, como si se tratara de una **línea de tiempo** sobre la cual podemos navegar, volviendo a código anterior o creando múltiples versiones del mismo.

Instalación de GIT

Para instalarlo debemos dirigirnos al sitio oficial:

https://git-scm.com/

y **descargar** la versión correspondiente a nuestro sistema operativo.

Una vez instalado, **en la terminal** corremos el comando `git --version` y **nos devolverá** la versión que tenemos instalada.

```
$ git --version
git version 2.37.1.windows.1
```

Configuración

Ahora toca configurar nuestro perfil de GIT dentro de la PC.

Para eso debemos establecer un **nombre de usuario** y un **correo** con el fin que cada versión que creemos de nuestro código **registre** que hemos sido el autor de **esos cambios**.

Establecemos el **nombre de usuario** y correo mediante los comandos:

```
git config --global user.name “username”
```

```
git config --global user.email correo@correo.com
```

```
git config --global core.editor “code --wait”
```

Finalmente corremos `git config --list` para validar que hemos agregado bien la configuración.

Nota: cualquier comando que corramos debe comenzar con la palabra git, seguido de la instrucción deseada.

Una vez configurado GIT, ya podemos comenzar.

Iniciar un repositorio

Para "habilitar" **GIT** en nuestro proyecto debemos dirigirnos desde la terminal a la carpeta donde lo tengamos guardado y **correr el comando**:

```
git init
```

Esas palabras mágicas crearán una carpeta oculta llamada **.git** donde a partir de ahora se van a **guardar los cambios** que vayamos registrando a medida que trabajamos.

En ese acto, ***creamos lo que se conoce como rama master***, es decir la línea de tiempo principal de nuestro proyecto.

Estados

Lo primero que debemos saber es que GIT posee 3 estados, estos son *working directory*, **stage** y **repository**.

WORKING DIRECTORY

Representa el código actual de nuestro proyecto que ha tenido cambios recientes sin registrar.

STAGE

Son los cambios que deseamos preparar para ser registrados en una siguiente versión.

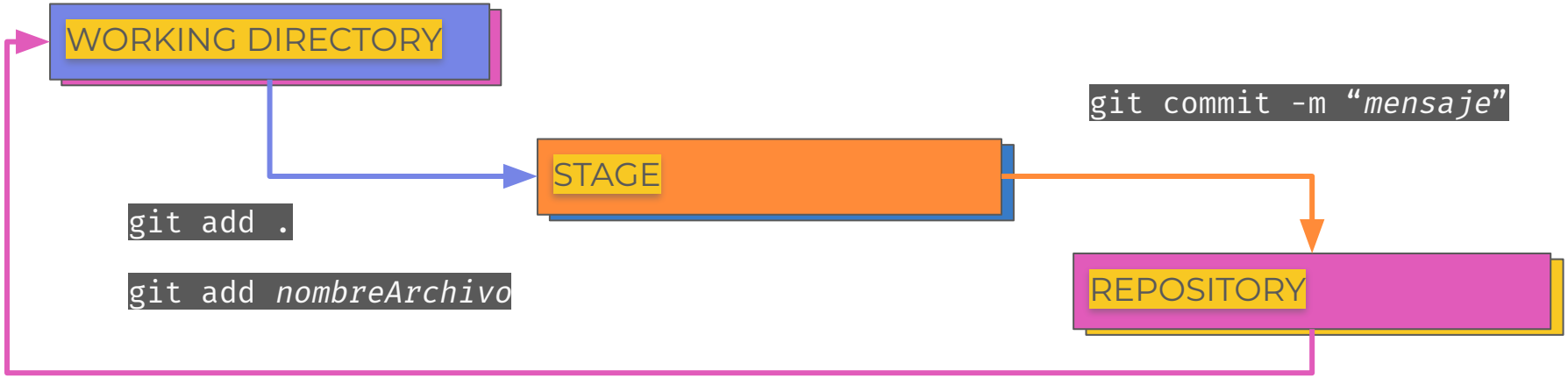
*Paso intermedio entre WD y Repository.

REPOSITORY

Es donde se registran las nuevas versiones de nuestro código en línea de tiempo

Estados

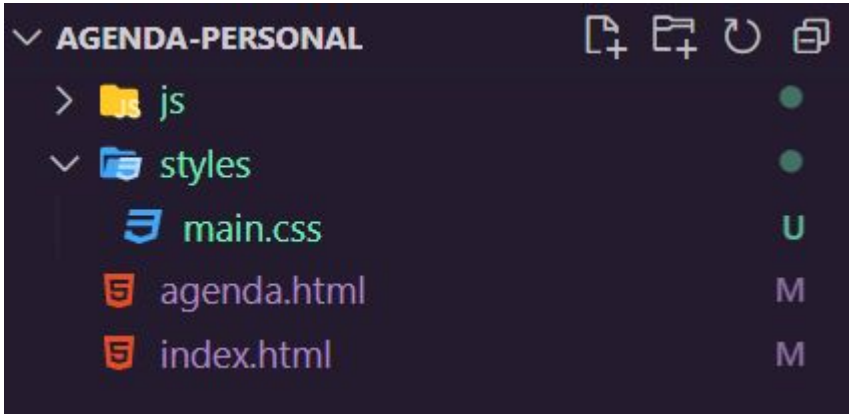
Trabajar con GIT es un proceso cíclico donde pasaremos por **los 3 estados** contínuamente y una vez registrada una nueva versión de nuestro código, **volveremos al estado inicial "working directory"**.



ADD

Al **crear un repositorio de git** o al **registrar una nueva versión**, nuestros archivos se encontrarán en el Working Directory.

En el primer caso en posición de “untracked” ya que GIT **no los conoce** y en el segundo caso, frente a cualquier cambio se pondrán en posición de “modified” el cual **indica que la versión actual** del archivo es diferente **a la última** de la versión registrada.

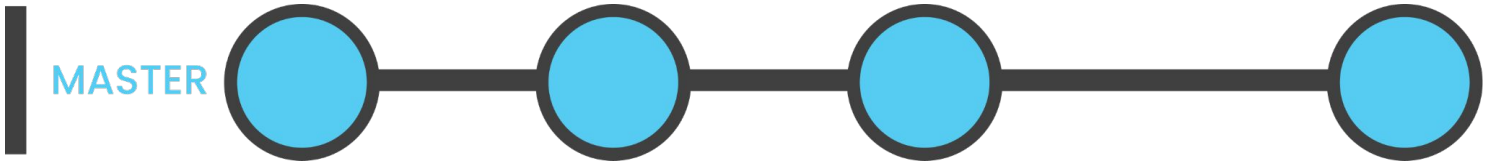


Para enviar esos cambios al stage y prepararlos para el commit, debemos usar el comando `git add` seguido del nombre del archivo o de un punto para agregar todos los cambios al stage

COMMIT

Cada vez que establecemos un nuevo commit mediante el comando `git commit -m “mensaje”`, creamos una nueva versión de nuestro proyecto y **registramos los cambios en la línea de tiempo**.

GIT se encarga de guardar esta información para poder volver a versiones anteriores de nuestro código si fuera necesario.



Esta representación indica que hemos registrado 4 versiones sobre nuestra rama master o rama principal.

STATUS

Si quisiéramos saber el estado de nuestros archivos, si estos se encuentran en el WD o en Stage, entonces **podemos utilizar** el comando `git status` que nos devolverá una lista con el estado de cada cambio realizado.

```
>> Agenda-Personal git:(master) 02:25 git add index.html
>> Agenda-Personal git:(master) 02:25 git add agenda.html
>> Agenda-Personal git:(master) 02:25 git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   agenda.html
        new file:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        js/
        styles/

>> Agenda-Personal git:(master) 02:25
```

LOG

Podemos ver todos los commits o versiones que hemos registrado mediante el comando `git log` el cual **nos entrega la lista de versiones registradas** y la información del autor de cada una.

```
>> Agenda-Personal git:(master) 02:26 git commit -m "Se agrega HTML al proyecto"
[master (root-commit) 2432e4a] Se agrega HTML al proyecto
 2 files changed, 79 insertions(+)
 create mode 100644 agenda.html
 create mode 100644 index.html
>> Agenda-Personal git:(master) 02:27 git log
commit 2432e4a7e412901a8736a2dcfdbdf9c41343642f (HEAD -> master)
Author: SairovARG <pmrovira92@gmail.com>
Date:   Tue Oct 11 02:27:09 2022 -0300

    Se agrega HTML al proyecto
```

GITHUB

Versiones en la Nube

GitHub

GITHUB

Es el espacio online donde podremos guardar una copia de nuestro código, si bien existen alternativas (gitlab), **github** es la más conocida y utilizada del mercado.

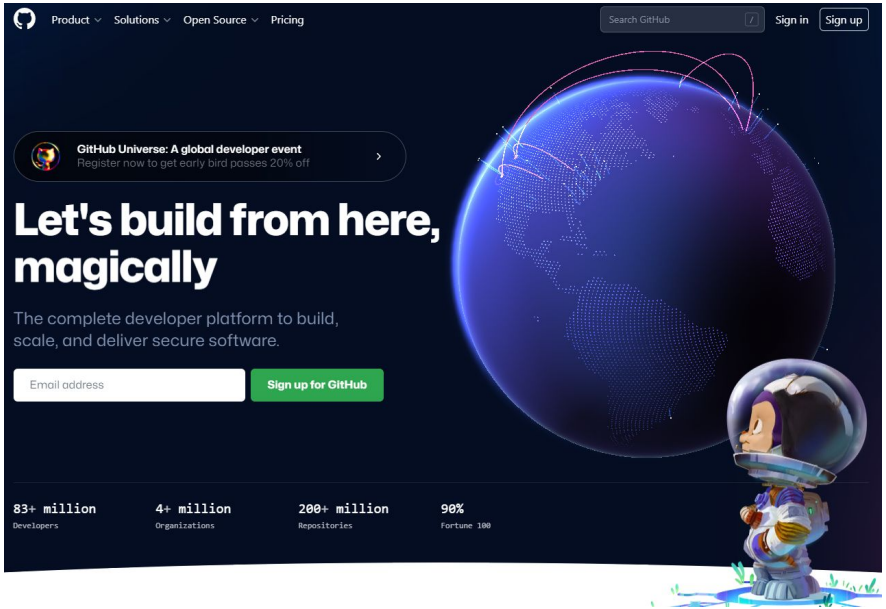
Esta herramienta es **muy importante** para un desarrollador ya que también sirve como forma de mostrar su trabajo.

Creación de cuenta

El primer paso es crearse una cuenta en:

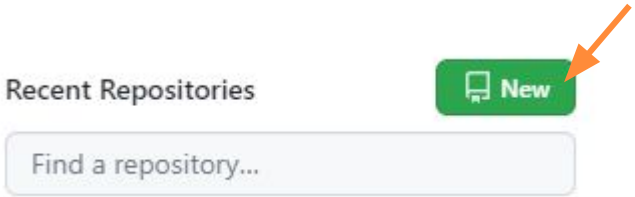
https://github.com/

Cuenta con una versión gratuita con muchos beneficios, como la posibilidad de **contar con infinitos repositorios** públicos.

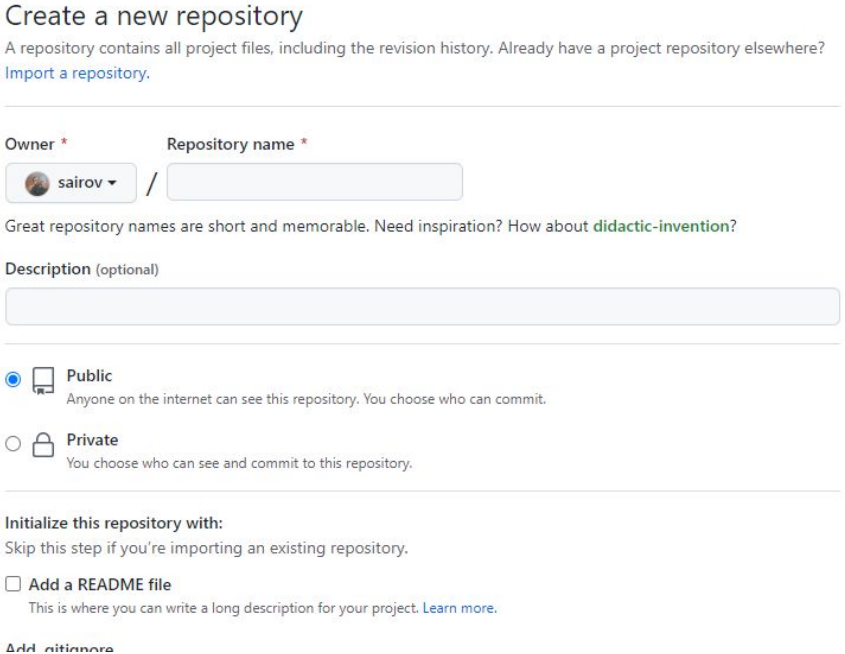


Nuevo Repositorio

El segundo paso es crear un nuevo repositorio remoto.

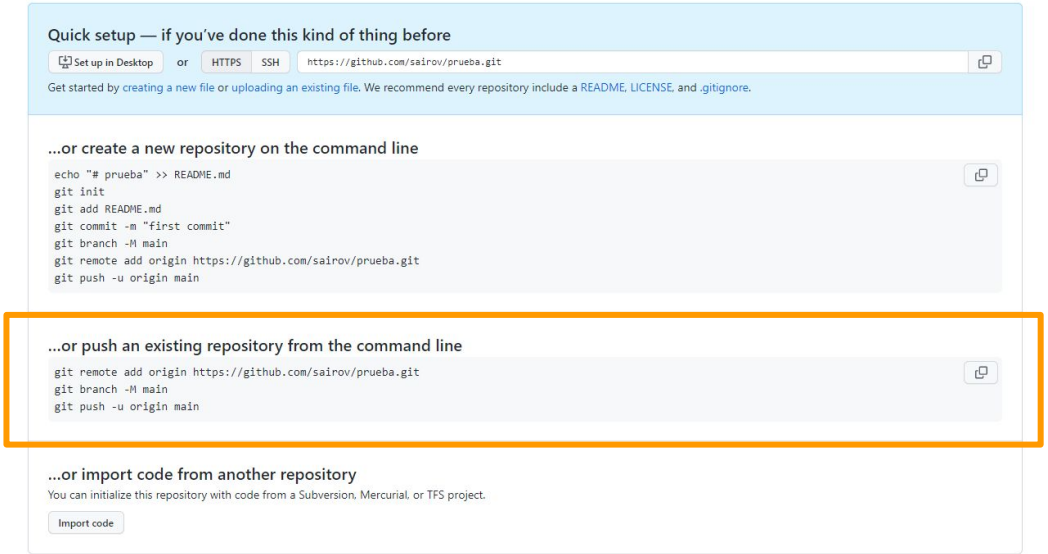


En la pantalla de creación debemos completar el nombre del repositorio y **una descripción (opcional)**, luego lo marcamos como público y le damos a Create repository



Nuevo Repositorio

Una vez creado, ya podemos subir nuestro proyecto siguiendo los pasos que indica github según el caso que nos corresponda.



No te olvides de dar el presente

Recordá:

- Revisar la Cartelera de Novedades.
- Hacer tus consultas en el Foro.

Todo en el Aula Virtual.

Gracias