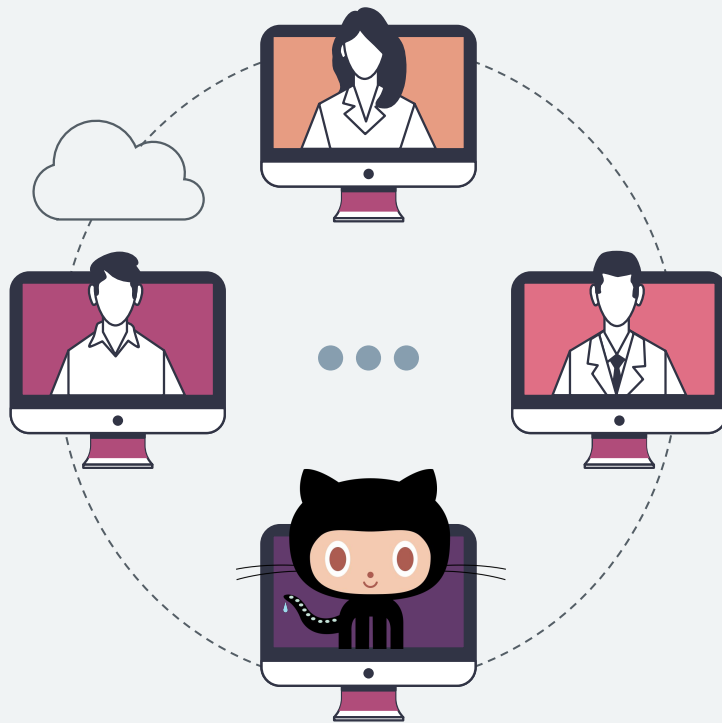


GIT&GITHUB

Clase 1

18/01/21

Introduccion a Git & Github



Sobre este curso

Es un curso práctico, eso implica que el alumno debe participar en las actividades.





git

**Instalar git en
tu computadora
AHORA**

<https://git-scm.com/downloads>
o Googlea "git downloads"



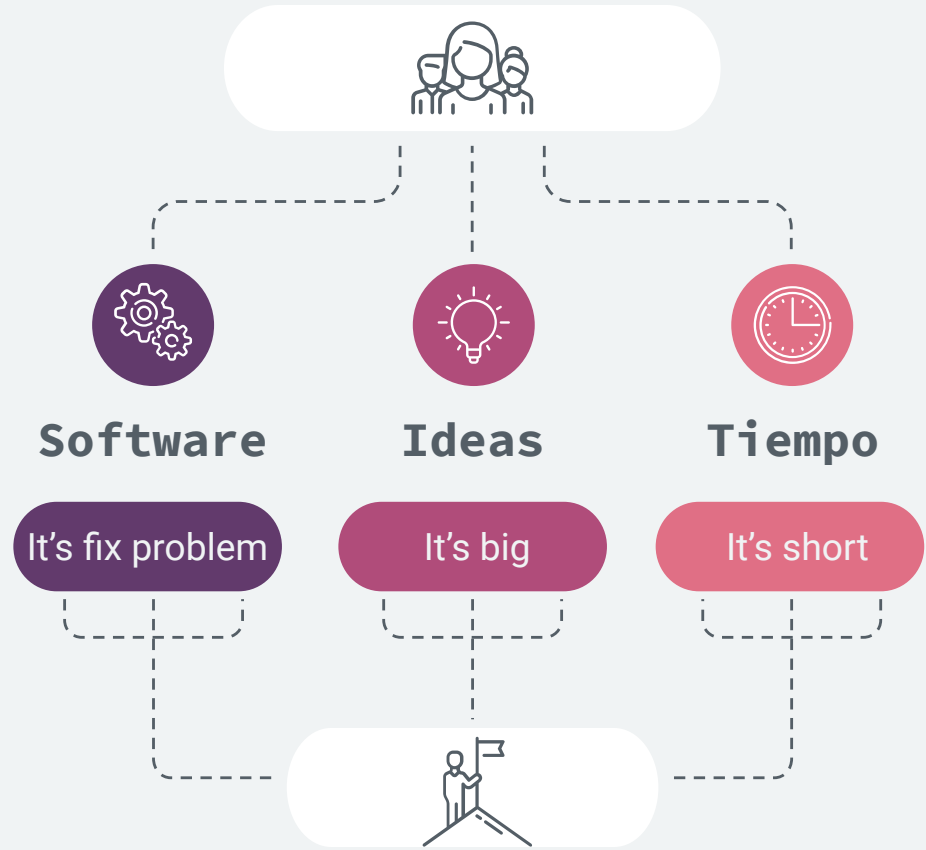
**Más tarde
crearemos una
cuenta de Github**

<https://github.com/>
o Googlea "Github"

Objetivos de la clase de hoy

Entender que son ;

- Los Sistemas de Control de Versionado
- Un Repositorio
- Git y Github
- Estados de Git
- Primeros comandos



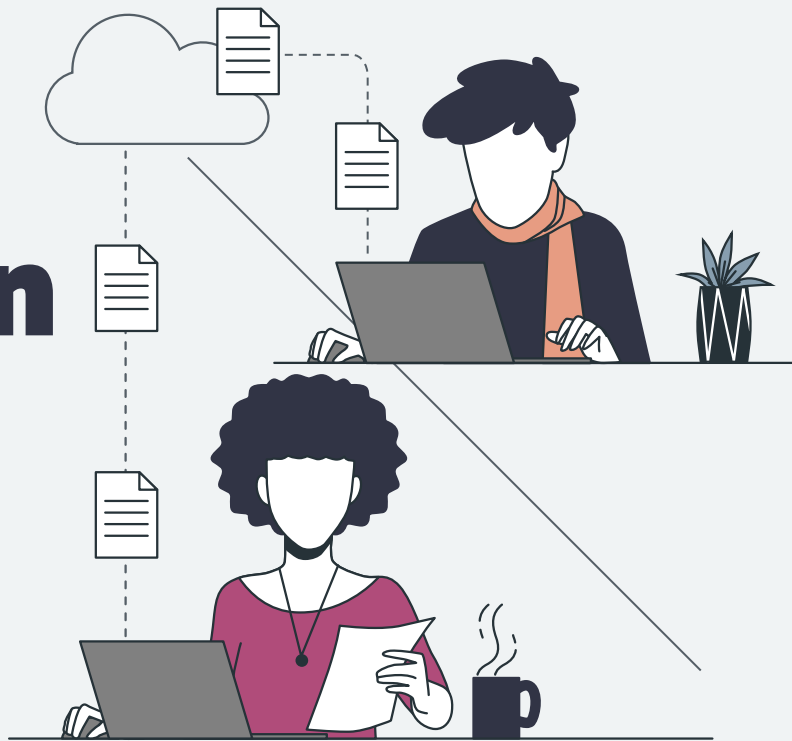
Comandos

- `git clone`
- `git init`
- `git remote`
- `git config --global user.name userName`
- `git config --global user.email userEmail`
- `git config --global --unset var`
- `git status`
- `git add .`
- `git add -u`
- `git commit -m "mensaje"`
- `git commit`
- `git push`
- `git pull`

01

Introducción a SCV

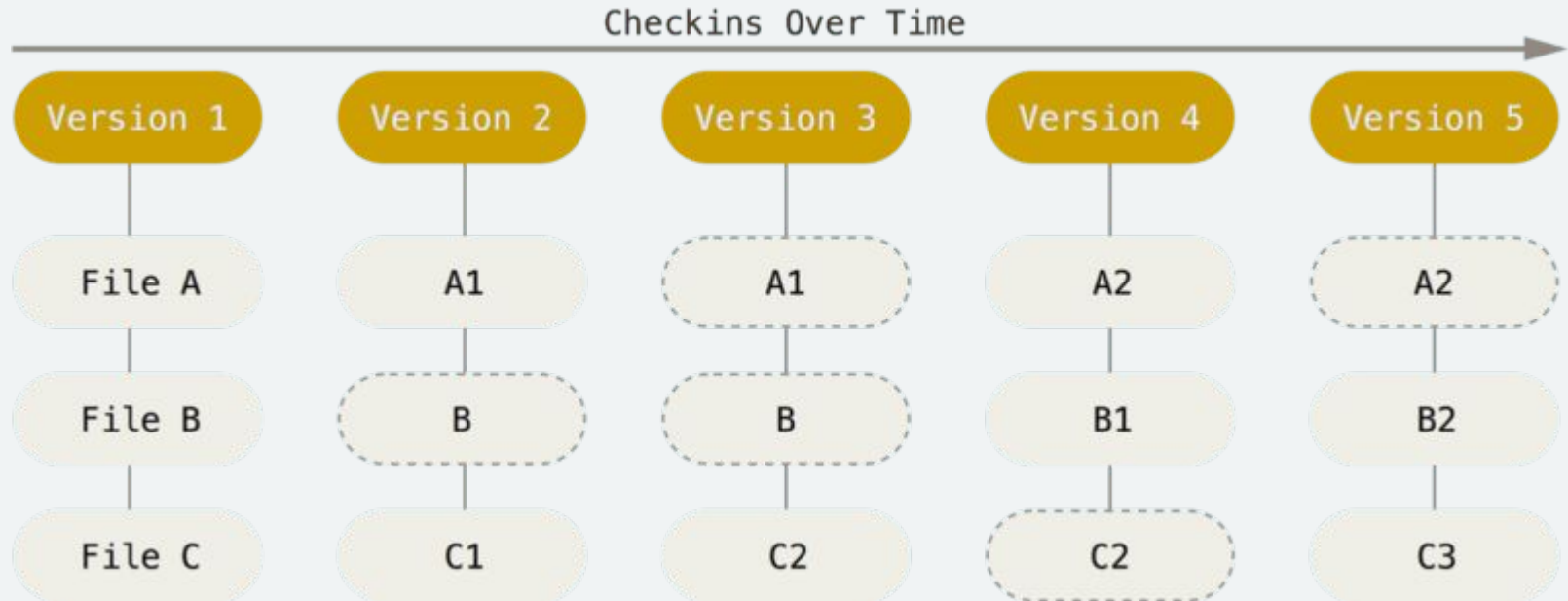
La base de todo para entender
Git y un poco de cultura
general.



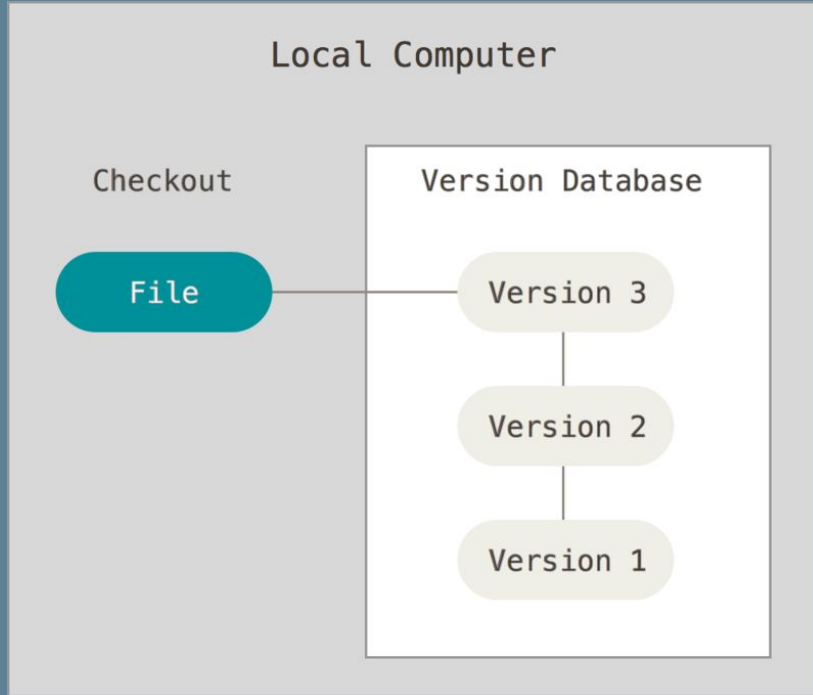


Un **SISTEMA DE CONTROL DE VERSIONADO** es un software realiza el seguimiento de los cambios de un conjunto de archivos. Básicamente lo que hace es construir una línea de tiempo que permite recuperar estas versiones anteriores de estos archivos en cualquier momento.

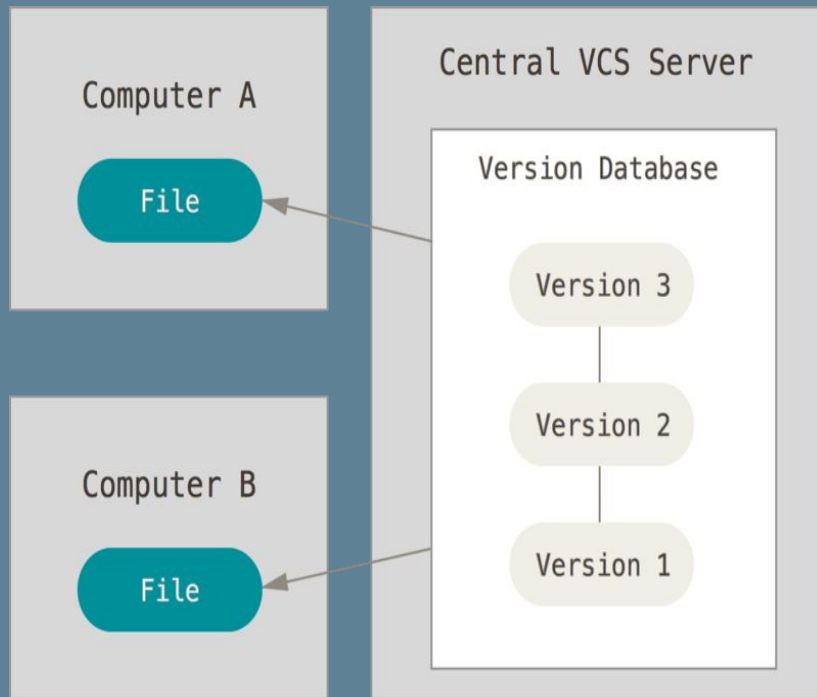
Un ejemplo de esa línea de tiempo



Locales



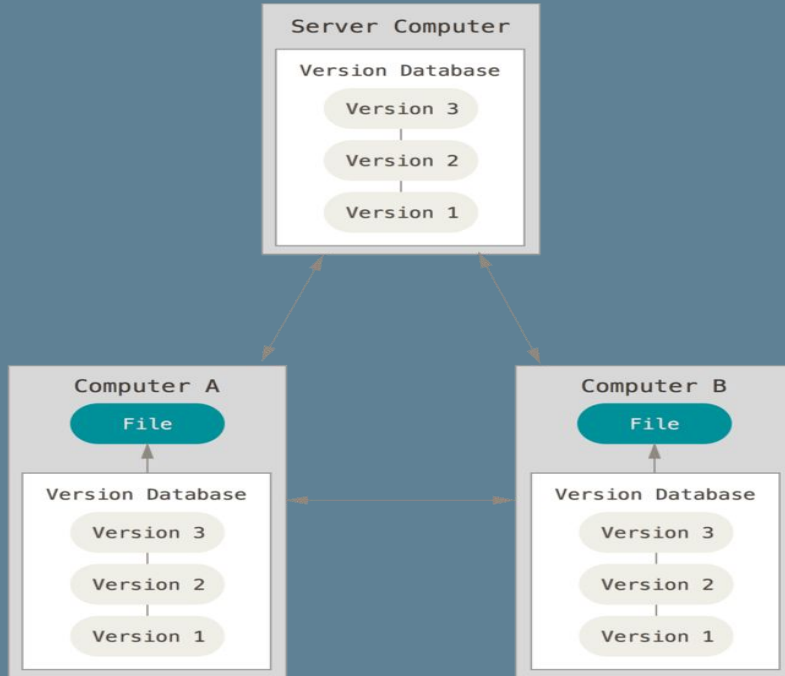
- ✓ Disponibilidad 100% (No depende de internet)
- ✓ Es muy sencillo de utilizar.
- ✗ Sin tolerancia a fallos, no existe backup.
- ✗ No permite el trabajo colaborativo.



Centralizados

- ✓ Permite el trabajo colaborativo.
- ✓ Es muy sencillo de utilizar.
- ✗ Sin tolerancia a fallos, no existe backup.
- ✗ Depende de internet para poder trabajar.
- ✗ Depende de internet para poder trabajar.

Distribuido



Permite el trabajo colaborativo.



100% de disponibilidad. Trabajar en la versión local y luego actualizas la centralizada cuando tengas internet.



Tolerancia a fallos. Existe multiples copias.

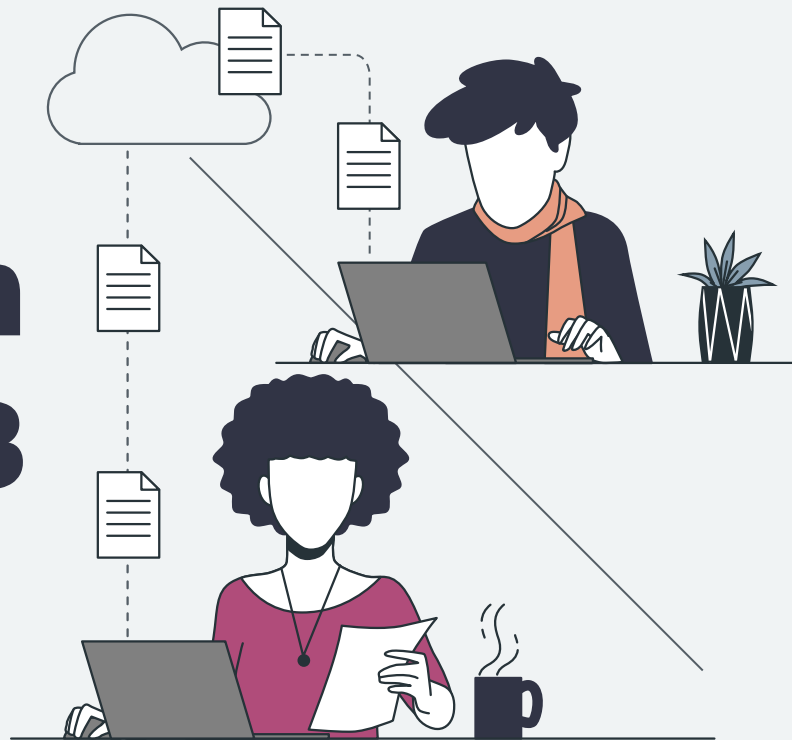


No es tan sencillo de utilizar.

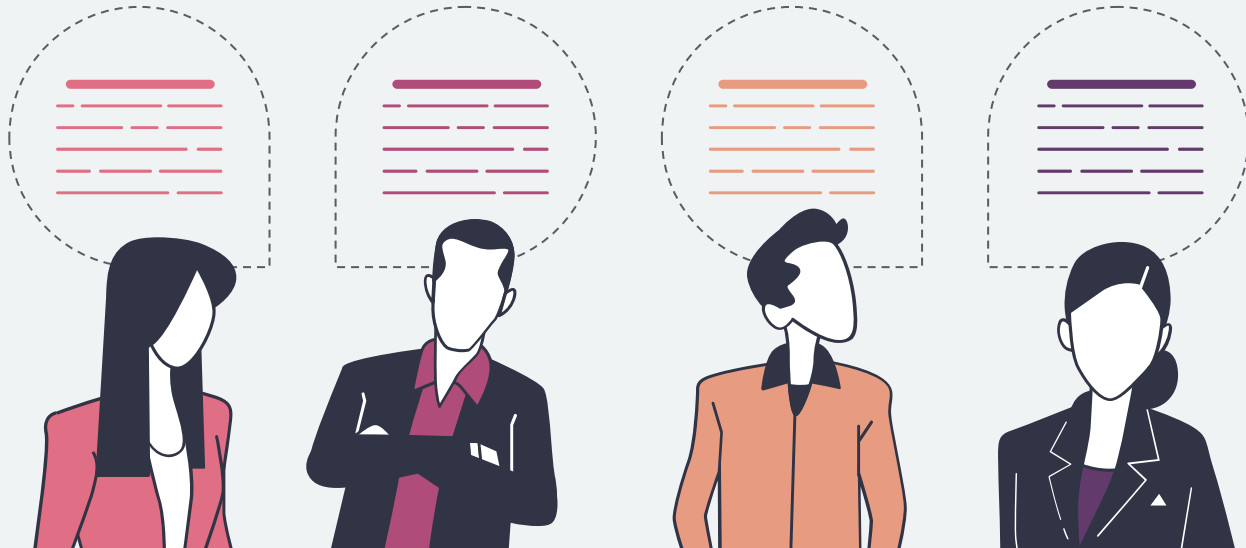
02

Introduccion a GIT/GITHUB

Ya sabemos lo básico,
podemos empezar con git.



Git, ¿De que tipo es?





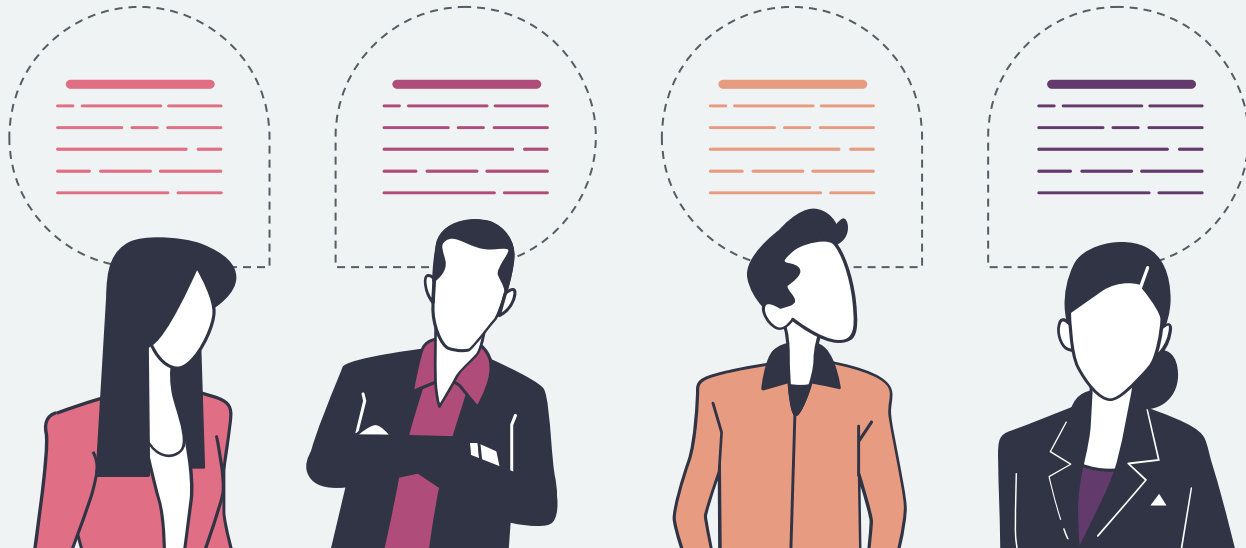
git

**Es un sistema de control
de versionado DISTRIBUIDO**

**Github es un servidor que
utiliza Git, y se utiliza
como la versión centralizada
de git.**

GitHub

**pero Github tiene un
par de funcionalidades
más.**



**Es como una red
social para
programadores,
pero en vez de
publicar lo que
comes...**




```

    function(b, c) {
      var d = a(c.form.querySelectorAll("input[type=checkbox])[name=" + b.el.name + "]"[0]);
      if (0 == d.index(b.el)) {
        var e = d.filter(":checked").length;
        return e >= b.arg || g.minChecked.replace("[count]", b.arg)
      }
    },
    maxSelected: function(a) {
      return null != a.val ? a.val.length <= a.arg || g.maxSelected.replace("[count]", a.arg) : null;
    },
    minSelected: function(a) {
      return null != a.val && a.val.length >= a.arg || g.minSelected.replace("[count]", a.arg)
    },
    radio: function(b) {
      var c = a(this.form.querySelectorAll("input[type=radio])[name=" + b.name + "]"[0]).filter(":checked").length;
      return 1 == c
    },
    custom: function(a, b) {
      var c = b.options.custom[a.arg],
          d = new RegExp(c.pattern);
      return d.test(a.val) || c.errorMessage
    },
    remote: function(a) {
      a.remote = a.arg
    }
  };
  b = function(b, c) {
    this.handler = !1, this.options = a.extend(10, {}, b, c), this.form = b, this.el = c, this.validate()
  }, b.prototype = {
    constructor: b,

```



**¿Existen otras
alternativas a
Github?**

Alternativa a github como version centralizada

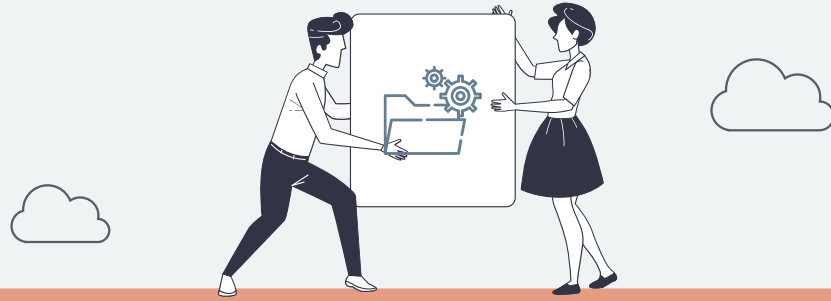


(<https://gitlab.com/gitlab-org>)

Alternativa a git como sistema de control de versionado



Es de Microsoft y normalmente es parte
de sus paquete



Un **REPOSITORIO** es la carpeta donde se realiza el seguimiento o no de los cambios realizados sobre un conjunto de archivos.

Ese repositorio puede encontrarse en tu computadora o en Github.



Actividad #1



Crearse un usuario de Github, y si ya lo tienes, recuperar el acceso a la cuenta. Necesitar un mail y un nombre de usuario para registrarte en Github, pero ten en cuenta;

- Si la cuenta es para Trabajo, elegir un nombre profesional. Un nombre fácil de reconocer que es de tu propiedad. Por ejemplo; GomezA.
- Si la cuenta es para uso personal, y no quieres ser relacionada con está. Utilizar un nickname para evitar ser relacionado. Por ejemplo; GameOver234.

Igualmente la elección del nombre es algo que el alumno puede hacer después, porque los nombre de usuario en Github son editables. También, una misma cuenta de Github puede tener varios mails relacionados.

Existen varios tipos de repositorios

Gratuitos y Públicos

Su contenido es público a toda la comunidad.

Gratuitos y Privados

Su contenido solo es visible para los colaboradores, máximo 3.

Pagos y Privados

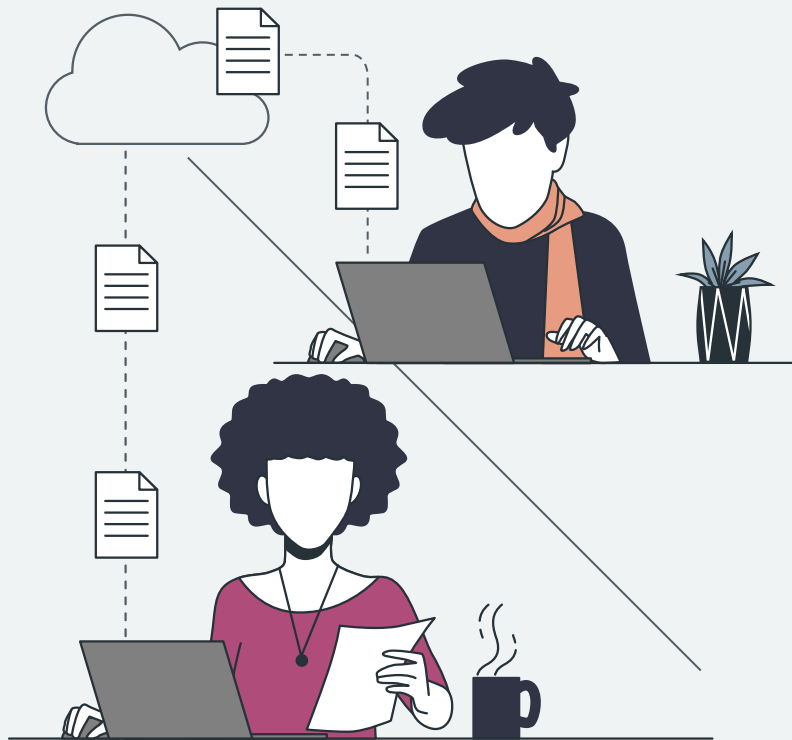
Su contenido solo visible para los colaboradores, el número depende del costo.



03

Comandos y GUI

Empecemos a utilizar la consola de comandos.



Existe una gran variedad de GUI para GIT



1

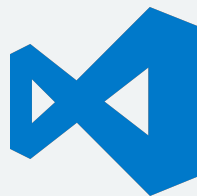
GitKraken

Multiplatafor
ma

2

**Source
tree**

Windows y
Mac



3

**Visual
studio
code**

4

**Git
tiene su
propio
GUI
MUY FEO**

Las empresas aman este tipo de software porque acelera mucho el tiempo de capacitación.



Configurando el espacio de trabajo

Lo primero antes de trabajar
con cualquier tecnología es
configurarla.

GIT funciona según el contenido de un archivo `.gitconfig`

SYSTEM



NO TOCAR. Este `.gitconfig` contiene información delicada, y no se debe modificar. NO LO MODIFIQUEN. Solo existe uno.

GLOBAL



El contenido de este archivo `.gitconfig` abarca todos los repositorios de un usuario de la computadora. Es una buena práctica configurar a este nivel. Solo existe uno.

LOCAL



Solo afecta un repositorio, justamente dentro del que se encuentra. Existe uno en cada repositorio.

Configurando GIT

0

Abrir la terminal en cualquier parte.

1

Tener a mano el nombre de usuario y email con los que te registraste en GITHUB.

2

Ejecutar los siguiente comandos.

```
git config --global user.name <user_name>
```

```
git config --global user.email <user_email>
```

3

Si escribiste algo mal, puedo borrarlo.

```
git config --global -unset <variable_name>
```





Creando mi primer repositorio

Necesitamos un repositorio
para trabajar.

Existen dos formas de crear un repositorio

**Desde Github
y luego
clonarlo.**

El más sencillo y ante natural.

**Crearlo
localmente y
luego subirlo.**

El más natural, pero un poco más complicado.

Creando un repositorio

0

Abrir la terminal en la ubicación donde se desea crear el repositorio.

1

Ejecutar el comando git init con el argumento según corresponda.

```
git init .
```

Si deseamos convertir la carpeta donde nos encontramos en repositorio.

```
git init <name>
```

Si deseamos crear una carpeta nueva y convertirla en repositorio con ese nombre



Creando un repositorio

¿Como estamos seguros de que se creó un repositorio?

Si habilitas ver los archivos ocultos en dentro de la carpeta que convertiste en repositorio. Deberías encontrar una carpeta `.git`, donde Git va guardar todo la información de tu proyecto. NO LA BORRES, NI MODIFIQUES.

Los hacemos por única vez para entender cómo funciona git.



Creando un repositorio

2

Debemos ir a Github y crear un repositorio.

3

Ahora debemos relacionar ambos repositorios.

```
git remote <url>
```





Estados de GIT

Images reveal large amounts of data, so remember: use an image instead of long texts

LOCAL EN TU COMPUTADORA

WORKING

TRACKING

UNTRAKING

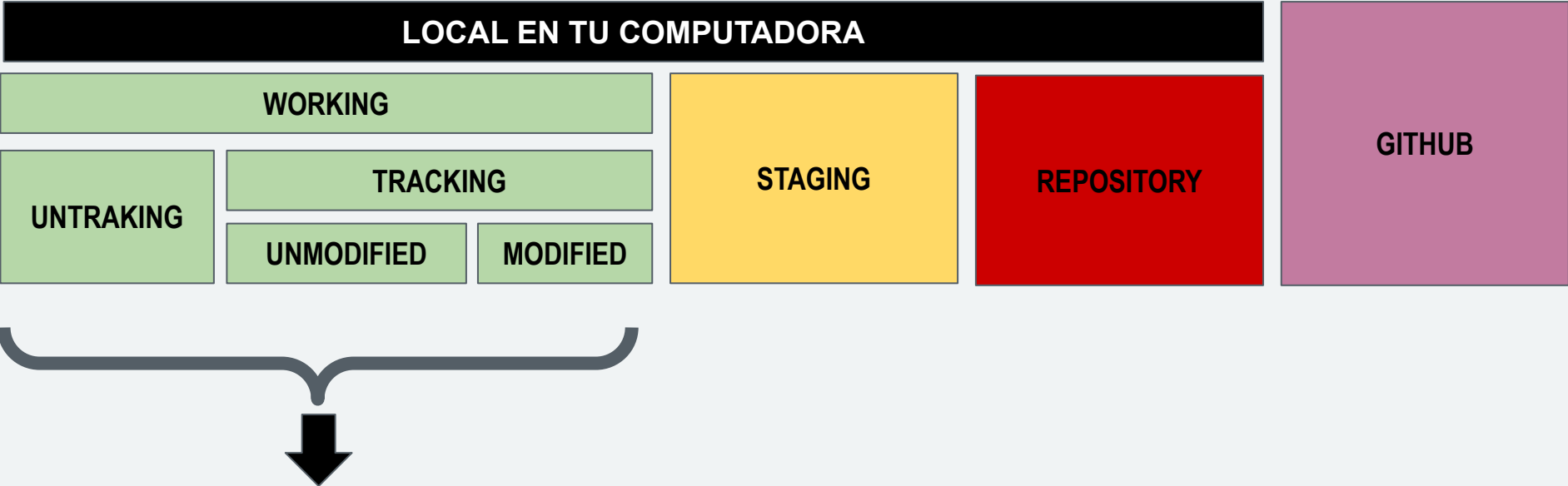
UNMODIFIED

MODIFIED

STAGING

REPOSITORY

GITHUB



Son todos los archivos con los que estamos trabajando. Son los cambios que más adelante convertiremos en una nueva versión de mi línea de tiempo.

LOCAL EN TU COMPUTADORA

WORKING

TRACKING

UNTRAKING

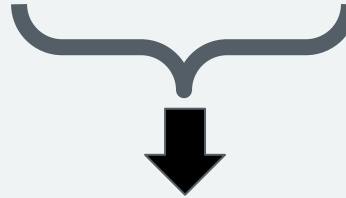
UNMODIFIED

MODIFIED

STAGING

REPOSITORY

GITHUB



Son los elegidos previamente de WORKIN para que formen parte de esa nueva versión que estamos por crear.

LOCAL EN TU COMPUTADORA

WORKING

TRACKING

UNTRAKING

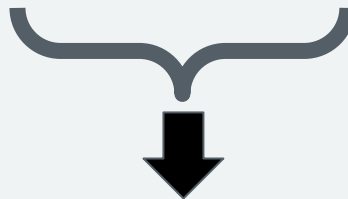
UNMODIFIED

MODIFIED

STAGING

REPOSITORY

GITHUB



Cuando pasamos los archivos de STAGING a REPOSITORY, creamos una nueva versión en nuestra línea de tiempo, con un número de identificación único, y acompañado de un comentario que detalla qué cambios se introdujeron.

LOCAL EN TU COMPUTADORA

WORKING

TRACKING

UNTRAKING

UNMODIFIED

MODIFIED

STAGING

REPOSITORY

GITHUB

Todo lo que ocurra de este lado, se puede borrar sin dejar rastros

Lo subido a la nube, no se puede borrar, solo puedes compensarlo.

LOCAL EN TU COMPUTADORA

WORKING

TRACKING

UNTRAKING

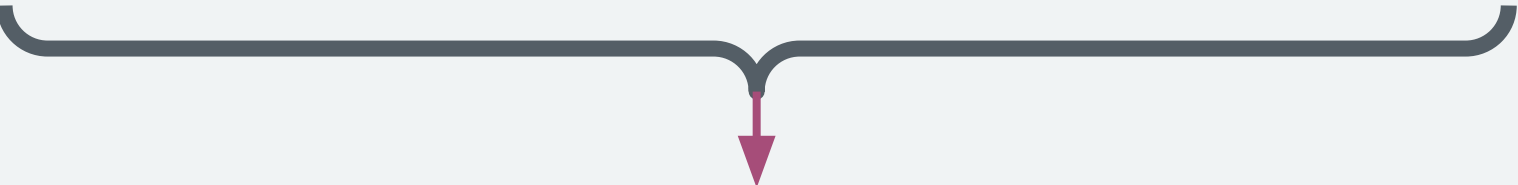
UNMODIFIED

MODIFIED

STAGING

REPOSITORY

GITHUB



Cualquier conflicto se
resuelve a este nivel. No en
la nube.

LOCAL EN TU COMPUTADORA

WORKING

TRACKING

UNTRAKING

UNMODIFIED

MODIFIED

STAGING

REPOSITORY

GITHUB

git clone <url>



LOCAL EN TU COMPUTADORA

WORKING

TRACKING

UNTRAKING

UNMODIFIED

MODIFIED

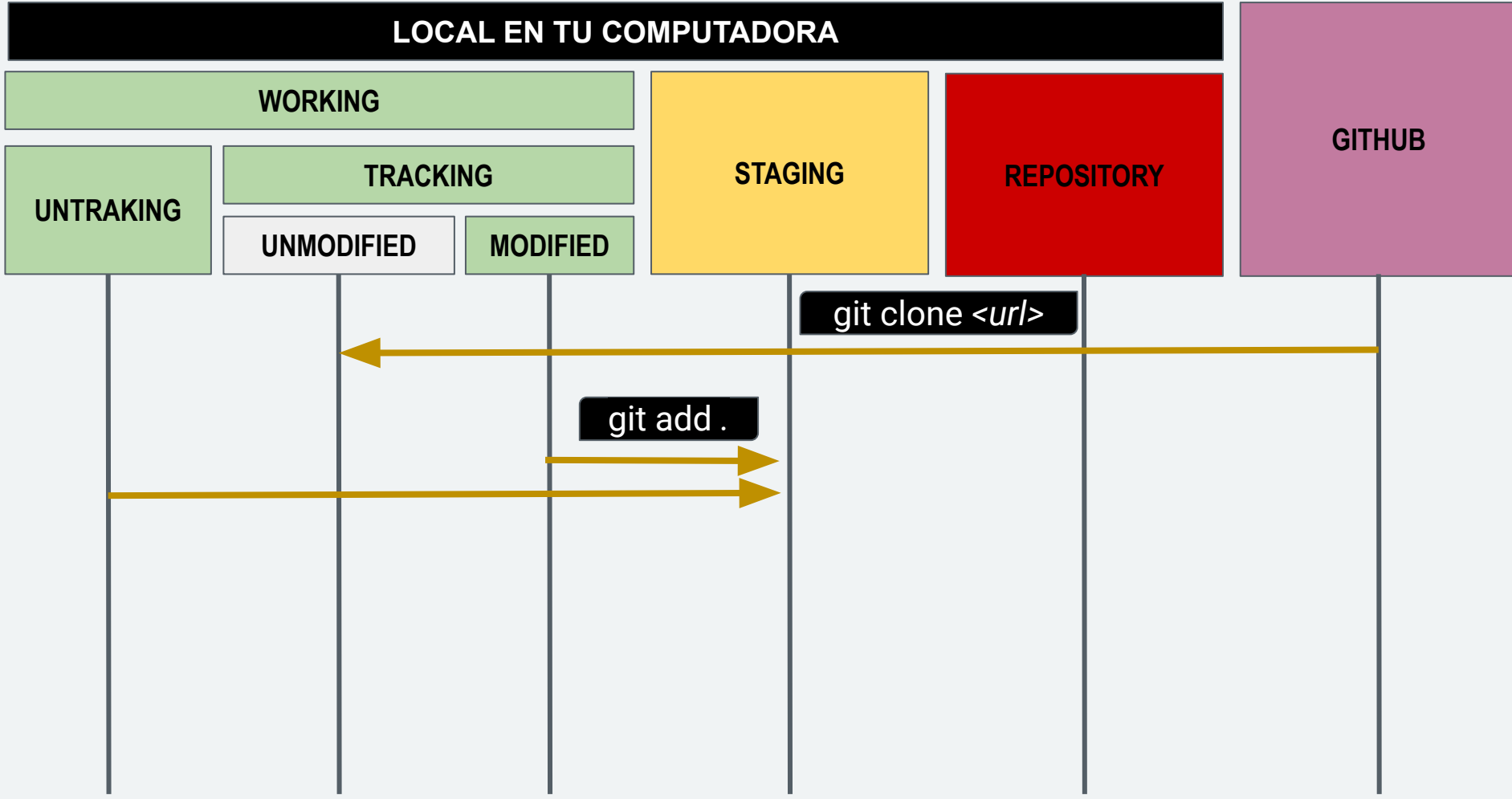
STAGING

REPOSITORY

GITHUB

git clone <url>

git add .



LOCAL EN TU COMPUTADORA

WORKING

TRACKING

UNTRAKING

UNMODIFIED

MODIFIED

STAGING

REPOSITORY

GITHUB

git clone <url>

git add .

git status

LOCAL EN TU COMPUTADORA

WORKING

UNTRAKING

TRACKING

UNMODIFIED

MODIFIED

STAGING

REPOSITORY

GITHUB

`git clone <url>`

`git add .`

`git status`

`git commit -m
"mensaje"`

LOCAL EN TU COMPUTADORA

WORKING

TRACKING

UNTRAKING

UNMODIFIED

MODIFIED

STAGING

REPOSITORY

GITHUB

git clone <url>

git add .

git status

git commit -m
"mensaje"

git log -l

LOCAL EN TU COMPUTADORA

WORKING

TRACKING

UNTRAKING

UNMODIFIED

MODIFIED

STAGING

REPOSITORY

GITHUB

git clone <url>

git add .

git status

git commit -m
"mensaje"

git log -l

git push

LOCAL EN TU COMPUTADORA

WORKING

UNTRAKING

TRACKING

UNMODIFIED

MODIFIED

STAGING

REPOSITORY

GITHUB

git pull

git add .

git status

git commit -m
"mensaje"

git log -l

git push

Thanks !

Do you have any questions?

rapa.matias.e@gmail.com

Aclarar en el mail el curso, día, turno y si corresponde, si es que está terminado el mismo. NO OLVIDAR ADJUNTAR EL ARCHIVO CON EL QUE ESTÁS TENIENDO PROBLEMAS

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik** and illustrations by **Stories**

