



Terminal, Git, GitHub y GitHub Pages

Introducción a Git

Crear un repositorio remoto en Github para controlar las versiones de un proyecto y publicar la página web utilizando Github Pages.

- Unidad 1: HTML y CSS
- Unidad 2: Bootstrap
- Unidad 3: JavaScript
- Unidad 4: Terminal, Git, GitHub y GitHub Pages



Te encuentras aquí



¿Qué aprenderás en esta sesión?

- *Utiliza las etapas del versionamiento de GIT, para mantener un repositorio de versiones.*

¿Con qué comando
puedo saber en qué
directorio me encuentro?



¿Qué comando permite crear directorios?



¿Cuál es la precaución que
debemos tener al usar el
comando rm?

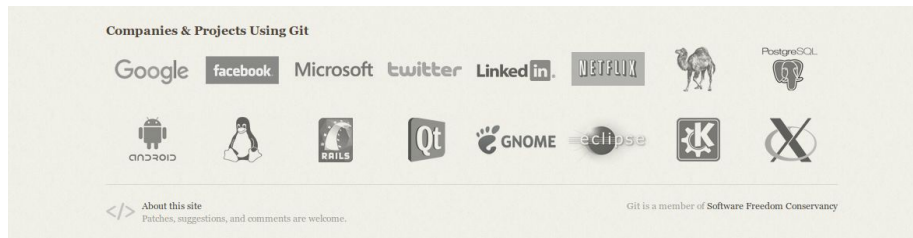


`/* Introducción a git */`

Introducción a Git

¿Qué es?

Git es un sistema de control de versiones gratuito, muy útil y ampliamente utilizado en el desarrollo.



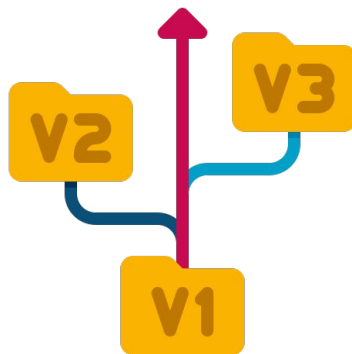
Permite:

- Recuperar versiones anteriores de nuestro código.
- Recuperar archivos borrados.
- Ayudar a gestionar cambios realizados por otras personas.
- Administrar un proyecto donde trabajan múltiples desarrolladores.

Introducción a Git

Control de versiones

- Para entender mejor qué es un sistema de control de versiones, imaginemos un editor de documento de texto como el proporcionado por Google, en el cual vamos añadiendo cambios y guardándolos. Si cerramos el programa solo tendremos los últimos cambios guardados. Utilizando git tendríamos acceso a todas las versiones guardadas, permitiendo incluso volver a una de ellas.



Introducción a Git

¿Cuándo debemos usar git?

- La recomendación es usarlo siempre que trabajemos desarrollando código (o con documento en texto plano), ya que nos evitará realizar trabajo extra si ocurre algún problema.
- Git también nos ayudará a hacer cambios en el sitio de forma ordenada sin poner en riesgo lo que ya está funcionando.



Introducción a Git

Formas de uso de git

- Existen distintas formas de trabajar con git. Algunos editores de texto traen incorporado formas automatizadas para usarlo, por ejemplo en Atom.
- También existen programas con interfaces gráficas como gitkraken o git Tower.
- Nosotros lo utilizaremos en nuestra terminal. Esto puede parecer a primera vista un poco más difícil, pero nos ayudará a entender mejor los conceptos más importantes.

/* Instalando y configurando Git */

Instalando Git

Verificando si se encuentra instalado

- El primer paso es verificar si ya tenemos instalado git en nuestro sistema. Esto lo podemos realizar escribiendo el comando

```
git --version
```

- En computadores con OSX es decir, computadores Mac, git viene instalado por defecto. Pero si por algún motivo no lo tienes, sigue los siguientes pasos:
 1. Entra al sitio de git.
 2. Descarga el archivo para OSX.
 3. Ejecuta el archivo descargado y sigue los pasos del instalador.

Instalando Git

En Linux

- En Linux, debemos utilizar el siguiente comando en la terminal.

```
sudo apt-get install git
```

Instalando Git

En Windows

- En Windows, si seguiste las instrucciones proporcionadas en la lectura para instalar el terminal, no deberías tener problemas. Pero, si por algún motivo no lo tienes, sigue los siguientes pasos:
1. Entra al sitio git/win.
 2. Descarga el archivo dependiendo de tu versión del sistema operativo.
 3. Ejecuta el archivo descargado y sigue los pasos del instalador.

Configurando Git

- Ahora que ya tenemos git instalado, nuestro siguiente paso será configurarlo en nuestro equipo.
- Principalmente, lo que tenemos que configurar es nuestro usuario en git.
- Para ello, utilizaremos los siguientes comandos:

```
git config --global user.name "Tu  
Nombre"
```

```
git config --global user.email  
tucorreo@mail.com
```


Configurando Git

- Una vez ingresados los comandos no veremos ninguna confirmación de la acción entonces puedes usar el comando.

```
git config --list
```

```
user.name=Nombre Apellido
```

Si ves este mensaje es porque lo lograste.

```
user.email=micorreo@mail.com
```

/* Uso básico de git */

Uso básico de git

Inicializando git

- Nuestro primer paso será escribir en la carpeta de proyecto lo siguiente

```
git init
```

- Todo lo que hace git lo realiza dentro de una carpeta oculta dentro del lugar donde fue inicializado. Si mostramos todos los archivos con **ls -a** podremos ver la carpeta **.git**. Todo ocurre de forma automática en el interior de este directorio.
- Es importante saber que la ejecución del comando git init solo lo debemos realizar una vez por proyecto.

Uso básico de git

Usando git



Uso básico de git

git add

- Agregamos nuestros archivos creados y cambios realizados utilizando un comando llamado git add seleccionando uno o varios archivos. Si queremos seleccionar todos los cambios debemos escribir:

```
git add --all
```

o

```
git add .
```

Uso básico de git

git commit

- Luego, debemos confirmar estos cambios, que equivale a cerrar la caja y agregarle una etiqueta con una descripción. Esto se logra con:

```
git commit -m "Nombre o descripción del commit"
```

- Es importante que la descripción del commit sea, valga la redundancia, descriptiva. Eso es para encontrar e identificar de manera más fácil las versiones de nuestro proyecto.

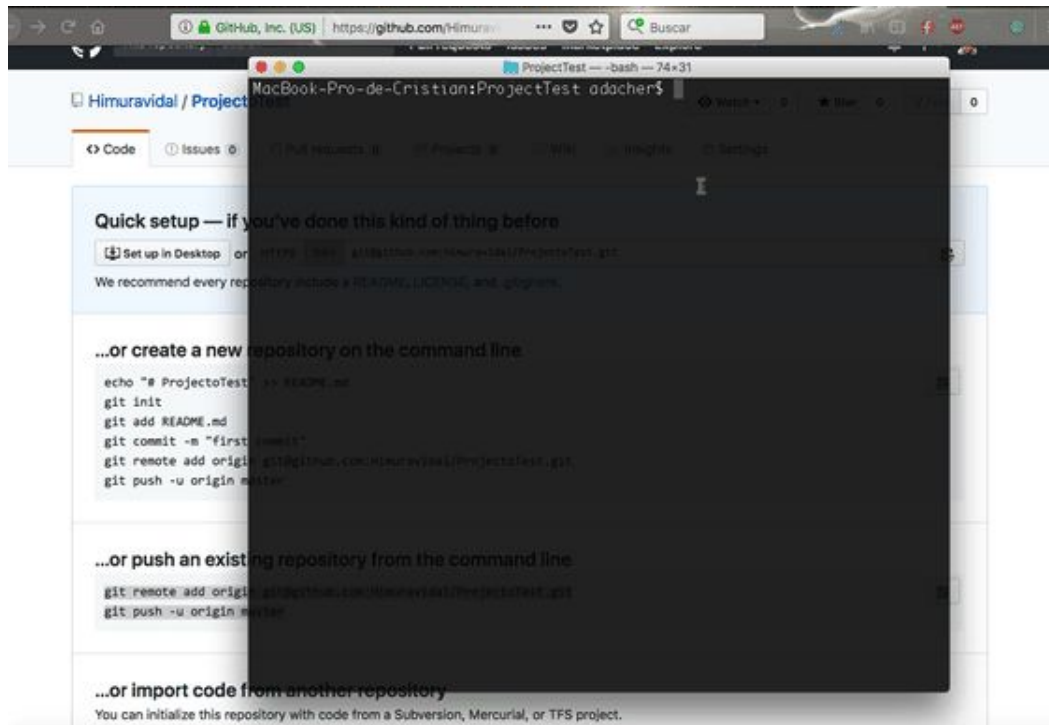
Uso básico de git

git push

- El último paso del flujo consiste en enviar la caja a destino. Esto se hace vía comando:

```
git push
```

{desafío}
latam_



/* Local o remoto */

¿Local o remoto?

- Agregar `git add` y confirmar `git commit` sucede completamente dentro de nuestro computador, en el envío `git push` se usan lugares de destino. Esto lo aprenderemos en el capítulo de GitHub.
- En resumen, el uso típico que haremos de git será `git init` para iniciar git en un proyecto y luego, por cada conjunto de cambios significativos: `git add`, `git commit` y `git push`.
- A cada conjunto de cambios commiteados le llamaremos versión.

/* Subiendo una nueva versión */

Subiendo una nueva versión

- Podemos revisar todas las versiones de un proyecto con:

```
git log
```

/* Gestionando los cambios */

Gestionando los cambios

- Si queremos ver los cambios introducidos en la consola, podemos usar el siguiente comando.

```
git diff
```

- **git diff** nos muestra toda la diferencia desde el último commit guardado.

Uso básico de git

Gestionando los cambios

- Para ver un resumen de qué archivos se han modificado podemos utilizar el siguiente comando:

```
git status
```

¿Existe algún concepto que
no hayas comprendido?





Próxima sesión...

- *Revisar material de estudio sincrónico que consiste en una **Guía de ejercicios** para practicar los conceptos aprendidos en todas las sesiones anteriores.*

{desafío}
latam_

*Academia de
talentos digitales*

