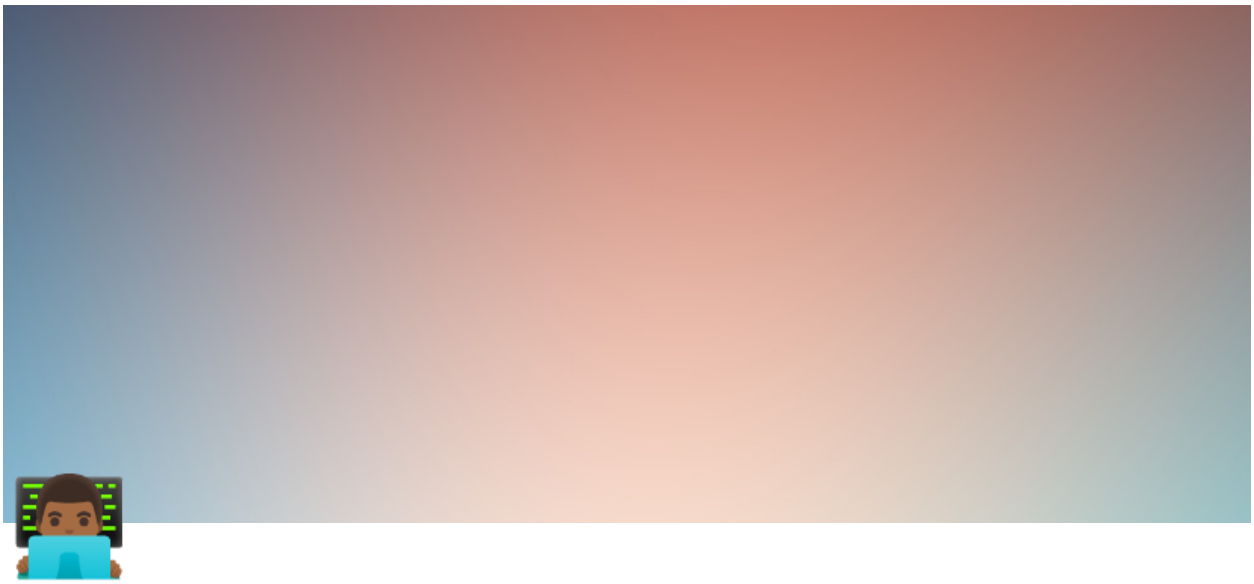

Curso de Git y GitHub - Parte 1

Software a utilizar

- VS Code: https://code.visualstudio.com/download
- Git: https://git-scm.com/downloads
- GitHub: https://github.com/

Introducción

Git es un sistema de control de versiones utilizado por muchos desarrolladores para gestionar y realizar seguimiento a los cambios que se hacen en el código de un proyecto. Es una herramienta útil para trabajar en colaboración y evitar la pérdida de información. En este artículo vamos a hablar sobre Git Español, una versión en español de Git.

Configuración global

Luego de haber instalado **git** en nuestro sistema, lo primero que debemos establecer el correo y el nombre de usuario de quien está trabajando en la máquina

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

Comandos básicos de git

Una vez que hemos creado el proyecto o folder en el que trabajaremos nuestro proyecto, debemos inicializar el repositorio

```
git init
```

A continuación agregaremos a nuestro proyecto el siguiente archivo index.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <meta http-equiv='X-UA-Compatible' content='IE=edge'>
    <title>Page Title</title>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <link rel='stylesheet' type='text/css' media='screen' href='main.css'>
    <script src='main.js'></script>
</head>
<body>
    <h1>Este es el primer ejemplo</h1>
</body>
</html>
```

En cualquier momento en el que estemos trabajando en nuestro proyecto, podemos revisar el estado de los archivos que tenemos. Los que aparezcan en rojo son los que no están siendo seguidos

```
git status
```

Agregamos los archivos para hacer seguimiento

```
git add .
```

Realizamos un commit para almacenar los cambios

```
git commit -m "Este es mi primer commit"
```

```
```

Siempre que deseamos ir guardando cambios deberemos hacer **git add .** y luego **git commit**

Conectándonos con GitHub

Una vez que tenemos hecho nuestro primer commit, es hora de conectar nuestro proyecto con GitHub, para ello, deberemos crear una cuenta en GitHub. Luego de ello, deberemos crear un proyecto, público y vacío. Esto nos dará el comando necesario para enlazar el proyecto recién creado en GitHub con nuestro proyecto.

```
git remote add origin https://github.com/denisjevct/FirstProject.git
git branch -M main
```

Una vez hecho esto, el proyecto está enlazado y cuando lo deseemos, podremos enviar los cambios al servidor remoto con el comando siguiente, el cual nos pedirá nuestros datos de GitHub:

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

Agregando un README.md

Los readme son un texto importante en todo proyecto de GitHub. Suelen estar escritos en lenguaje **markdown** y permiten describir de que trata el proyecto que estamos viendo. Para ello, crearemos en nuestro proyecto, un archivo llamado **README.md** con el siguiente contenido:

```
# Este es mi primer Readme

En un futuro iremos mejorando este readme
```

Para más detalles ver el enlace siguiente: https://docs.github.com/es/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/quickstart-for-writing-on-github

Una vez creado el archivo, deberemos guardar los cambios y actualizar el git local y el remoto:

```
git add .
git commit -m "ADD: README.md"
git push -u origin main
```

Si actualizamos ahora la página de github, veremos el contenido de nuestro readme.

Navegando entre commit

Lo primero que haremos es modificar el index.html para que el siguiente contenido en el body

```
<!DOCTYPE html>
<html>
......
<body>
    <h1>Este es el primer ejemplo</h1>

    <p>Ahora he agregado este nuevo texto para seguir mejorando el sitio</p>
</body>
</html>
```

Una vez hecho los cambios guardamos los cambios en local y remoto

```
git add .
git commit -m "CHANGE: Agregar nuevas líneas al body"
git push -u origin main
```

Para obtener la lista detallada de cada commit utilizamos el siguiente comando

```
git log
```

Para desplazarnos a alguno de los commit utilizamos el comando **git checkout <SHA1>** siendo esto último, dato obtenido con **git log**

```
git checkout e83aba0a3cee9992071ef0d506c09eccf8d83bec
```

Con esto podemos observar, el archivo index ha cambiado para mostrarnos como se encontraba en el momento del primer commit, así mismo podemos observar como el archivo README.md ha desaparecido.

Para regresar al momento actual del código hacemos el comando

```
git checkout main
```

Conclusión

Ahora sólo queda seguir ejercitando creando más archivos y modificándolos, haciendo commit y navegar entre ellos para quedar más claros de su funcionamiento.