

PRESENTACIÓN

Nombre: Marcos José Morillo Suarez

Matricula: 2020-10624

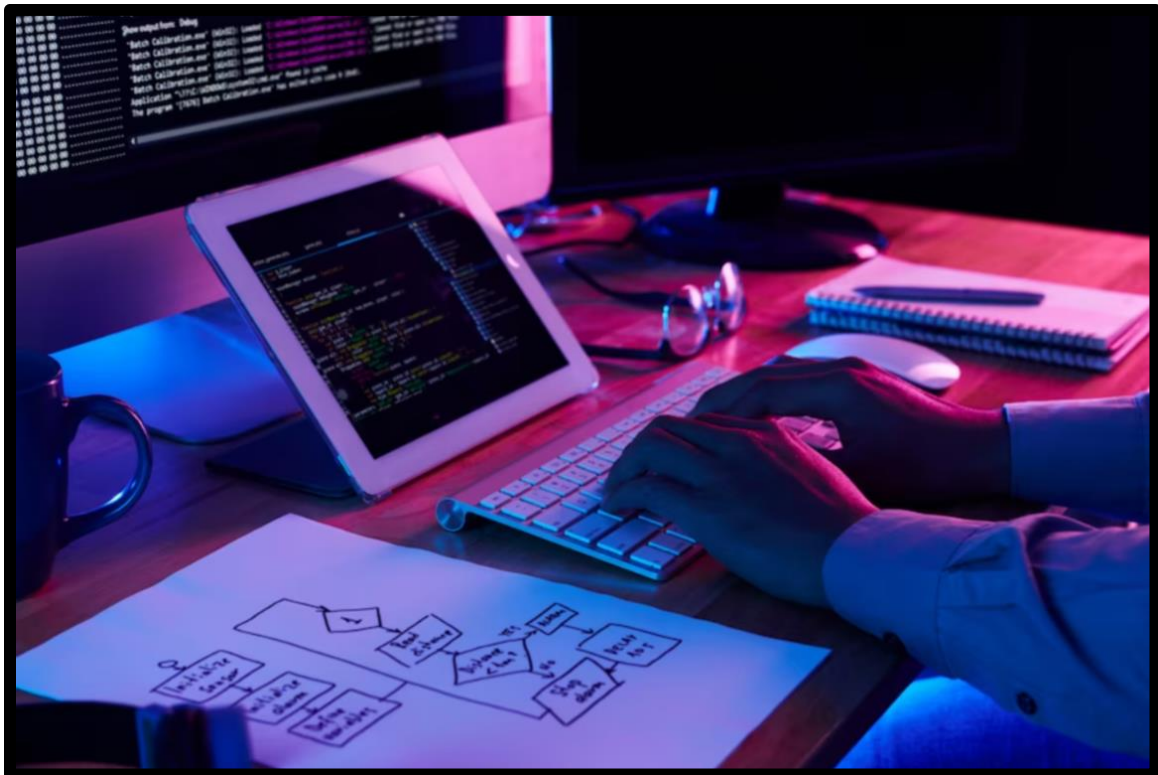
Materia: Programación III

Tema: Herramientas de administración de fuentes

Docente: Kelly Tejeda

Instituto: ITLA

Periodo: 2020-C-1



TEMA 1 - CUESTIONARIO: GIT



1. ¿Qué es Git?

Git es un sistema de control de versiones distribuido, lo que significa que permite a los desarrolladores rastrear los cambios en sus proyectos de software a lo largo del tiempo. Permite a múltiples personas colaborar en un proyecto al mismo tiempo, gestionando de manera eficiente las diferentes versiones del código y facilitando la fusión de los cambios realizados por diferentes colaboradores.

2. ¿Cuál es el propósito del comando git init en Git?

El comando git init se utiliza para inicializar un nuevo repositorio Git en un directorio existente o vacío. Al ejecutar este comando, se crea un nuevo subdirectorio llamado .git en el directorio actual. Este subdirectorio contiene todos los archivos necesarios para el repositorio Git, incluidos metadatos y la base de datos de objetos de Git.

3. ¿Qué representa una rama en Git y cómo se utiliza?

En Git, una rama es una línea independiente de desarrollo que representa una secuencia particular de commits. Las ramas permiten a los desarrolladores trabajar en funcionalidades separadas del proyecto sin afectar la rama principal (generalmente llamada main o master). Se utilizan para experimentar nuevas características, arreglar errores o trabajar en cualquier tarea sin interferir con el trabajo en otras ramas.

4. ¿Cómo puedo determinar en qué rama estoy actualmente en Git?

Puedes utilizar el comando git branch para ver una lista de todas las ramas disponibles en el repositorio local. La rama actual se resalta con un asterisco (*). También puedes utilizar el comando git status, que muestra información sobre la rama actual y el estado de los archivos en el directorio de trabajo.

5. ¿Quién es la persona responsable de la creación de Git y cuándo fue desarrollado?

Git fue creado por Linus Torvalds, el mismo creador del kernel de Linux, en 2005.

Torvalds desarrolló Git para ayudar en el desarrollo del kernel de Linux y superar las limitaciones de otros sistemas de control de versiones disponibles en ese momento.

6. ¿Cuáles son algunos de los comandos esenciales de Git y para qué se utilizan?

- **git clone:** clona un repositorio Git existente en un nuevo directorio local.
- **git add:** añade cambios al área de preparación, también conocida como el "staging area", para ser incluidos en el próximo commit.
- **git commit:** registra los cambios realizados en el repositorio localmente, creando un nuevo commit con un mensaje descriptivo.
- **git push:** envía los cambios locales a un repositorio remoto.
- **git pull:** obtiene los cambios del repositorio remoto y los fusiona con el repositorio local.
- **git checkout:** cambia entre ramas o restaura archivos desde un commit anterior.

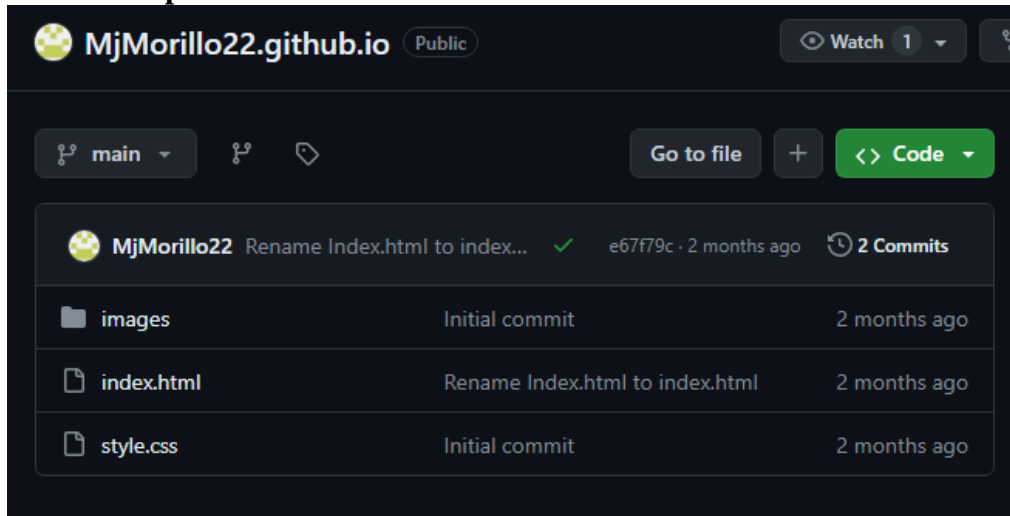
7. ¿Puedes mencionar algunos de los repositorios de Git más reconocidos y utilizados en la actualidad?

- GitHub
- GitLab
- Bitbucket

TEMA 2 - EJERCICIO PRACTICO: GIT



1. Crear el Repositorio en GitHub.



2. Clonar el Repositorio en tu Computadora.

```
Marcos Morillo@LAPTOP-LE7N07UA MINGW64 ~/Desktop/TAREA 3 (master)
$ git clone https://github.com/MjMorillo22/MjMorillo22.github.io.git
Cloning into 'MjMorillo22.github.io'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 9 (delta 0), reused 7 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), 256.55 KiB | 2.59 MiB/s, done.

Marcos Morillo@LAPTOP-LE7N07UA MINGW64 ~/Desktop/TAREA 3 (master)
$
```

3. Configurar las Ramas.

```
Marcos Morillo@LAPTOP-LE7N07UA MINGW64 ~/Desktop/TAREA 3 (master)
$ git checkout -b main
Switched to a new branch 'main'

Marcos Morillo@LAPTOP-LE7N07UA MINGW64 ~/Desktop/TAREA 3 (main)
$ git checkout -b dev
Switched to a new branch 'dev'

Marcos Morillo@LAPTOP-LE7N07UA MINGW64 ~/Desktop/TAREA 3 (dev)
$ git checkout -b QA
Switched to a new branch 'QA'
```

4. Crear Ramas para Funcionalidades y Correcciones.

```
Marcos Morillo@LAPTOP-LE7N07UA MINGW64 ~/Desktop/TAREA 3 (QA)
$ git checkout -b feature/nueva-caracteristica
Switched to a new branch 'feature/nueva-caracteristica'

Marcos Morillo@LAPTOP-LE7N07UA MINGW64 ~/Desktop/TAREA 3 (feature/nueva-caracteristica)
$ git checkout -b hotfix/error-critico
Switched to a new branch 'hotfix/error-critico'

Marcos Morillo@LAPTOP-LE7N07UA MINGW64 ~/Desktop/TAREA 3 (hotfix/error-critico)
$ git branch
  QA
  dev
  feature/nueva-caracteristica
* hotfix/error-critico
  main
  master
```

5. Agregar y Cometer Cambios.

```
Marcos Morillo@LAPTOP-LE7N07UA MINGW64 ~/Desktop/Tarea3 (main)
$ git add .

Marcos Morillo@LAPTOP-LE7N07UA MINGW64 ~/Desktop/Tarea3 (main)
$ git commit - "Commit inicial"
error: pathspec '-' did not match any file(s) known to git
error: pathspec 'Commit inicial' did not match any file(s) known to git

Marcos Morillo@LAPTOP-LE7N07UA MINGW64 ~/Desktop/Tarea3 (main)
$ git commit -m "Commit inicial"
[main (root-commit) 90b9828] Commit inicial
5 files changed, 110 insertions(+)
 create mode 100644 Desktop/Tarea3/images/canita.jpg
 create mode 100644 Desktop/Tarea3/images/no1a.jpg
 create mode 100644 Desktop/Tarea3/index.html
 create mode 100644 Desktop/Tarea3/miNombre.txt
 create mode 100644 Desktop/Tarea3/style.css
```