****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**INFORME**

**“INSTALACION Y CONFIGURACION DE GIT”**

Curso: Programación II

Docente: Ing. Enrique Lanchipa Valencia

***Ticona Chambi, Jhon Thomas (2018062232)***

**Tacna – Perú**

**2020**

INDICE

[I. INFORMACIÓN GENERAL 3](#_Toc394507353)

[- Objetivos 3](#_Toc394507354)

[- Equipos, materiales, programas y recursos utilizados 3](#_Toc394507355)

[II. MARCO TEORICO 3](#_Toc394507356)

[III. PROCEDIMIENTO 4](#_Toc394507357)

[IV. ANALISIS E INTERPRETACION DE RESULTADOS 10](#_Toc394507358)

[V. CUESTIONARIO **¡Error! Marcador no definido.**](#_Toc394507359)

[CONCLUSIONES 12](#_Toc394507360)

[RECOMENDACIONES 12](#_Toc394507361)

[BIBLIOGRAFIA **¡Error! Marcador no definido.**](#_Toc394507362)

[WEBGRAFIA 12](#_Toc394507363)

**INFORME**

**TEMA: INSTALACION DE GIT Y GIT HUB**

1. **INFORMACIÓN GENERAL**

* **Objetivos:**

Instalar un sistema de control de versiones y un gestor de proyectos

* Conocer la herramienta Git o un sistema de control de versiones
* Conocer algunos comandos básicos de Git

**Equipos, materiales, programas y recursos utilizados:**

* Computadora con sistema operativo Windows, Linux , Mac y Solaris
* Driver para GIT

1. **MARCO TEORICO**

* ¿Qué es Git?

Git es un sistema de control de versiones desarrollado por Linus Torvalds (el hombre que creó Linux).

* ¿Qué es Git Hub?

GitHub es un sistema de gestión de proyectos y control de versiones de código, así como una plataforma de red social diseñada para desarrolladores. ¿Pero para qué se usa GitHub? Bueno, en general, permite trabajar en colaboración con otras personas de todo el mundo, planificar proyectos y realizar un seguimiento del trabajo.

* ¿Qué es Hub?

El hub de GitHub es lo que convierte una línea de comandos, como Git, en la red social más grande para desarrolladores.

Además de contribuir a un determinado proyecto, GitHub le permite a los usuarios socializar con personas de ideas afines. Puedes seguir a las personas y ver qué hacen o con quién se conectan.

* ¿Qué es un sistema de control de versiones?

El sistema de control de versiones ayuda a registrar los cambios realizados al código. Aún más, registra quién realizó los cambios y puede restaurar el código borrado o modificado.

No hay códigos sobrescritos ya que Git guarda varias copias en el repositorio.

1. **PROCEDIMIENTO**

## **INSTALACIÓN DE GIT**

**Paso 1:** Descargar el software en la página oficial de Git.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente<https://git-scm.com/>

**Paso 2:** Una vez descargado el controlador procedemos a ejecutarlo.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**Paso 3:** Luego aparecerá una ventana pidiendo que leamos la información, luego le damos a Next.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**Paso 4:** Elegimos la ubicación de instalación de Git en nuestro ordenador.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**Paso 5:** Seleccionamos con componentes de deseamos instalar.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**Paso 6:** Seleccionamos el menú principal de nuestra carpeta

**Paso 7:** Elegimos el editor que deseamos utilizar en GIT.

**Paso 8:** Elegimos como usar la línea de comandosInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente.

1. Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

   Descripción generada automáticamenteSeleccionamos la opción recomendada -> Desde la línea de comandos y desde software de terceros..

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**Paso 9:** Elegimos la biblioteca que deseamos usar para GIT.

**Paso 10:** Configuramos las versiones de final de línea.

1. Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

   Descripción generada automáticamenteElegimos las la primera opción

**Paso 11:** Elegimos el emulador de terminal que usaremos con el GIT bash.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 12:** Elegimos la opción por defecto para el Git Pull.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 13:** Seleccionamos el asistente de credencial que deseamos configurar.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**Paso 14:** Habilitamos el almacenamiento en el cache del sistema de archivos

**Paso 15:** Elegimos si deseamos usar el soporte experimental para seudo consolas.

1. Luego le damos a instalar.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**Paso 16:** Una vez terminada la instalación le damos a next para finalizar.

## **CONFIGURACION INICIAL**

Texto

Descripción generada automáticamente**Paso 1:** Establecer un correo de usuario

**Paso 1:** Establecer un nombre de usuario.

Texto

Descripción generada automáticamente

**Paso 4:** Verificamos la información en la consola parámetro -e

Texto

Descripción generada automáticamente

Una vez que observamos que la información es correcta salimos usado la tecla

” Esc + : + q “

Captura de pantalla de un celular

Descripción generada automáticamente

**Paso 5:** Con el parámetro -l también podremos ver nuestra información

Texto

Descripción generada automáticamente

## **REPOSITORIO LOCAL**

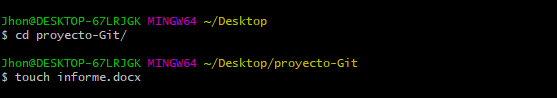
1. **Crear directorio**

**Podemos crear un directorio con el comando - mkdir “nombre del directorio”**

**Captura de pantalla de computadora

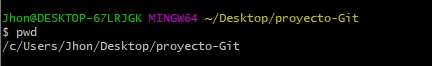
Descripción generada automáticamente**

1. **Crear archivo en directorio**
2. **Primero debemos acceder al directorio**
3. **Luego crear un archivo colocamos el comando - touch “el nombre del archivo”. “y la extensión”**



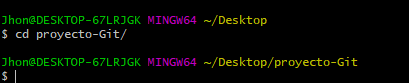
1. **Verificar ubicación actual**

**Colocamos el comando - pwd para saber la ubicación actual.**



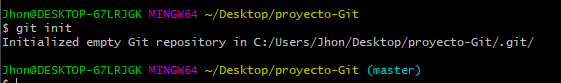
1. **Ubicarse en el directorio de trabajo**

**Podemos acceder a cualquier a los directorio con el comando - cd “nombre del directorio/” o colocar la primera letra y darle a la tecla TAB para autocompletar el nombre.**

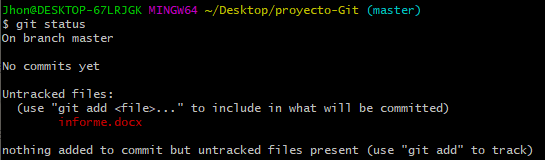


1. **Iniciar repositorio local**

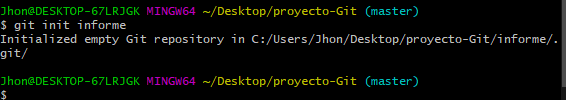
**Con el comando – git init podemos inicializar nuestro repositorio**



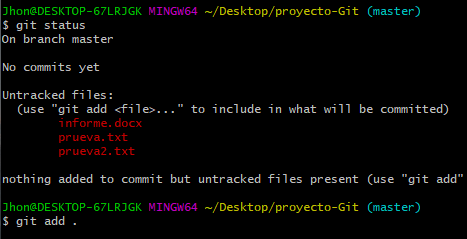
1. **Verificar el estado del repositorio y archivos**



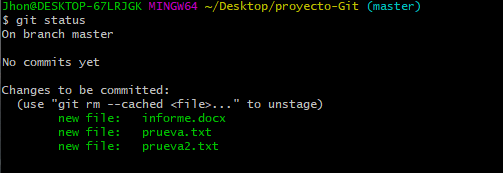
1. **Subir archivos individuales de la zona de trabajo a la zona de Stage**
   1. **Colocamos el comando git add y el nombre del achivo**



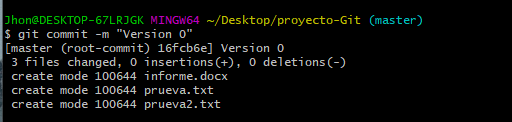
1. **Subir todos los archivos de la zona de trabajo a la zona stage**
   1. **Colocamos el comando git add .**



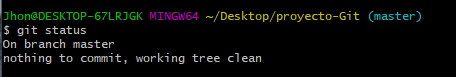
1. **Verificar el estado del repositorio y archivos**



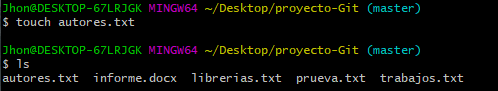
1. **Crear el primer commit**



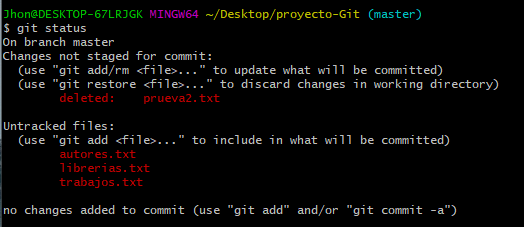
1. **Verificar el estado del repositorio y archivos**



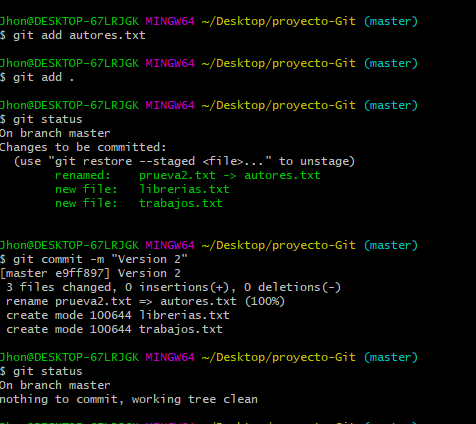
1. **Crear nuevos archivos y modificar algunos creados anteriormente**



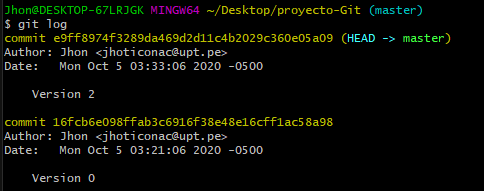
1. **Verificar el estado del repositorio**



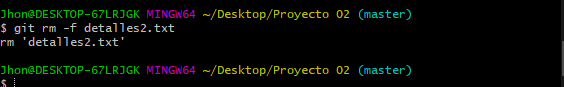
1. **Volver a realizar los pasos del 7 al 11**



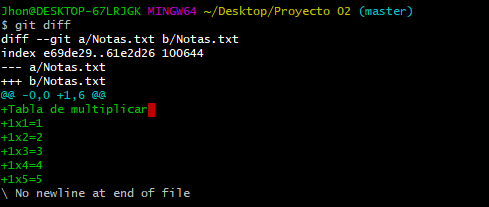
1. **Visualiza las anteriores versiones creadas**



1. **Modificar el contenido de archivos ya sea agregando o quitando información.**



1. **Visualizar cambios realizados con GIT DIFF**



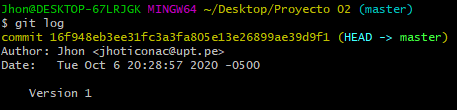
1. **Crear archivos README.md**



1. **Subir todos los archivos a la zona de STAGE.**



1. **Visualiza versiones realizadas**



1. **Visualizar archivos de la zona de STAGE al repositorio local con COMMIT**

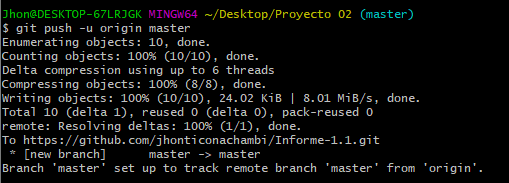
**Texto

Descripción generada automáticamente**

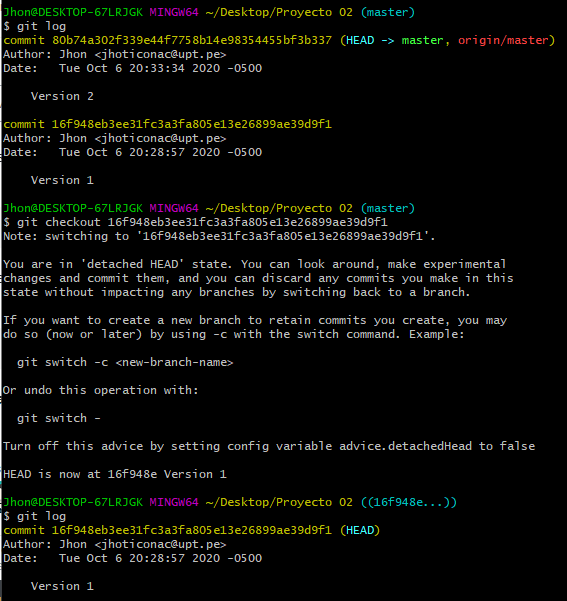
1. **Conectarse desde repositorio local GIT con repositorio remoto GITHUB**



1. **Subir del repositorio local al repositorio remoto**



1. **Restaurar a una de las versiones anteriores**



1. **ANALISIS E INTERPRETACION DE RESULTADOS**

Luego de lo observado logramos conocer algunos comandos básicos de GIT los cuales no será muy útiles para el control de las versiones de nuestros programas que desarrollemos.

Lo cual nos permitirá crear repositorios y mayor control de las modificaciones que se ejecuten.

# CONCLUSIONES

Podemos concluir que GITHUB es una herramienta imprescindible que nos ayudara a llevar e historial completo de modificaciones de nuestros proyectos y poder administrar nuestro código.

# RECOMENDACIONES

Tener en cuenta que debemos escribir bien los comandos que ejecutamos en el GIT BASH y tomar en cuenta los mensajes los cuales nos indican el error que hemos cometido.

# WEBGRAFIA

<https://docs.github.com/es/free-pro-team@latest/github/managing-files-in-a-repository/renaming-a-file-using-the-command-line>

<https://www.hostinger.es/tutoriales/comandos-de-git>