

# Taller de Herramientas Computacionales

Hector Chaparro Reza

Enero 14, 2019

*Este es la segunda bitácora del curso intersemestral de Taller de Herramientas Computacionales, con la fecha (08/01/2019). Aquí resumiré lo más relevante de la teoría. Los problemas y las soluciones que surgieron a lo largo de la práctica serán comentadas en el cuerpo de la bitácora.*

# Git

## *Github como herramienta de trabajo*

### *1. Instalación y uso*

#### *(a) Instalación*

*Para la instalación utilizamos el comando "sudo dnf install git" para fedora, lo que hace sudo es dar permiso de administrado e instalar el software en el sistema root. Git es un programa que permite subir archivos a la red de diferentes codigos fuente con eficiencia y confiabilidad. Posteriormente creamos nuestra cuenta en github.com que es una página para usar el software.*

#### *(b) Herramienta:*

##### *i. Creando un repositorio:*

*Al crear mi cuenta no tenía idea de como funcionaba github, en otras clases los profesores subían archivos ahí pero aqui me enseñaron como hacer eso utilizando la bash de Linux para esto creamos un directorio con el comando mkdir y para inicializarlo como un repositorio de git utilizamos el comando git config --global user.name "315264757" y git config --global user.email "hectorchaparro@ciencias.unam.mx" para especificar que queremos vincularlo con nuestra cuenta en el servidor de git y después usamos el comando git init el cual inicializa el repositorio creando un archivo .git dentro de el directorio (está es la diferencia entre un repositorio y un directorio) al hacer esto en clase, la terminal nos mandaba un error esto pasó por que a la hora de hacer la configuración de la cuenta escribimos user.mail y era email.*

##### *ii. Creando un archivo de texto plano:*

*Creamos un archivo de texto plano llamado README.md con el comando cat, dentro del archivo de texto en el bash usamos el editor vi donde podíamos escribir algo y luego lo terminamos con el comando shwift+: para poder usar el comando*

*wq que es write y quit para salvar lo que habíamos escrito.*

*iii. Subir archivos a la red:*

*Al finalizar la sesión aadimos los archivos con el comandp `git add *` y lo comentamos con `git commit`, esto nos llevo de nuevo al editor vi pero yo sentí que era mejor usar `git commit -m` para hacer un comentario rápido. Después usamos el comando `git push` que sube la información a internet y checamos los cambios con `git status` que muestra el estado del repositorio.*