

## **Maria Emilia Arzadum: Mochila 6**

### **CONSIGNA**

#### **Comandos básicos de GIT**

<b>git init</b>	inicializar un nuevo repositorio
<b>git status</b>	muestra en qué estado se encuentran los archivos
<b>untracked</b>	archivo son rastrear (no está en git)
<b>unstaged</b>	archivo en git pero no hicimos git add todavía
(git add)	
<b>staged</b>	archivo no guardado
(git commit)	
<b>tracked</b>	archivo rastreado. enviado al repositorio. los cambios son definitivos
<b>git add</b>	añade el archivo a staging
<b>git add nombre_del_archivo</b>	añade ese archivo
<b>git add.</b>	añade todos los archivos
<b>git commit</b>	guarda los archivos en el repositorio local
<b>git commit -m</b>	git commit + mensaje
<b>git reset head</b>	quita archivos de staged o los devuelve a su estado anterior
<b>git rm</b>	elimina el archivo de nuestro directorio de trabajo y no elimina su historial del sistema de versiones
<b>git rm --cached</b>	mueve los archivos que indiquemos al estado untracked
<b>git rm --force</b>	elimina los archivos de git y del disco duro

#### **RAMAS o BRANCH**

Los commits se aplican sobre una rama que por defecto es la rama master o main.  
Crear una rama consiste en copiar un commit de una rama determinada, pasándola a otra rama para así continuar el trabajo sin afectar el flujo de trabajo de la rama master o main.

La rama se va a crear desde el lugar que estoy ubicado:

<b>git status</b>	me muestra en que rama estoy, versión del commit y/o si hay algún commit que enviar.
<b>git branch nombre_rama</b>	copia el último commit a esta rama
<b>git show</b>	último commit que está pegado a dos ramas
<b>git checkout nombre_rama</b>	para migrar de una rama a otra
<b>git commit -am</b>	para no perder los cambios realizados previo a migrar

#### **Fusión de ramas**

**git merge** el merge va a ocurrir donde estoy ubicado generando que el último commit de esa rama sea el principal

<b>git show</b>	muestra los cambios sobre un archivo
<b>git diff</b>	muestra la diferencia entre una versión y otra

`git diff commit1 commit2`

**`git diff --staged`**

muestra los cambios entre versiones por etapas.

**`git log`**

local

muestra el ID de los commits, el historial de confirmaciones del repositorio

**`git log -p`**

archivos y sus cambios

muestra el historial de confirmación incluyendo todos los

**`git checkout -b nombre_rama`**

crear rama y cambiarnos a ella (con un solo comando)

**`git checkout IDcommit`**

también permite movernos entre ramas

permite volver a cualquier versión anterior de un archivo,

**`git reset`**

que hicimos después de ese commit

permite volver a versiones anteriores pero borra los cambios

**`git reset --hard`**

árbol desde el commit se descartan

restablece el árbol de trabajo y el índice. los cambios en el

**`git reset --soft`**

“cambia a ser committed”. (Head: versión del commit que estoy viendo)

reinicia el head para commit. cambia los archivos a