

CS6 - Actividad 2 - Introducción a la informática

1. Una cheatsheet (recordatorio) de los comandos más utilizados hasta el momento, con una explicación de la sintaxis de cada uno y qué hacen. Tomar como referencia la cheatsheet de la clase de la terminal.

<code>git init <carpeta></code>	Crea un repositorio vacío en <carpeta> . Sin argumentos permite inicializar la carpeta actual como un repositorio.
<code>git clone <repositorio></code>	Clona un <repositorio> en nuestra máquina local. El <repositorio> puede ser local o remoto mediante HTTP or SSH .
<code>git config user.name <autor></code>	Define al <autor> para todas las confirmaciones en el repositorio actual.
<code>git add <carpeta archivo></code>	Agrega los cambios realizados en dicha <carpeta/archivo> para la siguiente confirmación. Acepta expresiones regulares.
<code>git commit -m "<mensaje>"</code>	Confirma los cambios en esa instancia, pero sin abrir o usar un editor de texto, usa el <mensaje> como mensaje de confirmación.
<code>git status</code>	Muestra la lista de archivos de los cambios montados, no montados y sin seguimiento.
<code>git log</code>	Muestra el historial de confirmaciones (formato por defecto). Para personalizar ver la sección de opciones adicionales.
<code>git diff</code>	Muestra los cambios no marcados entre el índice y la carpeta de trabajo.
<code>git branch <rama></code>	Crea una nueva <rama> . Sin argumentos muestra la lista de ramas del repositorio.
<code>git checkout -b <rama></code>	Crea una nueva <rama> en base a la rama actual. Si quitamos -b permite posicionarse en la <rama> .
<code>git merge <rama></code>	Mezcla la <rama> con la rama actual.

`git remote add
<nombre> <enlace>`

Crea una nueva conexión a un repositorio remoto. El `<nombre>` es el alias del `<enlace>` al cual se comunica.

`git fetch
<remoto> <rama>`

Obtiene la `<rama>` del repositorio `<remoto>`. Sino se especifica la rama trae toda todas las ramas.

`git pull
<remoto>`

Obtiene una copia del repositorio `<remoto>` y la mezcla al instante con el repositorio local.

`git push
<remoto> <rama>`

Envia la `<rama>` al repositorio `<remoto>` actualizandola si existe y sino creandola.