

Introducción a la Informática.

Módulo 1. Inmersión.

Sesión 1. Mayo 31 de 2021

Lucas Pasteris.

lpasteris@digitalhouse.com

Gisselle Marandet:

gmarandet@digitalhouse.com

Link Miro:

https://miro.com/app/board/o9J_IB87VQk=

Juan Pablo Montivero

Pamela Ayelen Sánchez Zarfino

Charly Correa Muñoz

Ariana Gala Ebis

Maria Gabriela Mateo

Discord:

<https://discord.gg/cQbS678e>

?rank Camada 1

Mentí:

<https://www.menti.com/et4u459wek>

[https://docs.google.com/document/d/1tX-](https://docs.google.com/document/d/1tX-0JbSxLu83VjSIMEcqNJUUnWTJwpkO1Hkp1qrzI_E/edit)

[0JbSxLu83VjSIMEcqNJUUnWTJwpkO1Hkp1qrzI_E/edit](https://docs.google.com/document/d/1tX-0JbSxLu83VjSIMEcqNJUUnWTJwpkO1Hkp1qrzI_E/edit)

Padlet:

<https://padlet.com/PedagogiaDH/ao87c6wihla6plme#>

Link para subir el Padlet:

[https://drive.google.com/drive/u/0/folders/](https://drive.google.com/drive/u/0/folders/1raBYyFt15CdkS6ILVYawgYVHTNZ2LNXM)

[1raBYyFt15CdkS6ILVYawgYVHTNZ2LNXM](https://drive.google.com/drive/u/0/folders/1raBYyFt15CdkS6ILVYawgYVHTNZ2LNXM)

Cuestionario:

[https://docs.google.com/forms/d/e/](https://docs.google.com/forms/d/e/1FAIpQLSeQdD_1Vt5b_PmJ0cfuEa0b453UrUHNmlB0yiwKircVzpNvoA/viewform)

[1FAIpQLSeQdD_1Vt5b_PmJ0cfuEa0b453UrUHNmlB0yiwKircVzpNvoA/viewform](https://docs.google.com/forms/d/e/1FAIpQLSeQdD_1Vt5b_PmJ0cfuEa0b453UrUHNmlB0yiwKircVzpNvoA/viewform)

Canal Discord:

<https://discord.com/channels/847593098288431216/847593098313072718>

Sesión 2. Junio 2 de 2021. Interfaz de usuario – Terminal (CLI)

Glosario:

Terminal

Shell

Consola

Entorno de Desarrollo Integrado

Comandos:

pwd: Mostrar el nombre de la carpeta en la que uno se encuentra situado

cd: Cambiar la carpeta de trabajo

ls: Listar el contenido de directorios

mkdir: Crear una carpeta

rmdir: Borrar una carpeta

gedit: Crear y editar archivos de texto

rm: Borrar archivos

cp: Copiar un archivo o carpeta en el directorio especificado

cp -r: Copiar carpetas

mv: Mover un archivo o carpeta a un archivo o carpeta.

find: Busca archivos o carpetas. Ej: find -name, find -size

-i: Ignora las mayúsculas

.: Permite buscar a partir del directorio en el que nos encontramos actualmente

grep: Permite buscar texto en archivos

-n: Muestra la línea donde encuentra la coincidencia dentro de los archivos

-r: Permite buscar en todas las subcarpetas recursivamente

"": Utilizar comillas cuando el nombre contiene espacios

Más trucos:

<https://www.gnu.org/software/grep/manual/grep.pdf>

Sesión 2 (Clase)

Git es un proyecto hecho inicialmente para el Kernel de Linux. Fue desarrollado por Linus Torvalds.

Git Bash es case-sensitive en Linux, (pero en Windows no).

Para copiar y pegar en Bash: Ctrl+Shift+C / Ctrl+Shift+V

mkdir subcarpeta subcarpeta2: Crea dos carpetas en la misma instrucción.

ls -R: Muestra los archivos en forma de árbol

rm: Solo remueve archivos. Ej rm texto.txt

`rm -r carpeta/`: La ``r`` significa recursivo. Permite borrar los archivos al interior de la carpeta. Elimina primero los que están más al interior.

`clear`: borrar la consola

Cuando se elimina un archivo desde la terminal, no va a la papelera de reciclaje, sino que se elimina definitivamente. (en Linux).

`^O`: Ctr + O

`nano texto.txt`: Permite editar el archivo de texto.

`Mv texto.txt text2.txt`: Cambia el nombre del archivo

`mv texto2.txt /home/....` : Mueve el archivo a la ruta *home...*

Virus ransomware: Secuestra información y la encripta.

CARPETA DRIVE

<https://drive.google.com/drive/folders/1NFwbwhw5liVwGjjxPP9FumktEhN-MjrM9?usp=sharing>

Resultado:

```
mauri@LAPTOP-P89RHVP4 MINGW64 ~/Desktop/DH
$ ls -R
.:
repaso_clase1y2/

./repaso_clase1y2:
historia_de_la_contabilidad/  infraestructura/  interfaces_de_usuario/  que_haremos_hoy/  servidores/

./repaso_clase1y2/historia_de_la_contabilidad:
repaso.txt

./repaso_clase1y2/infraestructura:
repaso.txt

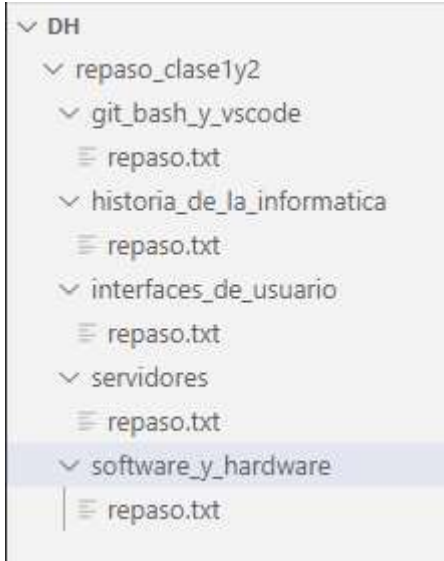
./repaso_clase1y2/interfaces_de_usuario:
repaso.txt

./repaso_clase1y2/que_haremos_hoy:
repaso.txt

./repaso_clase1y2/servidores:
repaso.txt

mauri@LAPTOP-P89RHVP4 MINGW64 ~/Desktop/DH
$ |
```

Captura 2:



Servidores:

```
GNU nano 4.9.3 repaso.txt
Los servidores se encargan de almacenar información, gestionar y dar una respuesta a las solicitudes de los clientes.
```

Sesión 3. Junio 3 de 2021.

Android es una distribución que viene de Unix.

Touch script.sh

En windows .bat; en Linux .sh o .bash

#!/bi/bash (Indica que lo que viene es un ejecutable en Linux).
echo "hola mundo"

chmod +x script.sh (Lo convierte en ejecutable).
./script.sh

En Windows:

echo off

cls

echo "hola mundo"

pause>null

Se debe guardar como .bat para que sea ejecutable

Para generar un archivo comprimido:

tar cvf ejemplo.tar carpeta/

cvf: Son condiciones c, v y f tienen cada una un significado: copiar...

c – crear un nuevo archivo .tar

v – muestra una descripción detallada del progreso de la compresión

f – nombre del archivo

Para descomprimir el archivo:

tar xvf ejemplo.tar

Link Drive:

<https://drive.google.com/drive/folders/1NFwbwhw5liVwGjjxPP9FumktEhNMjrM9>

```
mauri@LAPTOP-P89RHVP4 MINGW64 ~/Desktop/DH/Peliculas
$ ls -R
.:
Accion/ Comedia/ Infantiles/ Romanticas/ Terror/

./Accion:
Cars.txt 'Diario de una pasion.txt' 'El hexorcista.txt' 'Smooth criminal.txt'

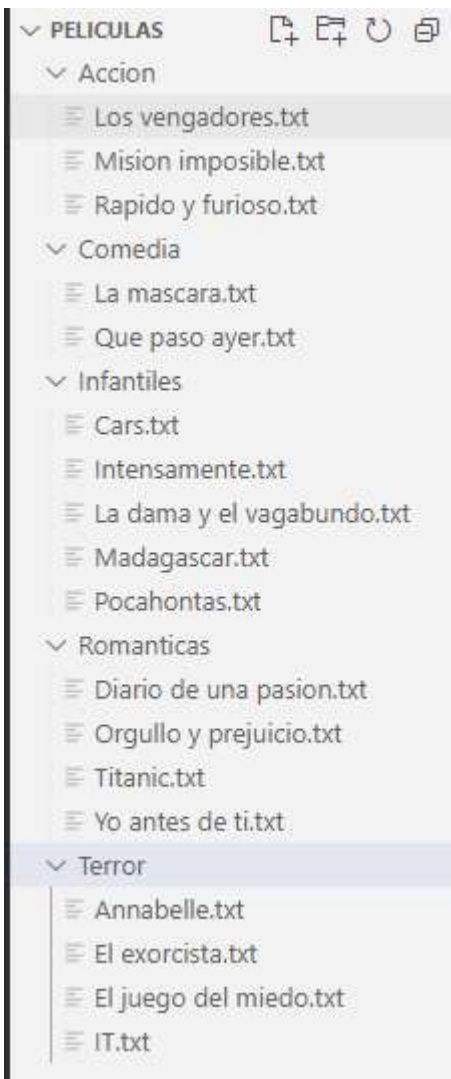
./Comedia:
IT.txt Madagaskar.txt 'Mision imposible.txt' 'Orgullo y prejuicio.txt'

./Infantiles:
'El juego del miedo.txt' Intenzamente.txt 'Rapido y furioso.txt' Titanic.txt

./Romanticas:
Annabelle.txt 'La mascara.txt' 'Los vengadores.txt' Pocahontas.txt

./Terror:
'La dama y el bagavundo.txt' 'Que paso ayer.txt' 'We will rock you.txt' 'Yo antes de ti.txt'

mauri@LAPTOP-P89RHVP4 MINGW64 ~/Desktop/DH/Peliculas
$ |
```



Sesión 4. Junio 7 de 2021. Git.

git -version: Permite verificar si git está instalado.

git: Muestra algunas indicaciones de ayuda.

git init: Crea un repositorio

git config user.name "nombreUsuario": Configuración

git config user.email "mailUsuario@email.com"

git config --global user.name "nombreUsuario": Fija el usuario globalmente

git config --global user.email "mailUsuario@email.com"

git remote add origin <https://github.com/nombreUsuario/nombreRepo.git>: Vincula el repositorio local con el remoto

git remote -v: Para verificar que el repositorio local y el remoto están vinculados

git add: Para agregar archivos

git add "nombreArchivo.ext": Agrega únicamente un archivo al repositorio

git add .: Agrega todos los archivos del repositorio

git status: Para verificar el estado de los archivos con relación al repositorio

git commit -m "Mensaje": Confirmar los archivos

git log: Historial de los cambios del proyecto

git rm --cached "nombreArchivo": Remueve archivo del seguimiento (lo contrario de git add)

En Visual Studio Code:

U: Untracked files

A: Added

M: Modificado

Glosario:

Repositorio local

Repositorio remoto

Commit: Historial de cambios que se han hecho en el proyecto

timestamp

Sesión sincrónica

git branch -M main: Cambiar master por main

git add -u: Trackea solo los archivos que están Untracked

Formulario:

https://docs.google.com/forms/d/e/1FAIpQLSeQdD_1Vt5b_PmJ0cfuEa0b453UrUHNmlB0yiwKircVzpNvoA/viewform

Código: 0521TDIIFT1C1LAED

Link

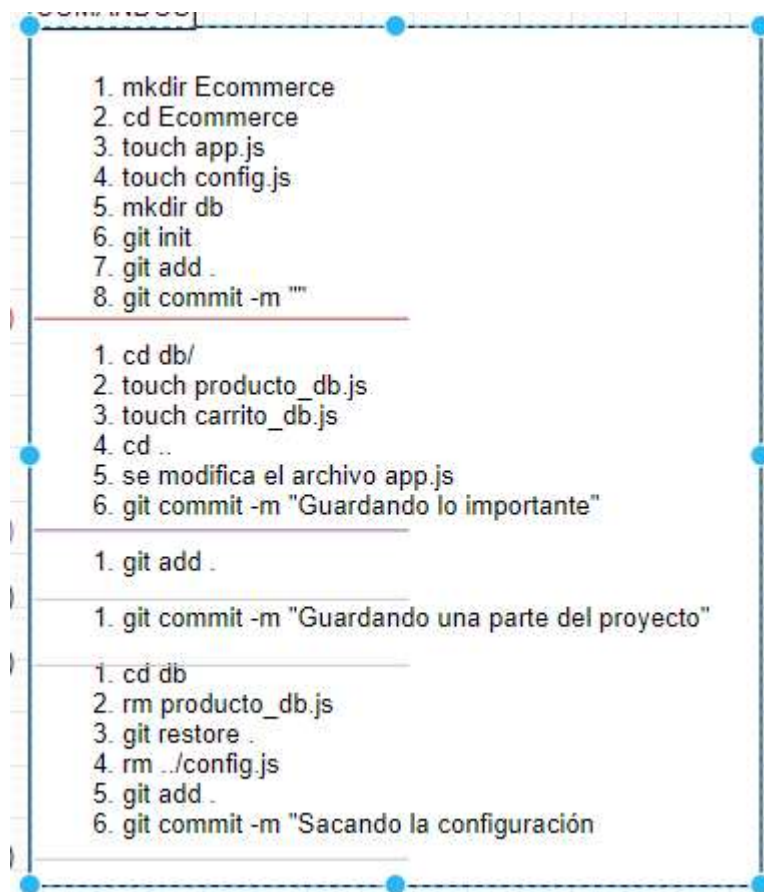
<https://drive.google.com/file/d/1616D4h13q-dGtqhOkjnyZkd622EXHcFB/view?usp=sharing>

https://drive.google.com/drive/folders/101iJLPgdP_9qzj6rfA_OGBPjdN0nU8wS

Diagrama equipo 1

<https://app.diagrams.net/#G1616D4h13q-dGtqhOkjnyZkd622EXHcFB>

h



Sesión 5. Junio 9 de 2021.

git push origin main: Para subir los cambios al repositorio remoto.

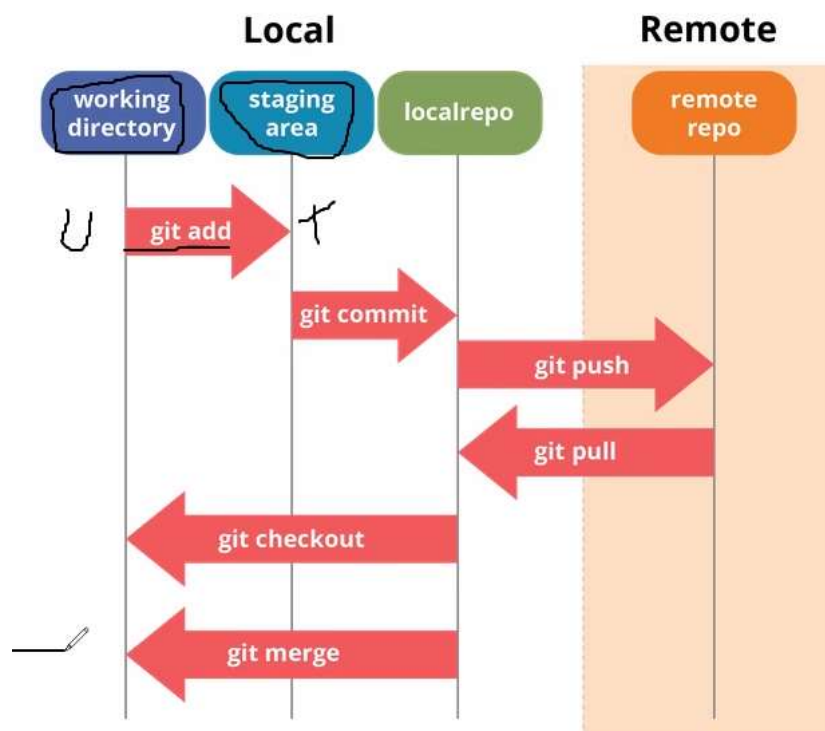
git clone "URL_repositorio": Crear una copia de todos los archivos existentes en el repositorio remoto. Se ejecuta una sola vez.

git pull origin main: Descarga los archivos nuevos o con cambios de la rama principal del repositorio remoto.

git branch...

(Falta)

Sincrónico.



git checkout nombreRama: Para cambiar de rama

Merge: Permite fusionar dos ramas

git log -stat: Muestra los cambios realizados en cada commit

El proceso para lograr la integración de los archivos sería:
pull

git checkout (nombre rama)
merge (rama de segunda importancia)
push origin

git branch -a: Muestra también las ramas del repositorio remoto

git merge nombreRama: Funciona la rama actual con la rama "nombreRama"

git push -u origin dev: 'Sube' la rama dev al repositorio remoto.

Sesión 6.

Sincrónico.

git commit --amend: Permite editar el commit
Presionar una tecla para empezar a editarlo
Para guardar: Crt+O
Para salir: UpperCase+ZZ
Genera un nuevo hash, ya que el commit fue editado.

Git log: Para ver la lista de commit

Shit + Q: Permite salir de la lista (??)

git reset: Para volver a un comando anterior. Existen tres modos: soft, mixed y hard.

git reset --hard xxxx: Se restablece (?) el commit xxxx (xxxx es el número del commit)

Los commit no se eliminan.

Con reset hard se borran todos los cambios de los commit anteriores posteriores al que se restablece.

Con otras versiones del reset, se pueden conservar algunos cambios de los commit posteriores.

Los commit modificados por amend se conservan por algún tiempo (30 días).

git reflog: Muestra todo el historial de commit

git diff xxx yyy: muestra las diferencias entre el commit xxx y el yyy.

Aprender git branching:

https://learngitbranching.js.org/?locale=es_ES

git restore archivo: Cuando se requiere traer un archivo desde el staging area al working directory

Consigna:

1. Clonar el repositorio de la cursada.
2. Crear una rama: **Apellido_Nombre**.
3. Crear una carpeta con el nombre de **Git_y_Github_Apellido_Nombre** en la rama.
4. En la misma crear un archivo respondiendo las siguientes preguntas:
 - a. ¿Quién Inventó el sistema de control de versión Git y por qué?
 - b. ¿A quién pertenece actualmente Github y por qué?
 - c. ¿Hay otra forma que no sea la terminal para trabajar con Github?
5. Subir la carpeta a la mochila.
6. ¡No te olvides de subir esta misma actividad a tu mochila individual!

¡Hasta la próxima!



Consigna:

<https://docs.google.com/document/d/14y4hCwxEqttLQr7cJGm2QRVCE2O2CQnHVDLk06yFX74/edit>

https://docs.google.com/document/d/1gPxgWA69lgJebe7mUOUOXz_ZmDzwWfsGNCRvIADa3JE/edit#heading=h.ivyp1fktx3ef

1. Aceptar invitación a colaborar.
2. Clonar repositorio remoto (ramas dev y main).
3. Crear rama Apellido_Nombre.
4. Crear archivos en la rama Apellido_Nombre. (En la carpeta de cada clase, creás una carpeta apellido_nombre y adentro van tus archivos).
5. Hacer merge con la rama dev local.
6. Hacer push de la rama dev local a la rama dev remota. (Git te va a pedir que hagás primero pull).
7. Para cada nueva tarea, repetir desde el paso 4.

Nuevas consignas:

https://docs.google.com/document/d/1njR4qb_z-b6hZnzH39nUObafUd5pVJ4IAmx5GkAE6BE/edit

https://docs.google.com/document/d/1TZCMI7Bw_jjA5Oj53MfLkSESjmdQ0NpNQr3UDe0S-8/edit