

Iniciamos 13:10

Registra tu asistencia en
bit.ly/tecodingabril2024



bash

Taller de bash scripting



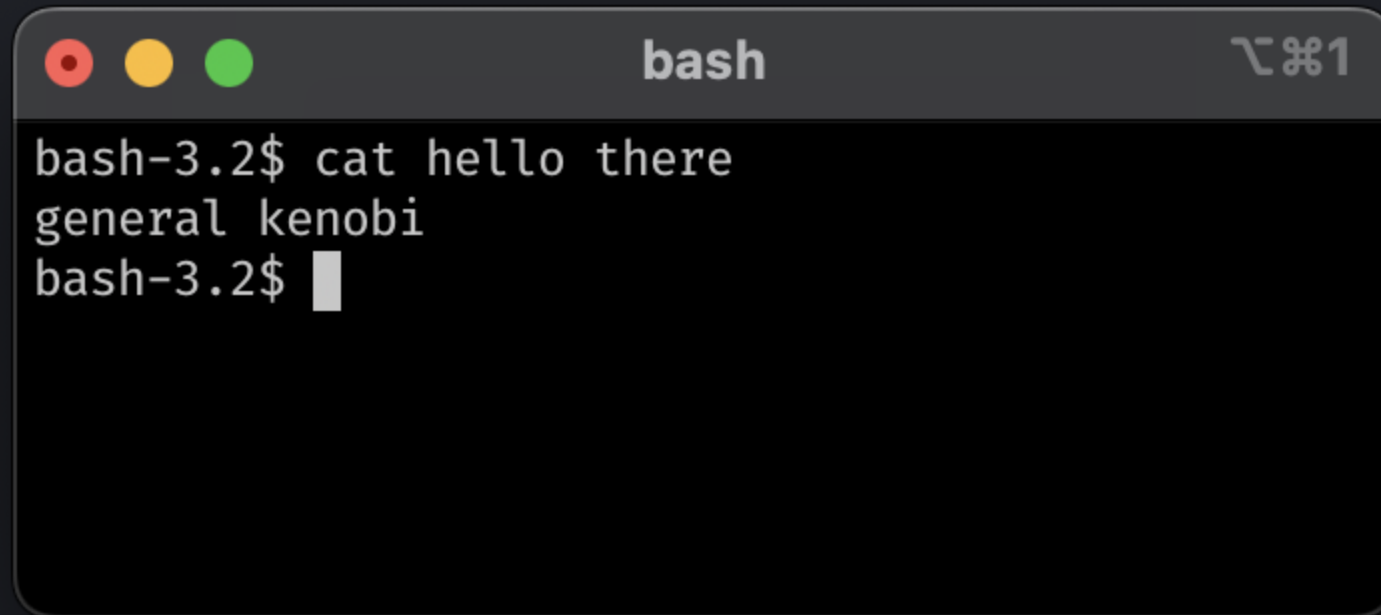


Han Rodríguez

VP TECoding
8° ITC

- AI Research @ Tec
- 2x Fellow @ MLH
- SWE Intern @ Intel
- SWE Intern @ MSFT

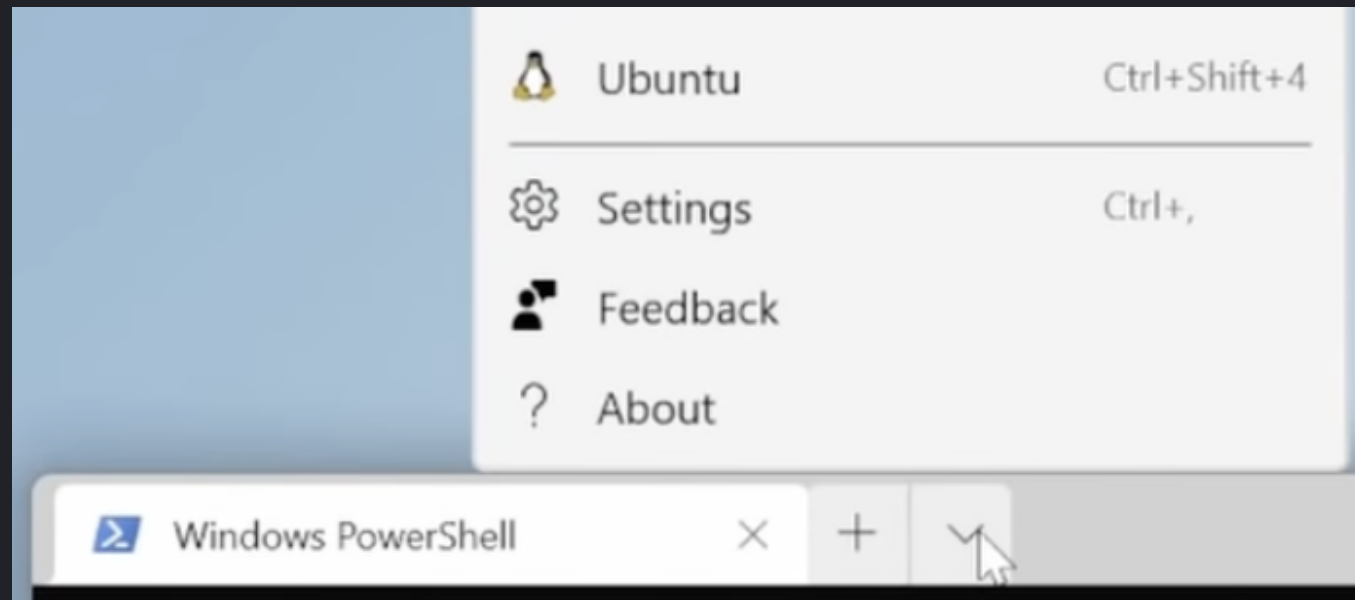
Interfaz de línea de comandos



A terminal window with a dark gray title bar. The title bar contains three colored window control buttons (red, yellow, green) on the left, the text "bash" in the center, and a window icon and "1" on the right. The terminal area is black with white text. It shows the prompt "bash-3.2\$" followed by the command "cat hello there". The output "general kenobi" is displayed on the next line. The prompt "bash-3.2\$" is followed by a white cursor block on the third line.

```
bash-3.2$ cat hello there
general kenobi
bash-3.2$ █
```

1. Instalar Windows Terminal de la Tienda Microsoft
2. Abrir la terminal como administrador (clic derecho)
3. `wsl.exe --install` **(es necesario reiniciar)**
4. Introducir nombre de usuario y contraseña nuevos
5. Abrir la terminal y crear una pestaña de Ubuntu



Verificar que estemos en el directorio de nuestra cuenta

Windows WSL: `/mnt/c/Users/username/`

macOS: `/Users/username/`

Comandos

<code>pwd</code>	# parent working dir.	- dirección de carpeta actual
<code>ls</code>	# list	- archivos en carpeta
<code>cd</code>	# change directory	- cambiar de carpeta
<code>mkdir</code>	# make directory	- crear carpeta
<code>rmdir</code>	# remove directory	- borrar carpeta vacía
<code>touch</code>	# create	- crear archivo vacío
<code>rm</code>	# remove	- borrar archivo o carpeta (-r)
<code>cp</code>	# copy	- copiar archivo o carpeta (-r)
<code>mv</code>	# move	- mover archivo o carpeta (-r)
<code>echo</code>	# print	- imprimir en pantalla
<code>chmod</code>	# change mode	- cambiar permisos de archivo/script
<code>cat</code>	# concatenate	- mostrar contenido de archivo
<code>grep</code>	# global regex print	- buscar patrón de texto
<code>></code>	# output redirection	- pasar salida de comando a archivo
<code> </code>	# pipe	- pasar salida de comando a otro comando
<code>man [cmd]</code>	# q: salir	
<code>[cmd] -h</code>	<code>[cmd] --help</code>	

Creemos nuestro primer script

holahola.sh

```
#!/usr/bin/env bash  
  
echo "Hola, mundo!"
```

Habilitar ejecución y probar

```
[gonzalpi@macos ~]$ chmod u+x holahola.sh  
[gonzalpi@macos ~]$ ./holahola.sh  
# Hola, mundo!
```


Habilitar globalmente

```
cd                # regresar a directorio raíz de bash
mkdir custom_scripts
nano .bashrc      # .zshrc para macOS, agregar sig. línea al final:
                  # export PATH="$HOME/custom_scripts:$PATH"
mv holahola.sh custom_scripts/holahola # mover script a carpeta
exit
```

```
holahola          # script puede correr desde cualquier directorio
```

Script con argumentos

El script recibe argumento para nombrar archivo

```
#!/usr/bin/env bash
if [[ -n "${1}" ]]
then nuevo_archivo="${1}.md"
else nuevo_archivo="nuevas_notas.md"
fi
if [[ -f "${nuevo_archivo}" ]]
then echo "El archivo ${nuevo_archivo} ya existe"
    exit 1
fi
echo "# Notas: ${nuevo_archivo}" > ${nuevo_archivo}
echo "El archivo ${nuevo_archivo} ha sido creado"
```

Ejercicio

El script recibe un segundo argumento y lo pone de título

```
#!/usr/bin/env bash
if [[ -n "${1}" ]]
then nuevo_archivo="${1}.md"
else nuevo_archivo="nuevas_notas.md"
fi
if [[ -f "${nuevo_archivo}" ]]
then echo "El archivo ${nuevo_archivo} ya existe"
    exit 1
fi
echo "# Notas: ${titulo}" > ${nuevo_archivo}
echo "El archivo ${nuevo_archivo} ha sido creado"
```

Solución

```
#!/usr/bin/env bash
if [[ -n "${1}" ]]
then nuevo_archivo="${1}.md"
else nuevo_archivo="nuevas_notas.md"
fi
if [[ -f "${nuevo_archivo}" ]]
then echo "El archivo ${nuevo_archivo} ya existe"
      exit 1
fi
if [[ -n "${2}" ]]
then titulo="${2}"
else titulo="${nuevo_archivo}"
fi
echo "# Notas: ${titulo}" > ${nuevo_archivo}
echo "El archivo ${nuevo_archivo} ha sido creado"
```