

Programa DevOps



Working Time #1

≡ Desafío 2 – Ramas

Es hora de que pongas en práctica todo lo aprendido. 🧐

Este apartado tiene el objetivo de ayudarte a seguir potenciando tus habilidades, por lo que a continuación encontrarás diferentes **desafíos** que podrás resolver de forma independiente y a tu ritmo.

En cada ejercicio conseguirás las resoluciones para que valides tus respuestas y puedas monitorear tu progreso. 😊

¡Manos a la obra!

1. Introducción:

Vamos a trabajar sobre comandos Git avanzados, estos comandos te permitirán conocer cómo manejar ramas dentro de git, posteriormente te servirá para manejar políticas de ramificación (branching Strategies) como pueden ser Gitflow o GitHub Flow.

2. ¿Dónde se lleva a cabo?

Línea de Comandos de GIT (git cmd)

Puede también, si su SO es Linux, utilizar git bash pero no voy a mostrar todos los comandos para Linux, deberá traducir los mismos de CMD a Bash en algunos casos.


3. Tiempo de dedicación

Entre 30 minutos y 1 hora

4. Recursos

Instale la versión de Git para su Sistema Operativo.

<https://git-scm.com/downloads>

( Nota: Las versiones de git para Windows, Linux y macOS pueden variar sutilmente dependiendo del sistema operativo, esto es debido a que git CMD utiliza sintaxis de DOS mientras que git Bash utiliza la de Unix.

Asimismo, puede haber variantes en la versión macOS debido al mismo punto.)

Debe tener realizado el ejercicio anterior: DESAFIO 1, en caso de que no lo tenga realizado o lo haya borrado de su equipo puede armar rápidamente el entorno ejecutando los siguientes comandos:

Abrir git cmd y posicionarse en el directorio donde desee trabajar

```
> git clone https://github.com/bominatvm/libro-git/
> cd libro-git
> git reset --hard 48ed8
> git remote remove origin
```

5. Desafío

Ejercicio 1

Crear una nueva rama bibliografía y mostrar las ramas del repositorio.

SOLUCIÓN:

```
> git branch bibliografía
> git branch -av
```

Ejercicio 2

1. Crear el fichero capitulos/capitulo4.txt y añadir el texto siguiente

En este capítulo veremos cómo usar GitHub para alojar repositorios en remoto.

2. Añadir los cambios a la zona de intercambio temporal.

3. Hacer un commit con el mensaje "Añadido capítulo 4."

4. Mostrar la historia del repositorio incluyendo todas las ramas.

SOLUCIÓN:

```
> Notepad.exe capitulos/capitulo4.txt
```

En este capítulo veremos cómo usar GitHub para alojar repositorios en remoto.

(guardar cambios)

```
> git add .
> git commit -m "Añadido capítulo 4."
> git log --graph --all --oneline
```

Ejercicio 3

1. Cambiar a la rama bibliografía.

2. Crear el fichero bibliografia.txt y añadir la siguiente referencia
Chacon, S. and Straub, B. Pro Git. Apress.
3. Añadir los cambios a la zona de intercambio temporal.
4. Hacer un commit con el mensaje "Añadida primera referencia bibliográfica."
5. Mostrar la historia del repositorio incluyendo todas las ramas.

SOLUCIÓN:

```
> git checkout bibliografia  
> Notepad.exe bibliografia.txt  
- Chacon, S. and Straub, B. Pro Git. Apress.  
(guardar los cambios)  
> git add .  
> git commit -m "Añadida primera referencia bibliográfica."  
> git log --graph --all --oneline
```

Ejercicio 4

1. Fusionar la rama bibliografia con la rama master.
2. Mostrar la historia del repositorio incluyendo todas las ramas.
3. Eliminar la rama bibliografia.
4. Mostrar de nuevo la historia del repositorio incluyendo todas las ramas.

SOLUCIÓN:

```
> git checkout master  
> git merge bibliografía  
> git log --graph --all --oneline  
> git branch -d bibliografia  
> git log --graph --all --oneline
```

