

# CUESTIONARIO PRÁCTICA FINAL

Yolanda Barreiro

## Tarea 1

***Créate un alias, que funciones sólo en este repositorio, y que equivalga al siguiente comando: git log - -oneline - -branches - -graph - -remotes ¿Qué comando has utilizado para crear el alias?***

git config alias.milog "log - -oneline - -branches - -graph - -remotes"

## Tarea 2

***¿Están las dos ramas del repositorio trackeando las correspondientes ramas remotas? ¿Qué comando has usado para averiguarlo?***

Sí, ambas ramas del repositorio están trackeando sus correspondientes ramas remotas.

El comando usado para saberlo ha sido:

git branch -vv

## Tarea 5

***Corrige la contraseña y modifica el commit anterior, sin crear un commit más en el log. ¿Qué comando has usado para hacer el punto anterior?***

git commit - -amend

## Tarea 7

***Te has arrepentido: ahora crees que sería mejor que estuviera todo en un solo commit. Haz los pasos necesarios para que tu rama temporal tenga un solo commit, sin tocar nada del código de los archivos. ¿Qué comandos has usado para conseguir el punto anterior?***

```
git reset - -soft 89b24a2  
git commit -m "Changes bbdd engine"
```

## **Tarea 9**

***Ahora vuelve a mover el HEAD al commit en el que estabas. ¿Qué comando has lanzado para hacer esto último?***

```
git checkout "#9-move-head-backwards"
```

## **Tarea 11**

***Tienes que hacer que el repositorio ignore la carpeta .settings y el archivo .project, pero como ya están siendo versionados necesitas primero desversionarlos. Una vez desversionados (sacados del repositorio local), haz que el repositorio los ignore. Crea un commit con los cambios. ¿Qué puntos has lanzado para los dos puntos anteriores?***

```
git rm -r - -cached .settings/  
git rm -r - -cached .project
```

```
git add .gitignore
```

```
git commit -m "Untrack and ignore .settings/ and .project"
```

## **Tarea 12**

***Te das cuenta de que has hecho el commit en develop, cuando tenías que haber creado una nueva rama para ello y haber hecho el commit ahí. Da los pasos necesarios para corregir el error.***

***¿Qué comandos has lanzado para resolver el error?***

```
git reset - -soft HEAD~1
```

```
git checkout -b "#12-replace-returnbook-with-returnabook"
```

```
$ git commit -m "Replace 'Return Book' with 'Return a Book'"
```

### **Tarea 13**

***Te piden que vuelvas a dejar la contraseña del admin como la que metiste en el punto 5. Teniendo en cuenta que ya tienes el cambio registrado en un commit, ¿qué harías para volver a poner esa contraseña sin tocar el código?***

```
$ git cherry-pick 64cdb8e
```

```
$ git add .
```

```
$ git cherry-pick --continue
```

### **Tarea 17**

***Tanto en local como en remoto sólo deberían quedar las ramas main y develop. Describe las acciones que has realizado y/o comandos que has lanzado para comprobarlo, y para borrar las ramas que se te hubieran quedado colgadas durante el ejercicio.***

Compruebo qué ramas hay tanto en local como en remoto:

```
git branch - -all
```

Miro en la opción "branches" del repositorio en GitHub y veo que las ramas remotas que aparecen sin borrar en la consola, ahí ya no están (pues fueron automáticamente borradas tras cada finalización los pull request).

Para borrar estas ramas remotas que siguen apareciendo localmente lanzo:

```
git fetch - -prune origin
```

Compruebo nuevamente con `git branch - -all` y veo que se han borrado todas las ramas sobrantes menos una, la cual estaba ya al inicio del proyecto. Para borrarla lanzo:

```
git push origin --delete feature/isbn
```

Compruebo de nuevo con `git branch - -all` y veo que ya solo quedan las ramas `main` y `develop` tanto en local como en remoto.

