

Práctica 02 Comandos Git Flow a nivel local

Realizar el ejercicio mediante línea de comandos en Windows, Linux o Mac

- o Crear una carpeta llamada **gitpracticaflow** e inicializar el repositorio con el comando correspondiente tomando en cuenta qué existen dos ramas: **main y develop**

```
mkdir gitpracticaflow
```

```
cd gitpracticaflow
```

```
git flow init
```

```
Initialized empty Git repository in
```

```
D:/UTNG/C8/DWI/U-II/gitpracticaflow/.git/
```

```
No branches exist yet. Base branches must be created now.
```

```
Branch name for production releases: [master] main
```

```
Branch name for "next release" development: [develop]
```

```
How to name your supporting branch prefixes?
```

```
Feature branches? [feature/]
```

```
Bugfix branches? [bugfix/]
```

```
Release branches? [release/]
```

```
Hotfix branches? [hotfix/]
```

```
Support branches? [support/]
```

```
Version tag prefix? [] 1.0
```

```
Hooks and filters directory?
```

```
[D:/UTNG/C8/DWI/U-II/gitpracticaflow/.git/hooks]
```

- o Comprueba el estado del repositorio a través del comando correspondiente.

```
git status
```

- o Crear una nueva funcionalidad llamada **fun-login** utilizando el patrón MVC y los comandos correspondientes de **gitflow**

```
git flow feature start fun-login
```

- o Crea tres carpetas llamadas: **model, views y controller**.

```
mkdir model
```

```
mkdir views
```

```
mkdir controller
```

- o Agregar un archivo para cada carpeta

- Model: **ModelLoginDTO.java**

```
echo. > model/ModelLoginDTO.java
```

- Views: **viewLogin.html**

```
echo. > views/viewLogin.html
```

- Controller: **ControllerLogin.java**

```
controller/ControllerLogin.java
```

- o Finaliza la funcionalidad

```
git flow feature finish fun-login
```

- o Verificar las diferencias entre las ramas **main** y **develop**

```
git diff main..develop
```

- o Crear una modificación a la funcionalidad fun-login

- o **ModelLoginDTO.java** agregar la instrucción **public class modelLoginDTO {}**

```
copy con ModelLoginDTO.java
```

```
public class modelLoginDTO{}
```

- o **viewsLogin.html** agregar la instrucción **<h1> Estoy entre ramas </h1>**

```
copy con viewsLogin.html
```

```
<h1>Estoy entre ramas</h1>
```

- o **ControlleLogin.java** agregar la instrucción **public class controleLogin.java{}**

```
copy con ControllerLogin.java
```

```
public class controllerLogin.java{}
```

- o Guardar los cambios al Stanging Area

```
git add .
```

- o Verificar la historia de los commits

```
git log
```

- o Verificar las diferencias entre las ramas main y develop a través del comando

```
git diff main..develop
```

- o Iniciar una liberación con la versión 1.0 como prefijo

```
git flow release start "version1"
```

- o Iniciar la liberación con el sufijo **".1"**.

```
git commit -m ".1"
```

- o Finalizar la liberación

```
git flow release finish 'version1'
```

- Verificar las diferencias entre las ramas main y develop a través del comando

```
git diff main..develop
```