

Prueba 1. GIT. CONTROL DE VERSIONES - Ejercicio 1 (Solución)

Entorno de Trabajo

- **Editor utilizado:** Visual Studio Code.
- **Control de Versiones (Local):** Git.
- **Control de Versiones (en la nube):** GitHub.
- **Sistema Operativo:** Windows.
- **Numero de ordenadores:** 1.

Desarrollo

1. El desarrollador principal crea el repositorio de Git local y el repositorio principal de GitHub.

- Se crea directorio de trabajo llamado "prueba1".
- Se inicializa el repositorio de Git local usando el comando:

> git init

- Se crea el fichero "index.html".

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Prueba1 - Git. Control de Versiones.</title>
</head>
<body>
  <h2>Esta es parte de la solucion de la Prueba1 del tema "Git. Control de Versiones."</h2>
</body>
</html>
```

- Se realiza el primer commit usando los siguientes comandos:

> git add -A

> git commit -m "initial"

Esto hace que se añada el fichero index.html al repositorio Git local.

- Se crea el repositorio del proyecto principal en GitHub con el nombre de "Deusto-Tema1-Git-Prueba1".

- Se enlaza el repositorio local con el repositorio principal de GitHub, usando los siguientes comandos:

- > **git remote add origin <https://github.com/UnshiftedSet54/Deusto-Tema1-Git-Prueba1.git>**
- > **git push -u origin main**

Esto hace que se añadan los commits del repositorio git local al repositorio de GitHub.

2. El segundo desarrollador realiza el "fork" del repositorio de GitHub.

- Para poder simular el "workflow" (flujo de trabajo) de varios desarrolladores, se crea un usuario diferente en Github. En este caso el usuario se llama "SmlRos". Además, al ser un solo ordenador, es necesario eliminar las credenciales del desarrollador principal.
- Una vez creado el usuario se realiza el fork del proyecto principal haciendo "click" en el botón "fork" en la parte superior derecha del repositorio remoto de Github.
- Una vez hecho el fork, es necesario clonar el proyecto principal en Github, de manera que se copien todos los archivos del proyecto principal a cualquier directorio de elección (en este caso "dev2") y se inicialice el repositorio de Git local. Esto se realiza con el comando:

- > **git clone <https://github.com/SmlRos/Deusto-Tema1-Git-Prueba1.git>**

Con esto ya se tiene una copia del proyecto en la que se puede trabajar sin modificar nada del proyecto principal.

3. Continuación del workflow del proyecto.

- Una vez realizado el fork del proyecto, creamos una rama en el proyecto clonado para poder añadir un header. Esto se realiza con el siguiente comando:

- > **git checkout -B feature-header**

Con este comando se crea una rama llamada "feature-header" y se posiciona sobre la misma.

- Se modifica el archivo index.html de la siguiente manera:

```
<header>
  <ul>
    <li>Inicio</li>
    <li>About</li>
  </ul>
</header>
```

- Se añaden los cambios al repositorio de git local y se realiza el commit.

- > **git add -A**
 - > **git commit -m "added header"**

- Se realiza el “push” de los commits.

- > **git push -u origin header**

Se utiliza -u para permitir crear la rama “header” en el repositorio remoto, ya que esta no existe en el mismo.

Nota: para este apartado tuve que ingresar las nuevas credenciales ssh del usuario nuevo ya que me aparecía el siguiente error:

! [remote rejected] header -> header (permission denied)
error: failed to push some refs to 'https://github.com/SmlRos/Deusto-Tema1-Git-Prueba1.git'

4. Creando el pull request y realizando el merge de los cambios.

- Desde el repositorio remoto Github, se hace click en “pull request” y después en “compare & pull request”.
- Se evalúan los cambios y se presiona en “create pull request”.
- Por último, se realiza el “merge” para fusionar los cambios con el repositorio principal remoto (siempre y cuando no exista ningún conflicto).