

Git - Proyecto Banderas

Introducción

En esta práctica vamos a repasar los conceptos básicos de GIT, incluyendo cómo realizar commits, trabajar con ramas y gestionar un repositorio. Además, visualizar el historial y las ramas del proyecto utilizando gitk. Por último, publicar el proyecto en GitHub para su entrega.

Requisitos Previos

1. Tener instalado GIT en tu sistema.
2. Tener instalado Docker Desktop en tu máquina.
3. Conocimientos básicos de HTML, CSS y contenedores Docker.
4. Una cuenta en [GitHub](#).

Gitk

Aunque no es necesario para trabajar con Git, vamos a empezar a ver algunas herramientas útiles para trabajar con Git, en este caso para Sistemas Linux. Vamos a instalar Gitk, que nos servirá para visualizar nuestros commits y ramas de una forma más visual que la terminal.

Para ello solo tienes que instalar Gitk desde el repositorio.

- **Actualizar los repositorios del sistema:** Abre una terminal y ejecuta el siguiente comando:

```
sudo apt update && sudo apt upgrade -y
```

(Reemplazar `apt` por el gestor de paquetes de tu distribución si usas otra: `dnf`, `yum`, `pacman`, etc.)

- **Instalar Gitk:**

- En distribuciones basadas en Debian/Ubuntu:

```
sudo apt install gitk -y
```

Parte 1: Configuración inicial del entorno

1. Instala Apache:

- Abre una terminal y ejecuta:

```
sudo apt update sudo apt install apache2
```

- Verifica que el servidor Apache está funcionando:

```
sudo systemctl status apache2
```

Si está activo, abre tu navegador y accede a <http://localhost>. Deberías ver la página de bienvenida de Apache.

2. Configura la carpeta de trabajo:

- Cambia al directorio donde Apache almacena sus archivos por defecto:

```
cd /var/www/html
```

- Borra los archivos existentes (si es necesario):

```
sudo rm -rf *
```

- Cambia los permisos para que puedas trabajar fácilmente:

```
sudo chown -R $USER:$USER /var/www/html
```

3. Inicializa un repositorio GIT:

- En el directorio `/var/www/html`, inicializa un repositorio GIT:

```
git init
```

4. Crea el archivo HTML inicial:

- En `/var/www/html`, crea un archivo llamado `index.html` con el siguiente contenido:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
    <title>Bandera</title>
    <style>
        body {
            margin: 0;
            padding: 0;
        }
        .franja {
            height: 100px;
            width: 100%;
        }
    </style>
</head>
<body>
    <div class="franja"
style="background-color: red;">
    </div>
    <div class="franja"
style="background-color: yellow;">
    </div>
    <div class="franja"
style="background-color: red;">
    </div>
</body>
</html>
```

5. Haz tu primer commit:

- Añade los cambios al área de preparación:

```
git add index.html
```

- Realiza un commit:

```
git commit -m "Añade la bandera inicial (España)"
```

Parte 2: Creación de banderas y commits

1. Cambia los colores para representar la bandera de otro país:

- Modifica el archivo `index.html`. Por ejemplo, para la bandera de Italia (verde, blanca y roja):

```
<div class="franja"  
style="background-color: green;">  
</div>  
<div class="franja"  
style="background-color: white;">  
</div>  
<div class="franja"  
style="background-color: red;">  
</div>
```

- Guarda los cambios y haz un commit:

```
git add index.html git commit -m "Cambia los colores para la bandera  
de Italia"
```

2. Repite el proceso para al menos dos países más:

- Francia (azul, blanca y roja):

```
<div class="franja"
style="background-color: blue;">
</div>
<div class="franja"
style="background-color: white;">
</div>
<div class="franja"
style="background-color: red;">
</div>
```

- Alemania (negro, rojo y amarillo):

```
<div class="franja"
style="background-color: black;">
</div>
<div class="franja"
style="background-color: red;">
</div>
<div class="franja"
style="background-color: yellow;">
</div>
```

- Realiza un commit por cada cambio.

Parte 3: Uso de ramas

1. Crea una rama para cada bandera:

- Por ejemplo, para Italia:

```
git checkout -b italia
```

- Haz los cambios necesarios en `index.html` y realiza un commit en esta rama:

```
git add index.html git commit -m "Crea la bandera de Italia en la rama italia"
```

2. Crea ramas adicionales:

- Francia:

```
git checkout -b francia
```

- Alemania:

```
git checkout -b alemania
```

3. Alterna entre ramas:

- Cambia de rama con:

```
git checkout italia
```

- Refresca el navegador (<http://localhost>) para ver la bandera correspondiente.

Parte 4: Visualización con gitk

1. Instala gitk:

- Si no tienes gitk instalado, ejecútalo en la terminal:

```
sudo apt install gitk
```

2. Abre gitk para visualizar tu repositorio:

- En el directorio del proyecto, ejecuta:

```
gitk --all
```

- Observa el historial de commits y las ramas que has creado.
-

Parte 5: Publicación en GitHub

1. Crea un nuevo repositorio en GitHub:

- Ve a [GitHub](#) y crea un repositorio llamado `banderas_git`.
- No añadas ningún archivo como README, .gitignore o licencia.

2. Conecta tu repositorio local con GitHub:

- En la terminal, dentro del directorio del proyecto, ejecuta:

```
git remote add origin https://github.com/tu-  
usuario/banderas_git.git git branch -M main git push -u origin  
nombreRama
```

3. Sube las ramas al repositorio remoto:

- Por ejemplo, para la rama `italia`:

```
git push origin nombreRama
```

- Repite este paso para todas las ramas creadas.
-

Entrega

1. Pon el enlace de tu repositorio `banderas_git` de GitHub en la plataforma.
 2. Asegúrate de que el repositorio incluye:
 - El archivo `index.html` modificado en cada rama con los colores correctos.
 - Un historial de commits claro y descriptivo.
 - Todas las ramas creadas correctamente.
-

Evaluación

- Uso correcto de GIT (commits y ramas).
- Visualización del historial y ramas con `gitk`.
- Publicación del proyecto en GitHub.
- Estructura y diseño correcto del archivo `index.html`.

