EPET N°12

Trabajo práctico 10 - Git Básico

Sistema de Gestión de Calidad de Software

October 10, 2024

1 Introducción

Guia de git guiada y luego ejercicio de práctica

2 Ejercicio 1: Flujo básico de Git con resolución de conflictos

Seguir las instrucciones paso a paso sin omitir instrucciónes. Esto le prmitirá operar sobre un proyecto básico de Git y resolución de conflictos.

2.1 Paso 1: Crear un repositorio local y archivo inicial

1. Crear un directorio para el proyecto:

```
$ mkdir git-ejercicio
$ cd git-ejercicio
```

2. Inicializar un repositorio de Git en el directorio:

```
$ git init
```

3. Crear un archivo Python simple llamado `hello.py` con el siguiente código:

```
print("Hello, world!")
```

4. Añadir y comitear el archivo al repositorio local:

```
$ git add hello.py
$ git commit -m "Primer commit: Archivo hello.py"
```

2.2 Paso 2: Crear una rama y trabajar en ella

1. Crear una nueva rama llamada `rama1` y moverse a ella:

```
$ git branch rama1
$ git checkout rama1
```

2. Modificar el archivo `hello.py` añadiendo otra línea:

```
print("Este es un cambio en rama1")
```

3. Añadir los cambios y hacer un commit:

```
$ git add hello.py
$ git commit -m "Modificacion en rama1"
```

2.3 Paso 3: Crear otra rama y trabajar en ella

1. Crear una nueva rama llamada `rama2` y moverse a ella:

```
$ git checkout -b rama2
```

2. Modificar `hello.py` nuevamente, pero con un cambio diferente:

```
print("Este es un cambio en rama2")
```

3. Añadir los cambios y hacer un commit:

```
$ git add hello.py
$ git commit -m "Modificacion en rama2"
```

2.4 Paso 4: Volver a la rama1 y hacer un merge

1. Moverse de vuelta a la rama `rama1`:

```
$ git checkout rama1
```

2. Hacer un merge de `rama2` en `rama1`:

```
$ git merge rama2
```

3. Resolver manualmente el conflicto si aparece. Asegurarse de quedarnos con los cambios de `rama1`.
4. Una vez resuelto el conflicto, añadir los archivos y realizar el commit:

```
$ git add hello.py
$ git commit -m "Conflicto resuelto: Manteniendo cambios de rama1"
```

3 Ejercicio 2: Problema completo

3.1 Crear un Repositorio y Archivo Inicial

Crea un nuevo repositorio local llamado "proyecto_ejercicio". Dentro del repositorio, crea un archivo llamado "contador.py" con el siguiente contenido:

```
contador = 0

def incrementar():
    global contador
    contador += 1
    print(f"Contador: {contador}")
```

3.2 Realizar el Primer Commit

Agregar y hacer el commit del archivo "contador.py" con el mensaje: "Añadir contador.py con función incrementar".

3.3 Crear una Rama feature_incrementar

Crear y cambiar a una nueva rama llamada feature_incrementar.

3.4 Modificar en feature_incrementar

Modificar la función incrementar para que tome un parámetro n y aumente el contador en n en lugar de 1.

```
    contador = 0

def incrementar(n):
    global contador
    contador += n
    print(f"Contador: {contador}")
```

3.5 Modifiar funcion en master

Realizar `commit` del archivo y volver a la rama principal `master`

En la rama `master` alguien modifica el archivo ``` ``contador.py`` ``` de la siguiente forma:

```
    contador = 1

def incrementar(n):
    global contador
    contador += n
    print(f"Contador: {contador}")
```

3.6 Realizar el merge de de la rama feature_incremental

Ahora intentaremos realizar el merge de la rama `feature_incremental` contra hacia la rama `master`. Se requiere que los dos cambios queden impatados, el contador que comience en "1" y el incremento que sea de n.

4 Entrega

Para la entrega de los ejercicios:

1. Hacer un push de su repositorio a una plataforma como GitHub.
2. Enviar el enlace del repositorio a la tarea del classroom para revisión.