**Ramas en Git Hub**

**Por:**

Sebastian Montalvo Londoño.

Análisis y desarrollo de software.

**Ficha:**

2453914

**Instructor:**

**Edwar Hernando Velásquez.**

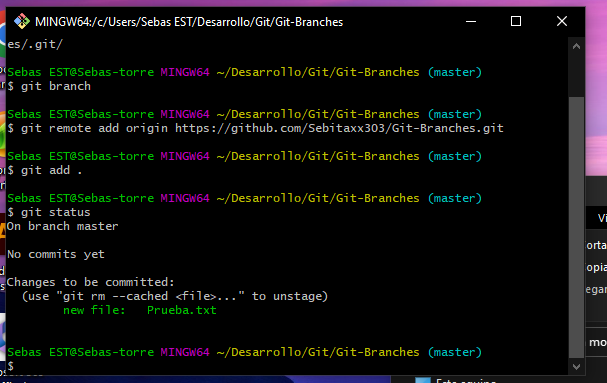
Centro Tecnológico del Mobiliario.

02/08/2023

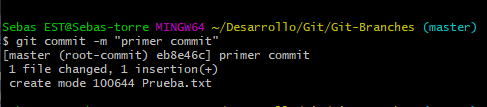
**Comandos Git Hub**

|  |  |  |
| --- | --- | --- |
| **1** | **Git log** | **Sirve para revisar los commit que se hayan hecho en la rama** |
| **2** | **Git checkout** | **Sirve para seleccionar un commit y remplazar el contenido actual de la rama por el que había en ese commit anterior.** |
| **3** | **Git checkout --** | **Sirve para descartar los cambios que se hayan hecho en un commit.** |
| **4** | **Git Branch -a** | **Sirve para mostrar todas las ramas creadas ya sean localmente o remotas.** |
| **5** | **Git fetch** | **Sirve para traer al repositorio remoto los cambios que se hayan hecho en el repositorio en línea.** |
| **6** | **Git diff** | **Sirve para identificar los cambios hechos en el archivo** |
| **7** | **Git Branch “nombre de la rama”** | **Sirve para crear una nueva rama a partir de la rama en la que se encuentre ubicado** |
| **8** | **Git merge** | **Sirve para unir los datos de una rama a otra rama** |

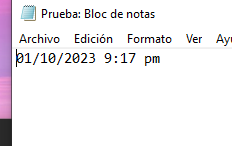
Creación y guardado del archivo de texto con la fecha:



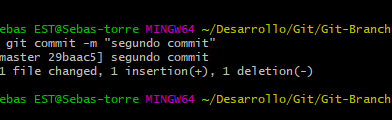
Primer commit:



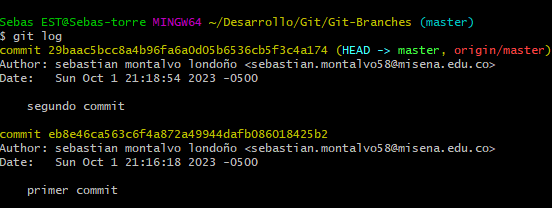
Cambios en el archivo de texto:



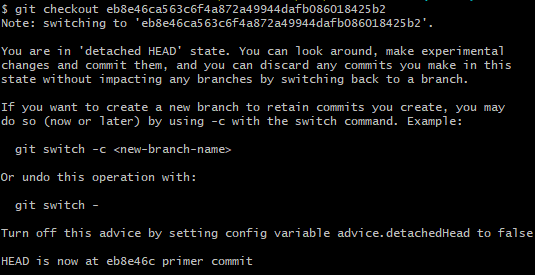
Segundo commit:



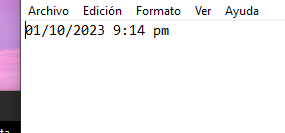
Con git log revisamos los commits realizados en la rama



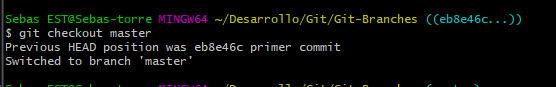
Con git checkout nos podemos parar en un commit anterior y ver el contenido que tenía:



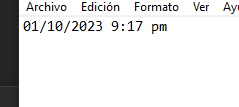
Archivo de texto en el primer commit:



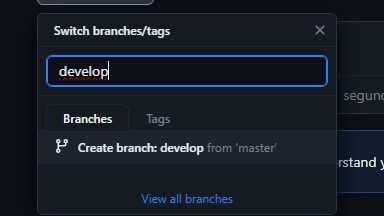
Para volver a la cabecera del proyecto, es decir, el commit mas reciente de la rama hay que hacer lo siguiente:



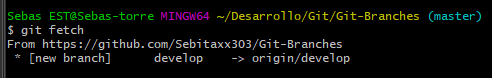
Y el archivo de texto vuelve a tener los datos del último commit:



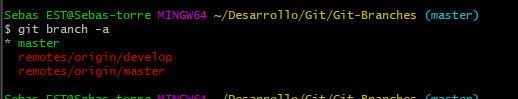
Creación de la rama DEVELOP desde el repositorio en línea:



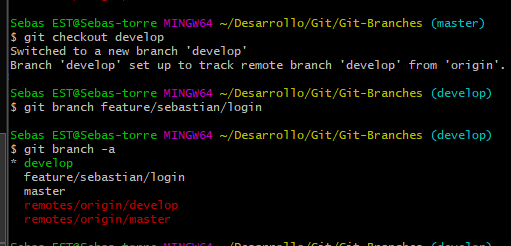
Traemos los cambios del repositorio remoto al local:



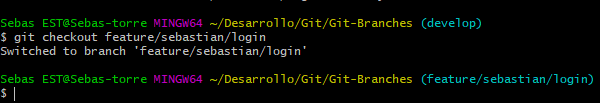
Y verificamos las ramas que hay:



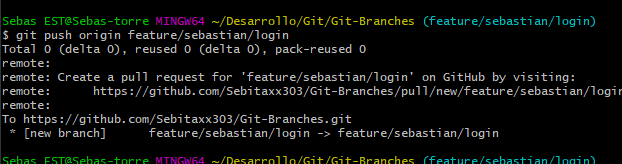
Generamos una rama de un login desde el repositorio local estando ubicado en la rama develop:

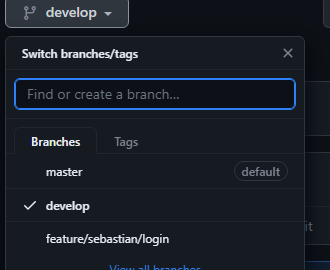


Nos ubicamos en la rama nueva:

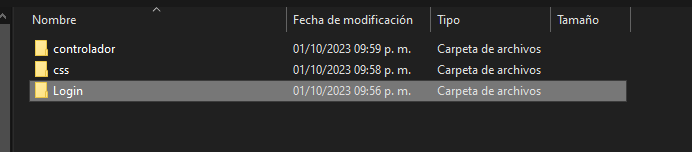


Guardamos la nueva rama en el repositorio remoto:

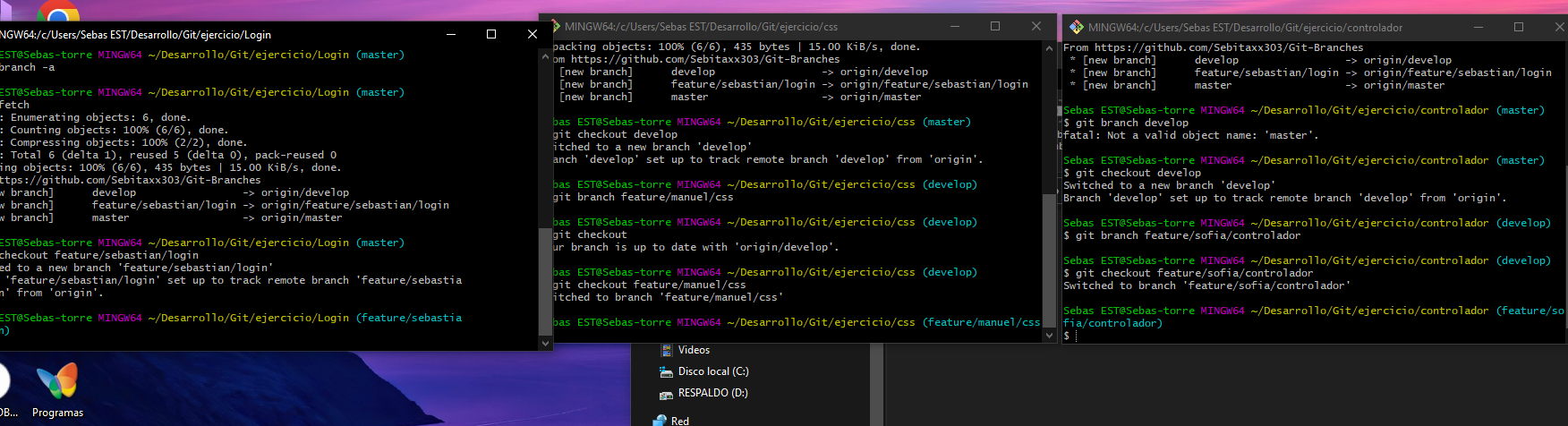




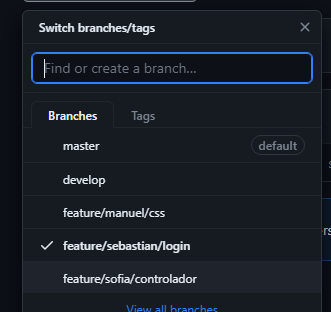
Creamos tres carpetas, simulando el repositorio local de tres trabajadores en un proyecto:

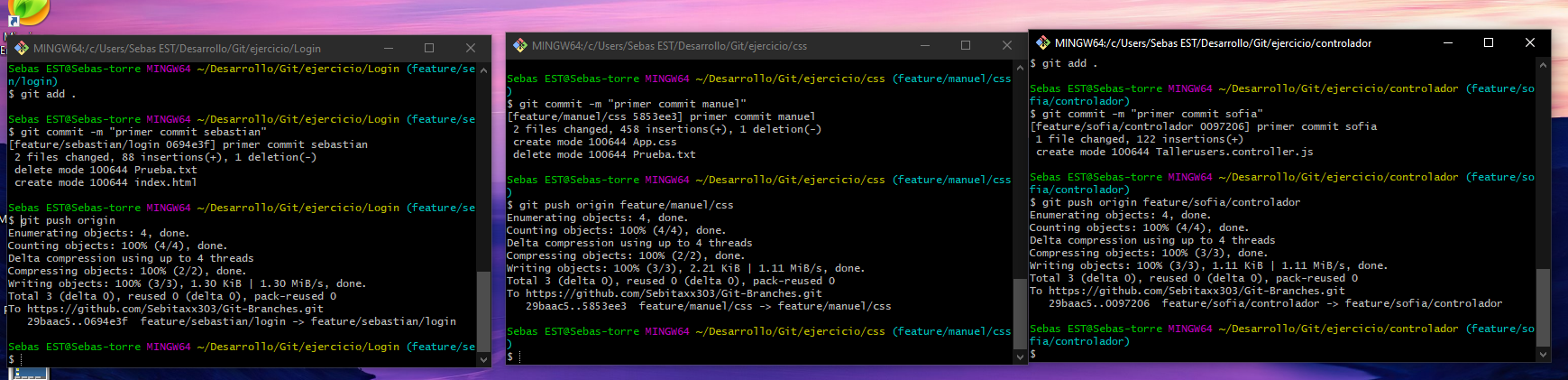


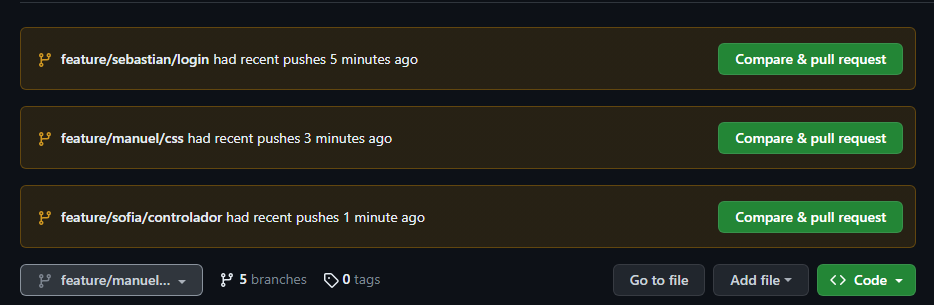
En cada repositorio local se inicializa con git, se hace fetch para tener acceso a la rama develop y se crean las ramas de cada repositorio de forma individual:



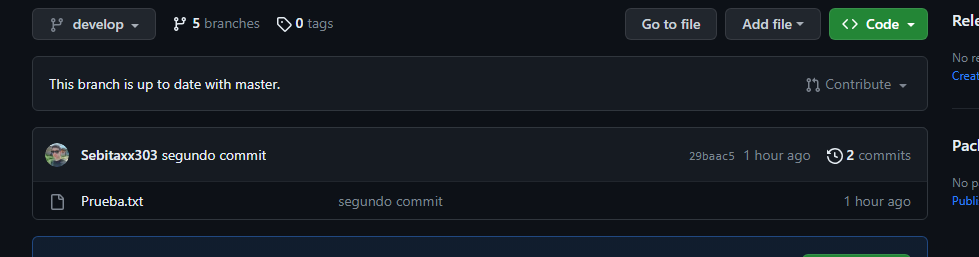
Para luego cada una guardarla en el repositorio remoto:



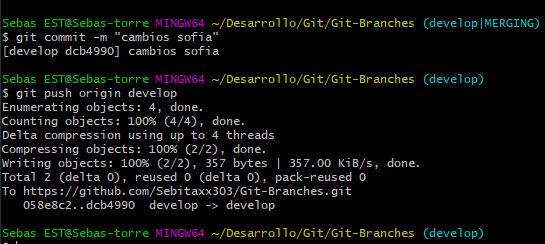
Luego en cada repositorio descendiente de develop se agrega algún archivo y se sube al repositorio remoto:



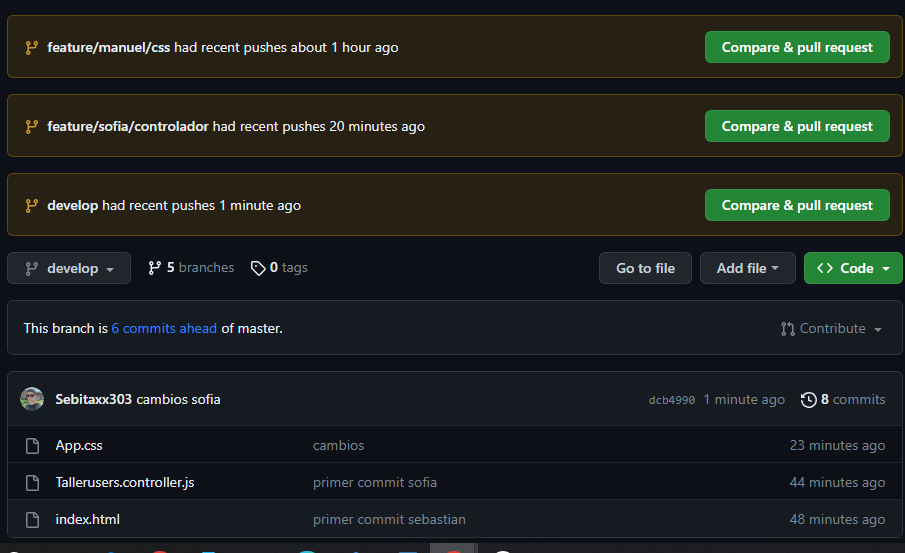
Una vez hecho esto debemos hacer un merge desde la rama en la que se quiere recibir el merge, en este caso develop la cual está vacía:



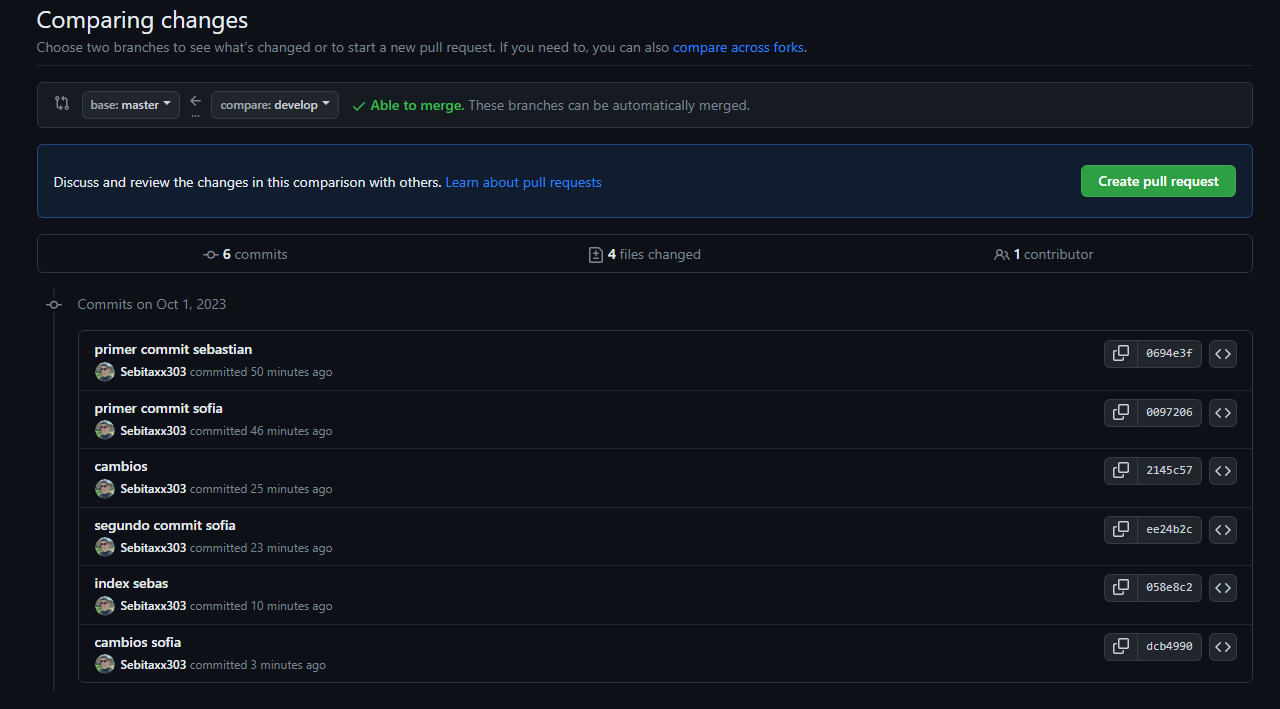
Una vez hecho merge:

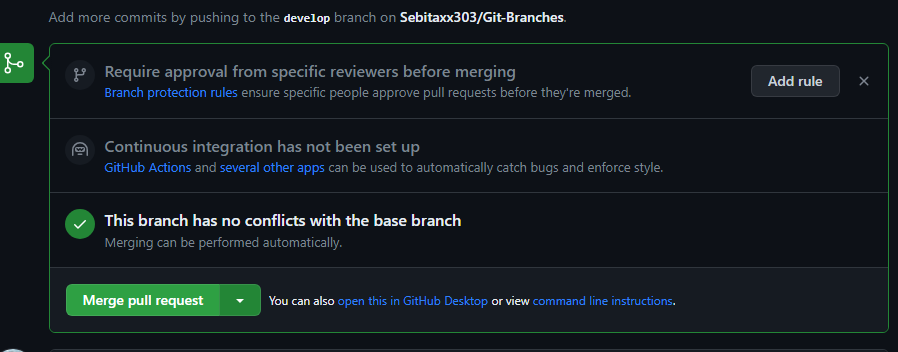


La rama develop queda con los archivos de las tres ramas de las cuales hizo merge:



Y para finalizar se hace el merge de develop a master:





La rama master queda con los archivos de la rama develop:

