**Programación Básica 2 – Universidad Nacional de La Matanza**

19

**Trabajo Práctico GIT**

**Miércoles-Turno Mañana**

**Integrantes del grupo:**

**Marmiroli, Nicolás**

**Cosentino, Rodrigo**

**Apellido, Fernando**

**Consignas**

**1.-Crear un repositorio local que se llame tp-git-pb2.**

**2.-Crear un repositorio remoto.**

**3.- Creamos un archivo llamado Calculadora.java y copiar el siguiente código**

****

**4.- Dividir el grupo en dos y realizar las siguientes acciones:**

* **El primer grupo cambiara el if-else por un switch y el segundo grupo cambiara el orden de las opciones, por ejemplo cambiando que el 3 sea para la suma y el 1 para multiplicar.**

**5. -Realizar un commit y push al repositorio remoto y responder**

* **¿Qué sucedió cuando se intentó hacer el segundo push?**

Cuando se intentó hacer el segundo push, surgió un conflicto con el repositorio, teniendo que seleccionar que partes del código del código inicial y que partes del código nuevo, quedarían en el código final a respaldar.

* **¿Qué acciones hay que tomar antes de realizar el push?**

Antes de empezar a trabajar y realizar un push, debemos realizar un pull, para asegurarnos de tener en nuestro repositorio local la última versión del código. Luego git add, para asegurarse que todo esté trackeado , y finalmente un commit, con la descripción mas clara de la versión, o cambio implementado.

* **¿Qué estrategia propondrían para evitar esta situación?**

Para evitar que los programadores tengan que resolver conflictos cada vez que hacen push de su código ( y ambos trabajan a la vez) podría crearse una bifurcación o ramas, que permitieran el trabajo individual, hasta que ambos tuvieran la versión final de su código, haciendo un único merge, de ambas ramas.

**6.- En el repositorio local ejecutar el siguiente comando: “ git checkout -b develop “**

* **¿Qué acción realiza el parámetro -b?**

El parámetro “-b” está creando la rama (Branch). En la línea de comado se está creando la rama y saltando para trabajar en ella, en la misma instrucción.

**7.- Crear ramas utilizando la recientemente creada develop utilizando los nombres de los desarrolladores**

**8.- Modificar los archivos revirtiendo cambios o realizando mejoras, siempre procurando editar los métodos y líneas. Ejemplo: separar cada acción (sumar, restar, multiplicar y dividir) en métodos**

**9.- Realizar en el repositorio remoto un merge request:**

* **¿Qué diferencias notaron en el proceso?**

El git merge hace que se fusione la rama en la que estábamos trabajando con la rama principal. Avisándonos si hay o no conflictos y en donde.

* **¿Qué estrategia utilizarían para evitar conflictos?**

Utilizaríamos medidas de división de trabajo y si hay que hacer alguna modificación que involucre el trabajo de otra persona realizar un merge de estas ramas para comparar el trabajo de ambos y minimizar los conflictos.