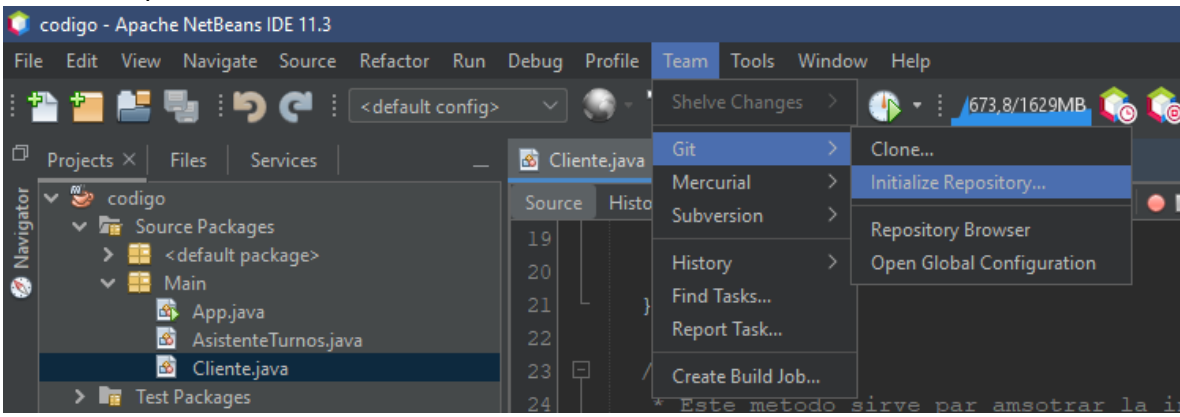


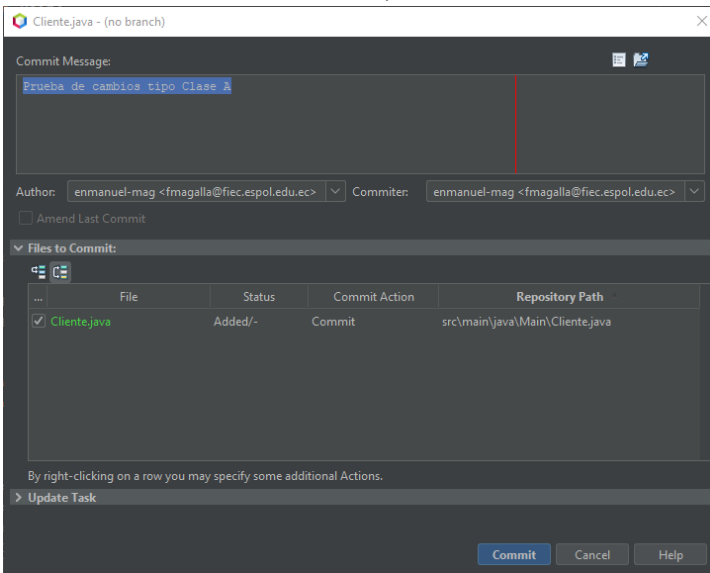
Taller de Git – BitBucket

Este taller tiene el objetivo de introducir el concepto de Git y manejo de BitBucket a través de un proyecto de Java.

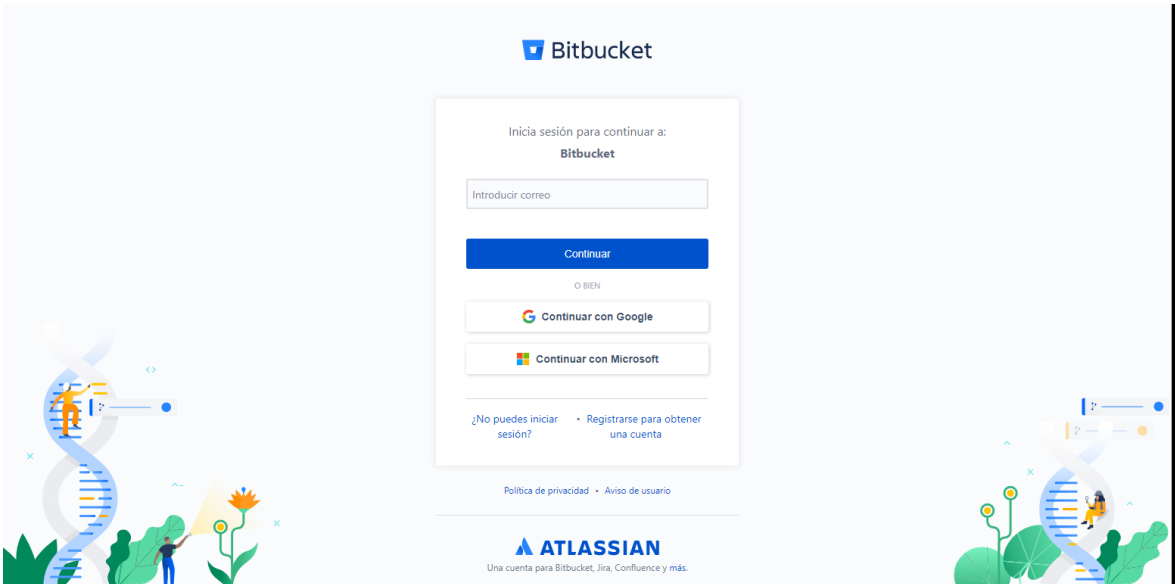
1. Descargamos [Git](#) abrimos el instalador y damos clic en siguiente a todo, no modificamos los valores por defecto.
2. Abrimos el proyecto. A continuación, vamos a la clase Cliente y completaremos el método toString de esa clase.
3. Una vez completado el método toString vamos a la parte superior del IDE NetBeans y damos clic en el botón **Team**. Luego ponemos el mouse sobre el apartado **Git** y luego en el menú que se muestra damos clic en el botón Initialize repository, finalmente damos clic en el botón **Ok** de la ventanita que se muestre.



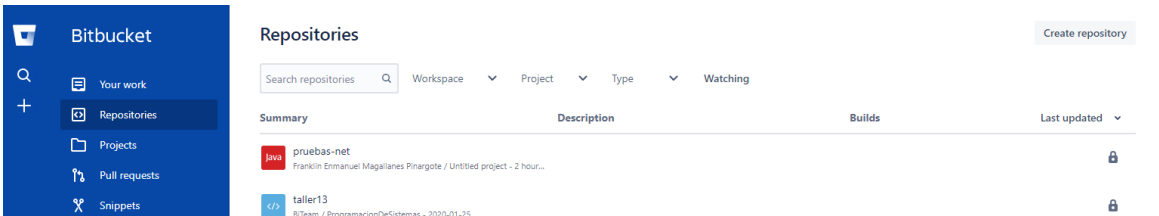
4. Ahora vamos de nuevo Teams > Add, con esto añadimos los archivos al stage del repo local.
5. Vamos de nuevo a Team > Commit y en la ventana que se muestra ingresaremos un mensaje para el Commit en el campo de texto de arriba. Abajo podemos ver los archivos que podemos añadir al commit. Finalmente, en el botón inferior derecha **Commit** para confirmar los cambios.



6. Ahora vamos a la página de [BitBucket](#) y creamos una cuenta o iniciamos sesión si ya tiene una (la cuenta tiene que usar su correo de ESPOL). Damos clic en el botón Continuar con Microsoft para usar la cuenta de ESPOL

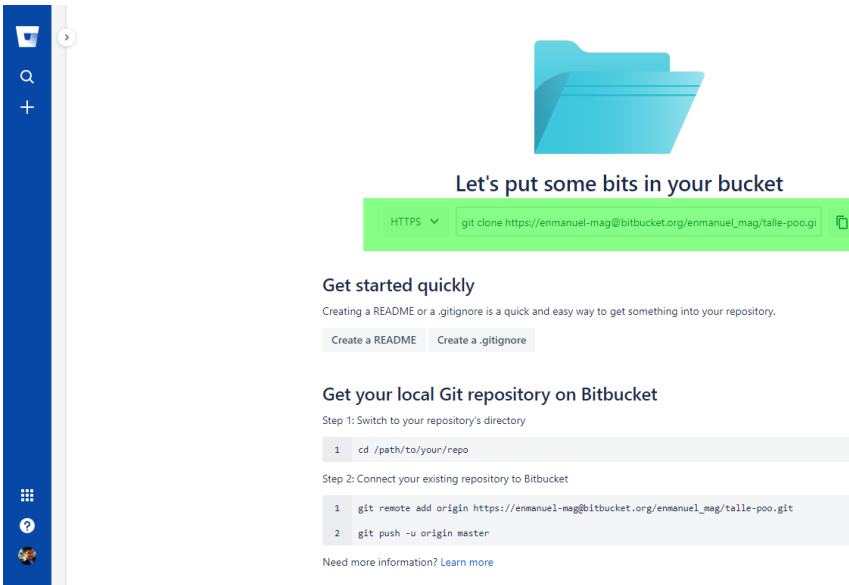


7. Una vez iniciada la sesión damos clic en el botón **Repositories** y luego en el botón Create Repositories (está en la parte superior derecha).

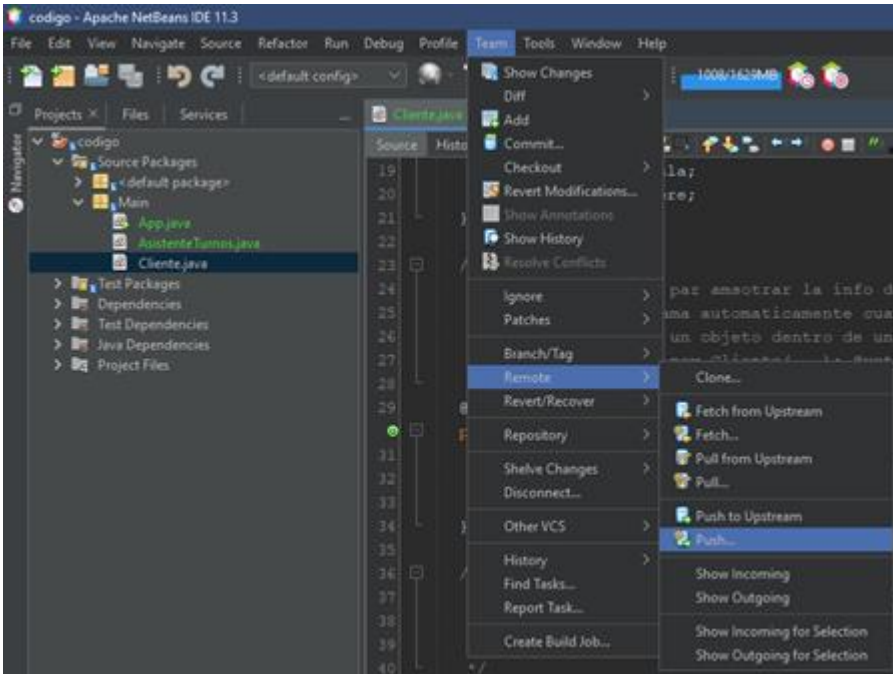


8. Veremos una pantalla como la siguiente. En el campo proyecto seleccionamos **Untitle proyecto**. En el campo **repository name** colocamos un nombre para el repo. Dejamos los demás campo por defecto. Finalmente damos clic en el botón **Create repository**

9. Veremos una pantalla como esta. De la área resaltada con verde de la imagen. Copiaremos solo el link que esta después de la palabra clone, es decir, copiaremos el link desde https://.....



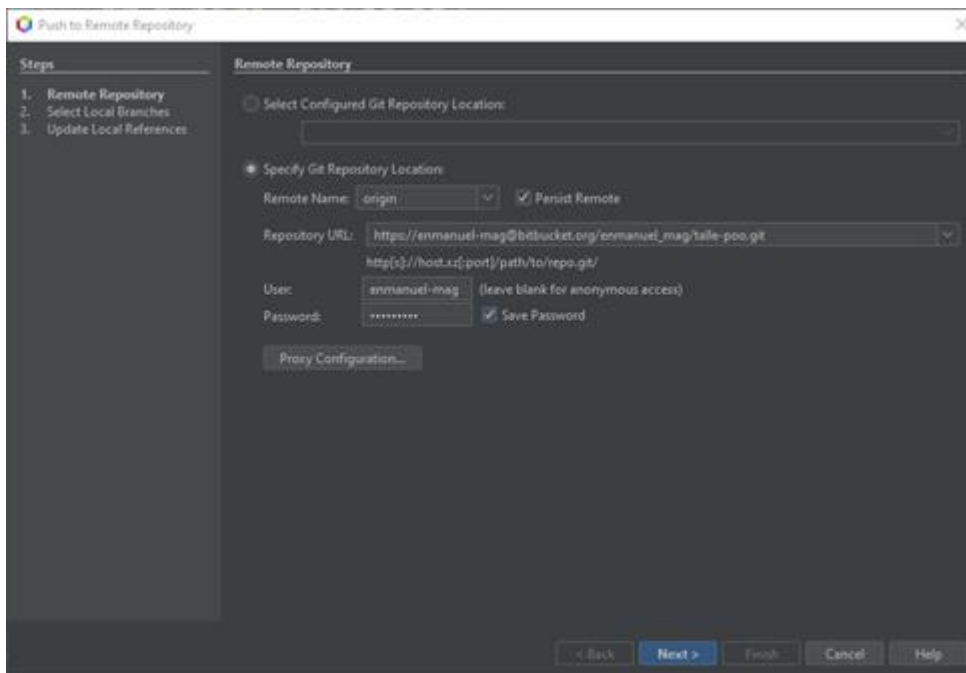
10. Ahora que tenemos un repositorio de Git **local y el remoto**, para añadir el repositorio **remoto** al local, tenemos que ir otra vez a la pestaña **Team** y llegar hasta el botón **Push...** y dar clic en él.



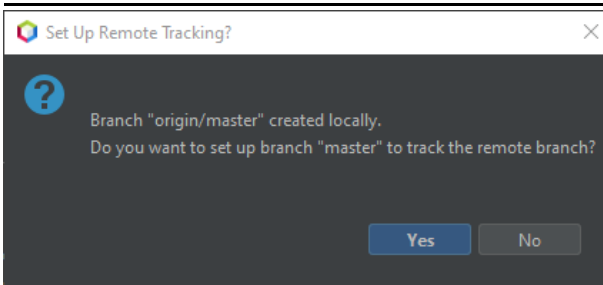
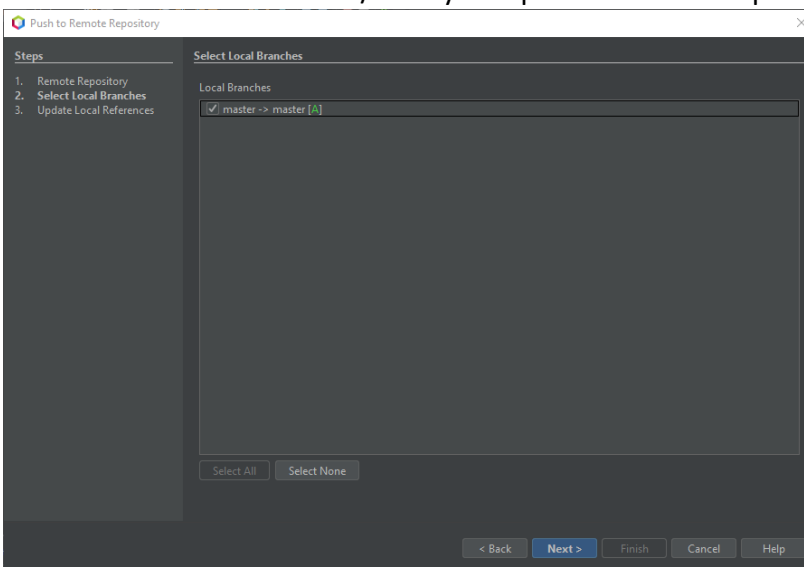
11. En la ventana que se muestra añadiremos los siguiente datos:

- En el campo Repository URL, pegaremos el link que copiamos anteriormente en la página de BitBucket.
- En el campo user colocaremos el username de su cuenta de BitBucket
- En el campo password colocamos la contraseña

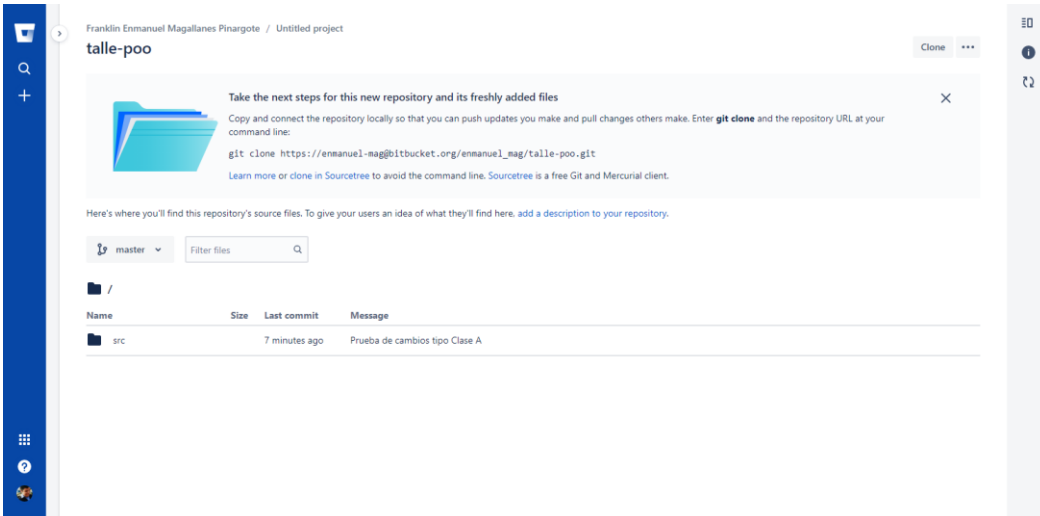
Ahora damos clic en Next



12. Damos clic en la rama master (es la única opción) y damos clic en next. Luego en la siguiente ventana damos clic en Next/Finally. Después en la alerta que se muestra damos clic en **Yes**



13. ¡Finalmente! Tenemos nuestro repositorio local y este apuntando a nuestro repositorio remoto para cuando deseemos hacer pull o push. A continuación, vamos a **Teams > Remote > Push to UpStream**. Y si vamos a la pagina del repositorio en BitBucket y la actualizamos veremos esto:



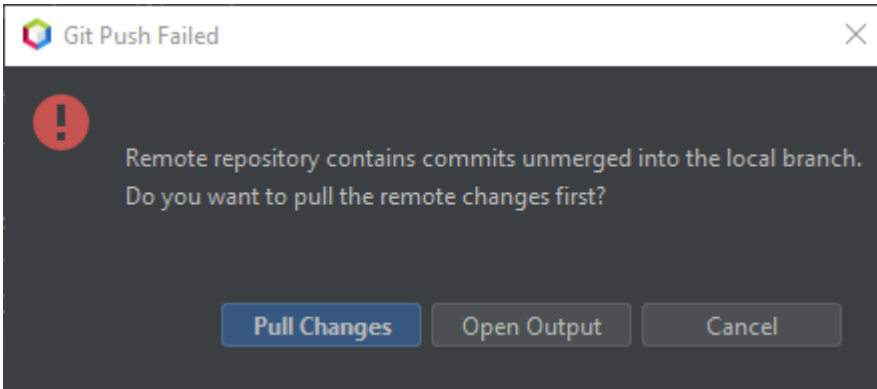
14. Ahora trabajaremos el tema de resolución de conflictos para esto vamos en la página de repo de BitBucket y buscaremos la clase Cliente y damos doble clic en él. Luego damos clic en el botón **Edit** de la parte derecha para editar el archivo y añadimos lo siguiente:

```
1 package Main;
2
3 import java.util.Objects;
4
5 public class Cliente {
6
7     private String cedula;
8     private String nombre;
9     private String motivo;
10    private int numero;
11
12    /**
13     * Constructor de la clase Cliente
14     *
15     * @param cedula identificacion del cliente
16     * @param nombre del cliente
17     */
18    public Cliente(String cedula, String nombre) {
19        this.cedula = cedula;
20        this.nombre = nombre;
21        System.out.println("Nuevo objeto creado!")
22    }
23
24    /**
25     * Este metodo sirve par amsotrar la info del objeto por meddio de un String,
26     * este metodo se llama automaticamente cuando se coloca a la varariable que
27     * hace referencia a un objeto dentro de un System.out.println(). Ejemplo:
28     * Cliente cliente = new Cliente(...); System.out.println(cliente);
29     */
30    @Override
31    public String toString() {
32        // Este metodo debe retornar un string que contener la informacion de todos los
33    }
```

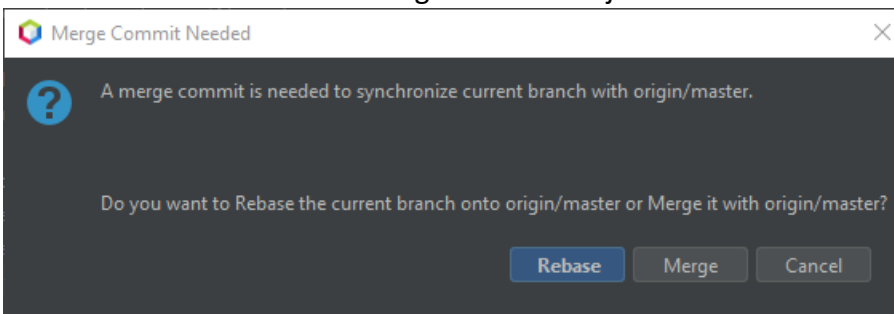
Syntax mode: Java ▼ Indent mode: Spaces ▼ Indent size: 2 ▼ Line wrap: Off ▼ View diff Commit Cancel

15. Damos clic en el botón Commit y otra vez en Commit de la ventana que se muestra. Ahora en NetBeans y vamos a la misma clase Cliente y agregamos en la misma línea que modificamos en Bitbuket pero esta vez con un texto diferente en el System.out.println().

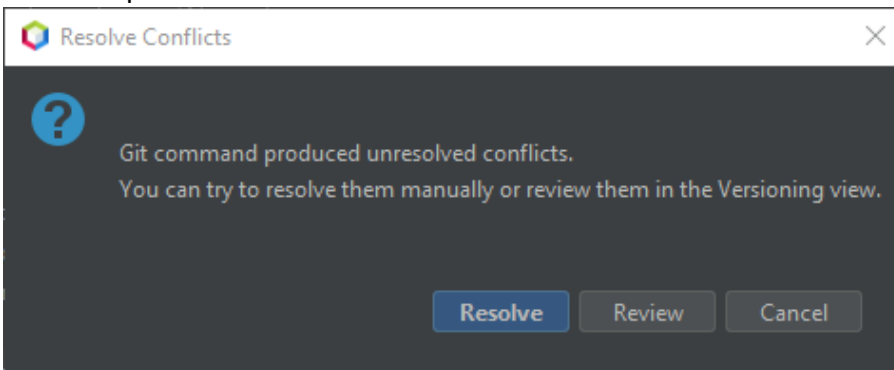
16. Ahora repetimos el proceso de **Teams > Add** (para añadir el cambio al stage), **Teams > Commit** (para confirmar los cambios, añadimos un mensaje para el commit) y finalmente **Teams > Remote > Push to UpStream**. Veremos que saldrá el siguiente mensaje.



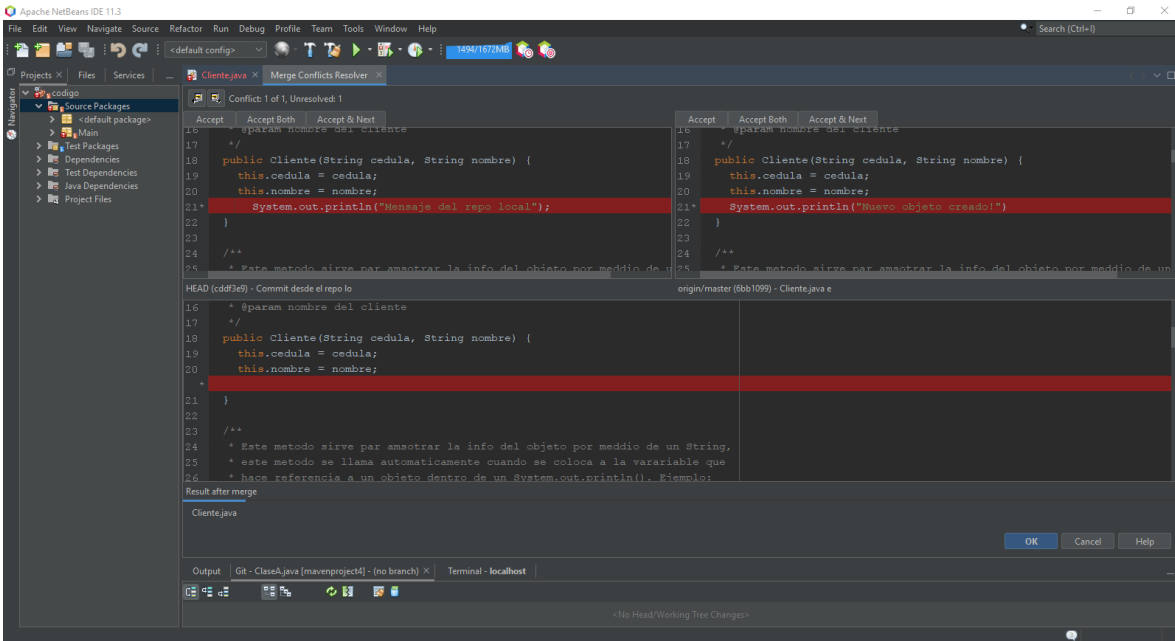
17. Ahora damos clic en **Pull Changes**. Si el cambio del repo remoto y el repo local no fueran en la misma línea del archivo, entonces solo bastaría con dar clic en **Pull Changes**, pero esto requiere intervención nuestra dado que tenemos que decidir que cambios vamos a conservar. Luego de dar clic en el botón saldrá el siguiente mensaje.



18. Ahora damos clic en **Merge**, para continuar con la resolución de conflictos. En la siguiente ventana que se nos muestra damos clic en **Resolve**.

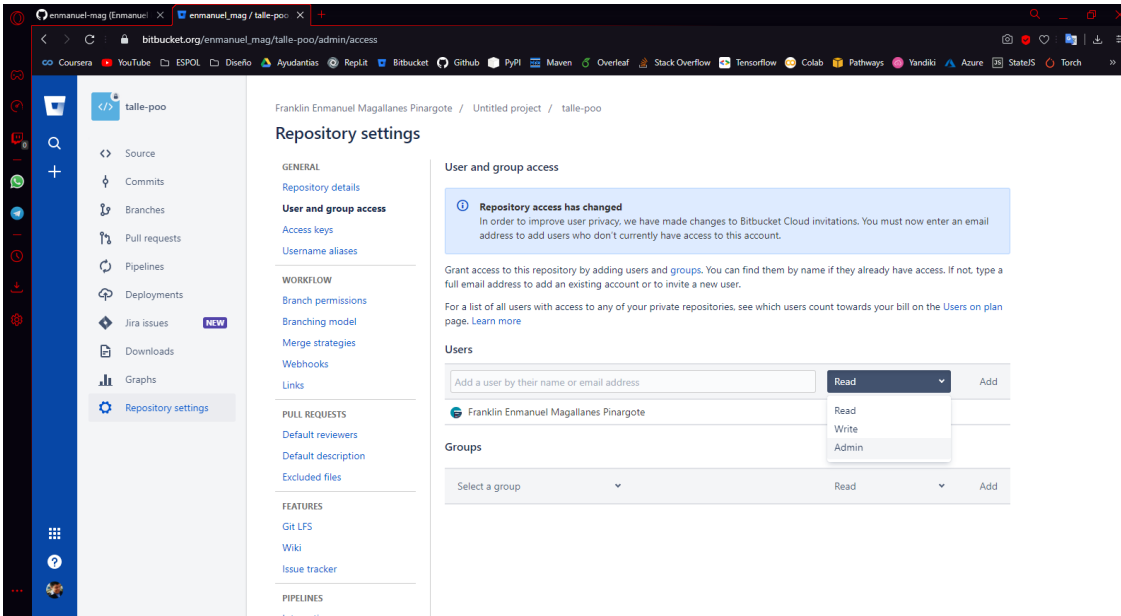


19. En la siguiente ventana veremos los cambios que vienen de “afuera” (repositorio remoto, para identificado nos muestra abajito un texto con la fuente, en este caso origin/master) y los cambios locales de nuestra computadora (lado derecho, muestra el texto de nuestro commit que colocamos hace unos segundos). Para aceptar uno u otro tenemos que dar clic en uno de los dos botón **Accept** que esta arriba de cada mini preview tanto del lado izquierdo como derecho. También podemos aceptar ambos cambios y se juntan ambas líneas. Debajo de las dos preview de los cambios veremos otra preview que muestra como como se verá nuestro código al hacer uno o ambos cambios. Finalmente, cuando decidimos que cambios conservar damos clic en el botón **Ok** de la parte inferior derecha.



20. Luego vamos a **Teams > Add**. Después **Teams > Commit** veremos que se carga un texto por defecto explicando que se está haciendo un merge para solucionar los conflictos, damos clic en **Commit**.
21. Por fin, damos click en **Teams > Remote > Pull to UpStream**, y con esto ya habremos actualizados los cambios con resolución de un conflicto.

22. Para añadir un nuevo miembro al repositorio tenemos que ir a Repository Settings:



23. En la opción **User and grup Access** podemos añadir a un nuevo miembro ya sea por su correo o username y setear su rol en repositorio, para que pueda hacer push y modificar el repo deben colocarlo como Admin preferiblemente.

24. Ahora para el nuevo miembro para clonar el repositorio al que fue invitado debe copiar el link de repositorio, el cual es el mismo que copiamos en el **paso 8**.

25. Para añadir el repositorio localmente con NetBeans tenemos dos casos:

- Si tenemos ya un proyecto con algún repositorio inicializados encontremos la opción para clonar en **Teams > Remote > Clone..**
- Si no tenemos algún proyecto abierto lo encontraremos en **Teams > Git > Clone...**

Con esto veremos esta ventana, en el cual en el campo Repository URL colocaremos el link de repo y mas abajo seleccionaremos el Path o la ruta donde queremos que se ubique este proyecto. Uno vez colocado eso damos clic en **Finish**. Luego se nos mostrará una ventana preguntando si queremos crear un proyecto con ese repositorio, damos clic en **Create project**, con esto NetBeans no pedirá la información de este proyecto, pareciera que estuviéramos creando un nuevo proyecto en el cual tendremos que copiar la carpeta **src** que se clonó, dentro de la carpeta del proyecto que acabamos de crear. Con esto ya tendremos el repositorio local y el mismo código del del repo remoto.