

Práctica 4.

Netbeans y GitHub

Repositorios remotos

Los repositorios remotos son versiones de tu proyecto que se encuentran alojados en Internet o en algún punto de la red.

Puedes tener varios, cada uno de los cuales puede ser de sólo lectura, o de lectura/escritura, según los permisos que tengas.

Colaborar con otros implica gestionar estos repositorios remotos, y mandar (push) y recibir (pull) datos de ellos cuando necesites compartir cosas.

Gestionar repositorios remotos implica:

- conocer cómo añadir repositorios nuevos
- eliminar aquellos que ya no son válidos
- gestionar ramas remotas e indicar si están bajo seguimiento o no
- ..y más cosas

Puedes ampliar los conceptos y los comandos Git para gestión remota en:

http://librosweb.es/libro/pro_git/capitulo_2/trabajando_con_repositorios_remotos.html

En esta práctica vamos a presentar estos conceptos con la ayuda de GitHub como servidor remoto y Netbeans con Git como IDE local

GitHub

GitHub es una plataforma online basada en Git, que nos permite almacenar nuestros repositorios Git en sus servidores.

1. Crear una cuenta en GitHub <https://github.com/>

Create your personal account

Username

This will be your username — you can enter your organi

Email Address

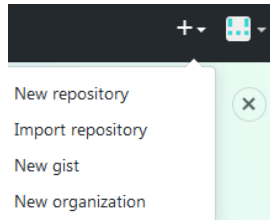
You will occasionally receive account related emails. We email with anyone.

Password

Use at least one lowercase letter, one numeral, and seve

2. Crear un Repositorio en GitHub

Debemos crear un repositorio en GitHub para nuestro proyecto. Clic en el menú (+) y elige la opción “**New repository**”



A continuación debemos completar el formulario que se nos presenta

Create a new repository

A repository contains all the files for your project, including the revision history

Owner: / Repository name:

Great repository names are short and memorable. Need inspiration? How at

Description (optional):

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if repository.

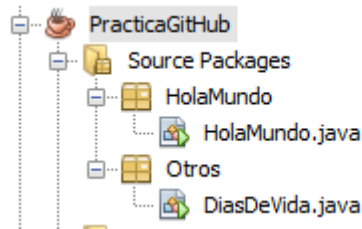
Add .gitignore: Add a license: ⓘ

Create repository

- Le damos un nombre al repositorio (en nuestro caso NetbeansGitHub)
- Escribimos una descripción del proyecto
- Escogemos si queremos que el repositorio sea público o privado. Privado solo para la versión de pago.
- El archivo **.gitignore** nos permite indicar que archivos queremos ignorar, es decir estos archivos no se subirán al repositorio. Para este ejemplo dejamos en **NONE**
- Añadimos una licencia de uso a nuestro proyecto, por ejemplo **GNU General Public License v2.0**, si no quieres asignar ningún tipo de licencia, tienes también la opción de NONE
- Para terminar clic en el botón “**Create repository**” y el repositorio se creara en unos pocos segundos.

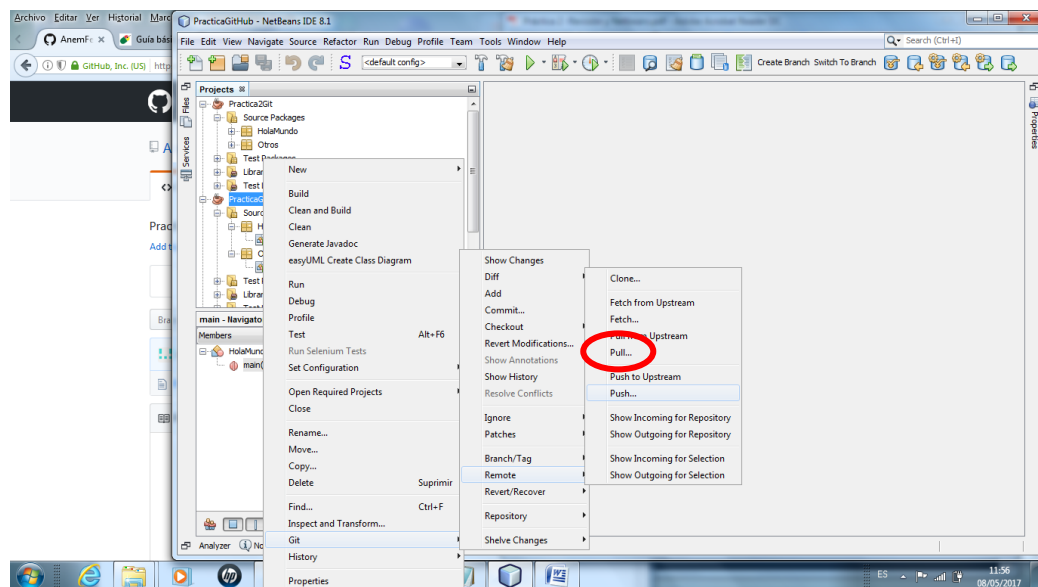
3. Subir un Proyecto NetBeans a GitHub

Tenemos un proyecto (en el ejemplo el proyecto PracticaGitHub) desarrollado en Netbeans, queremos subirlo al repositorio remoto de GitHub

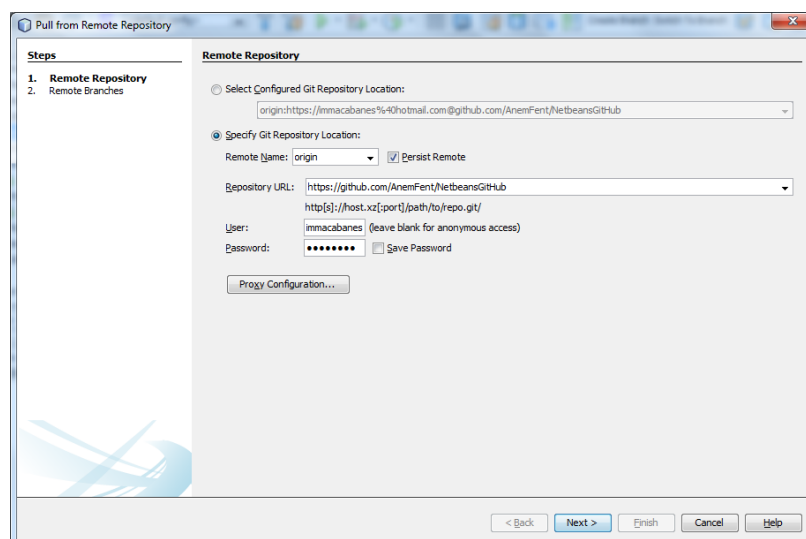


3.1. Desde Netbeans hacemos un pull para mezclar el repositorio de GitHub con el repositorio local.

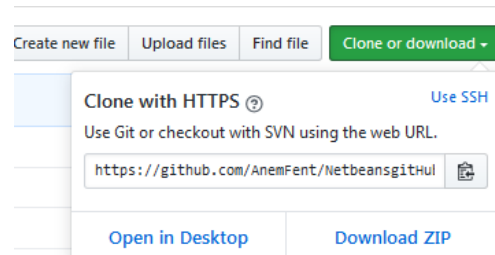
Para esto clic derecho sobre el proyecto, entonces **GIT -> Remote -> Pull**



Debemos configurar los parámetros de conexión de la siguiente manera:

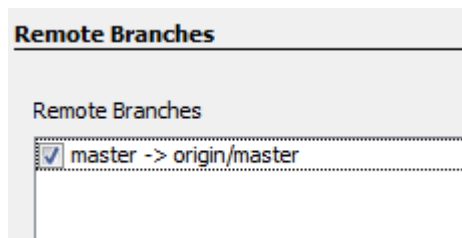


- Seleccionamos “**Specify Git Repository Location**”
- Escribimos el nombre para el repositorio que mezclaremos con el repositorio local
- Pegamos la URL del repositorio GitHub
 - Si no la sé...
 - Puedo solicitarla a GitHub :



- Identificamos usuario y password
- presionamos NEXT

- para continuar, seleccionamos la Rama master y presionamos **Finish**.

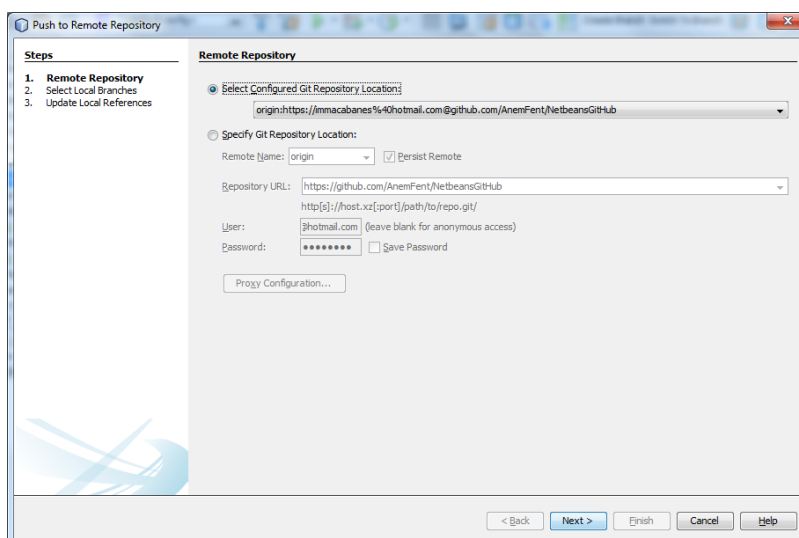


- En la ventana *OUTPUT* de NetBeans podremos ver que la mezcla se realiza con éxito

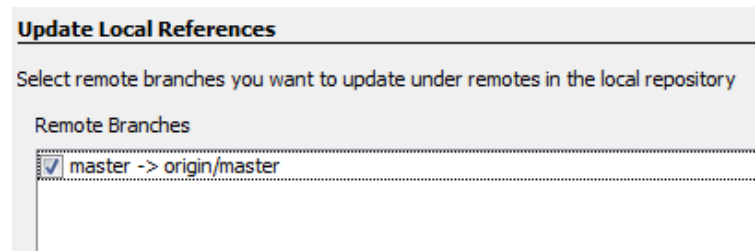
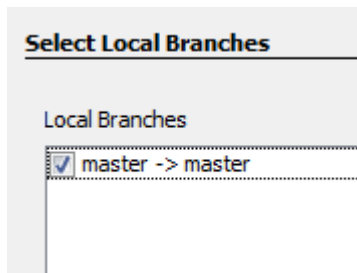
3.2. Subir el proyecto al repositorio

Ahora que hemos mezclado ambos repositorios, estamos listos para subir el repositorio local a *GitHub*.

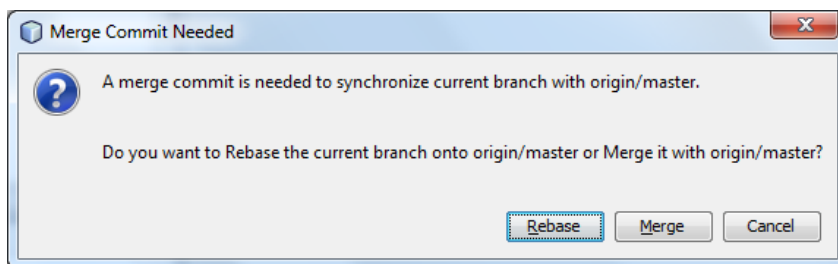
Clic derecho sobre el proyecto -> **Git -> remote -> push**,



- se abre nuevamente la ventana de configuración
- los datos de conexión ya están llenos
- solo nos queda presionar **NEXT**, seleccionar el recuadro que aparece y presionar **Finish**.



Nos pregunta: Merge (mezclamos)



Nuevamente podemos revisar en la ventana **OUTPUT** de NetBeans que los archivos se subieron al repositorio.

3.3. En GitHub podemos ver que en nuestro repositorio se encuentran los archivos que acabamos de subir

4. Importar un proyecto ya existente en GitHub

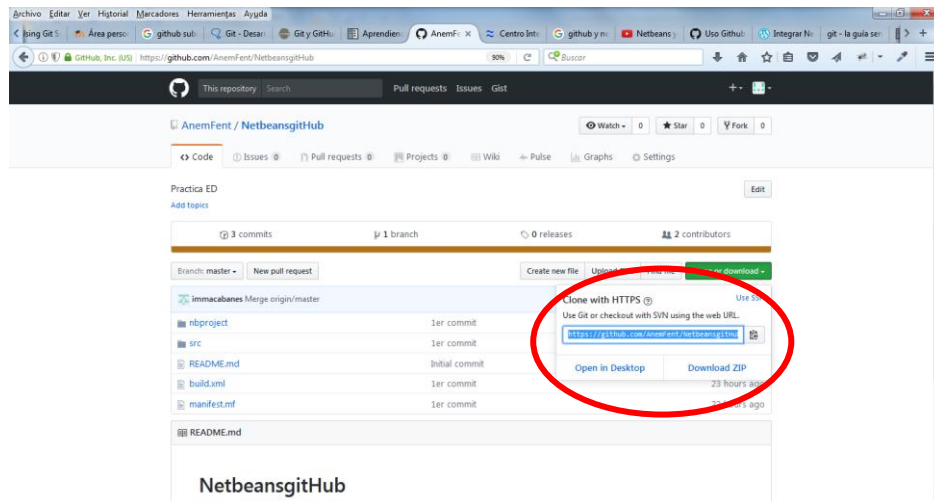
Ahora partimos de un Repositorio ubicado en GitHub y queremos clonarlo en nuestra máquina local

Atención. En el ejemplo:

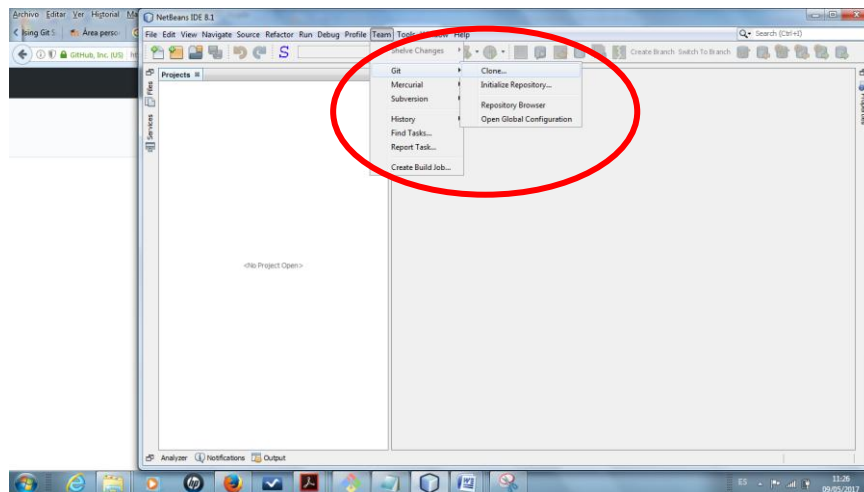
El Repositorio se llama NetBeansGitHub

El Proyecto Netbeans guardado en ese Repositorio se llama PracticaGitHub

Siempre puedo solicitar a GitHub la URL del Repositorio que quiero clonar:

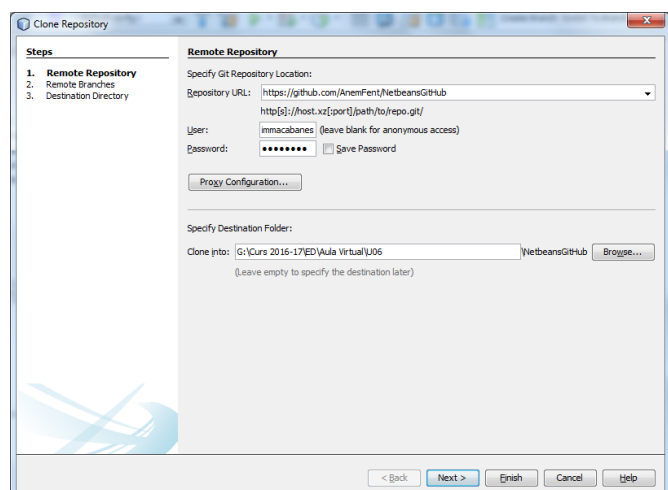


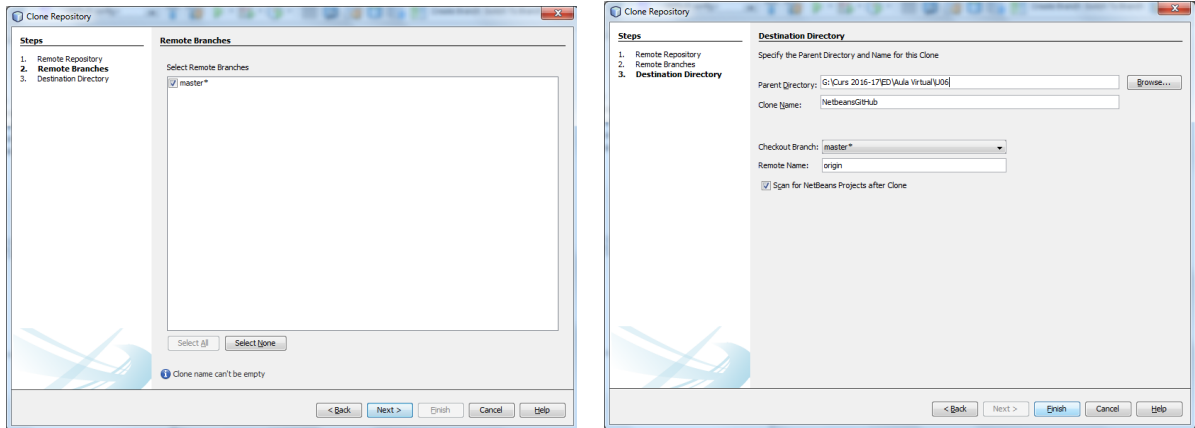
Desde NetBeans: Team > Remote > Clone



Como antes, configuramos los parámetros de conexión...

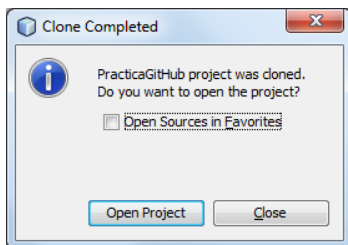
El clonado puede ser anónimo si el repositorio es público.



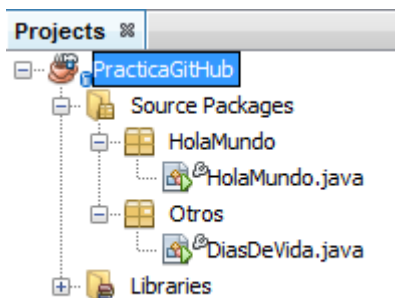


Ojo: La carpeta por defecto es el nombre del Repositorio, no del Proyecto

(Si trabajamos en Netbeans conviene pues que repositorio y proyecto se llamen igual)



Y ya tenemos el repositorio copiado y activo Git

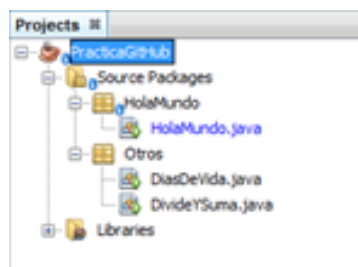
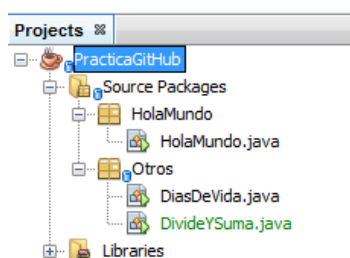


5. Modificar un proyecto clonado del remoto y volver a subirlo a GitHub

Vamos a modificar el proyecto clonado. Por ejemplo,

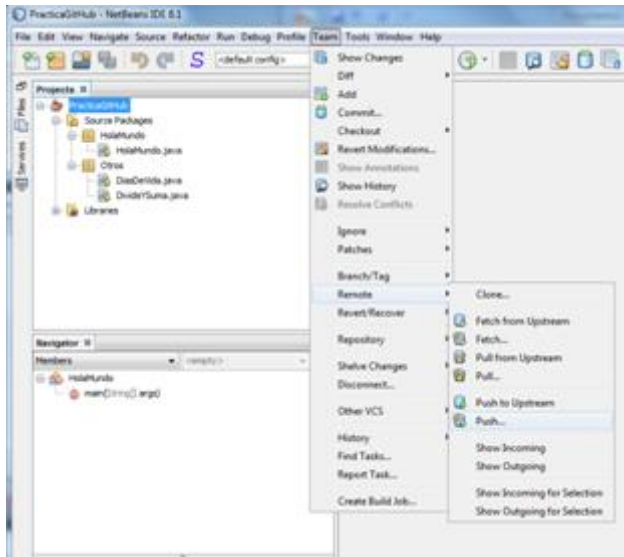
Añadimos ficheros al proyecto (aparecen en verde)

...o modificamos ficheros (aparecen en azul)

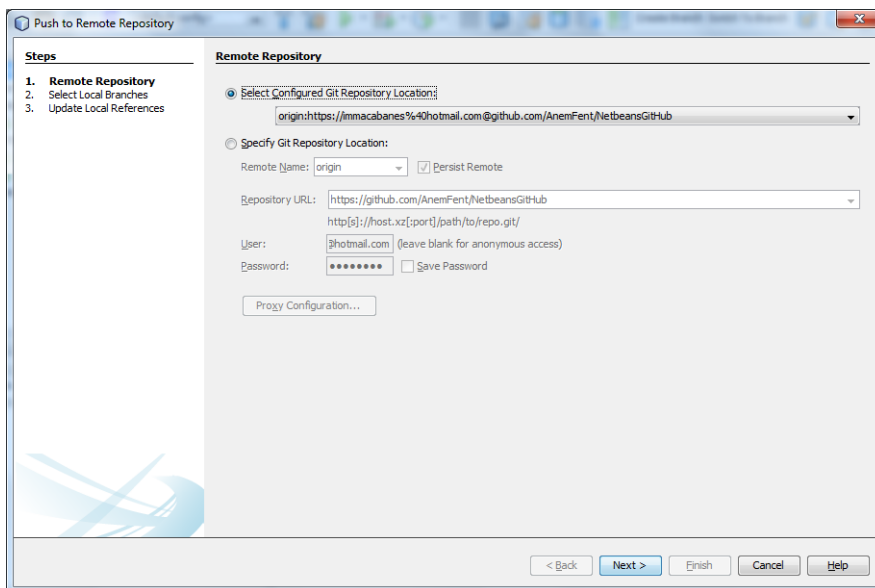


Antes de subirlo, hacemos **add y commit**.

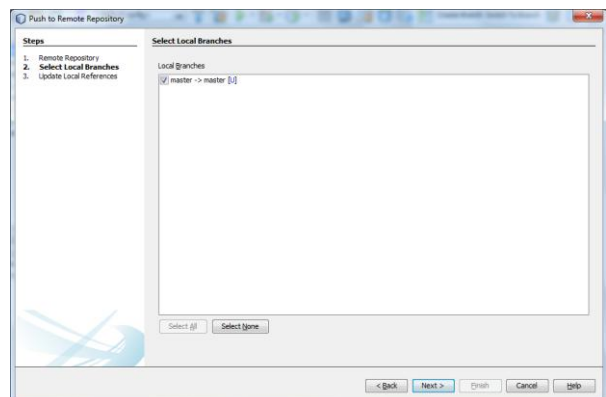
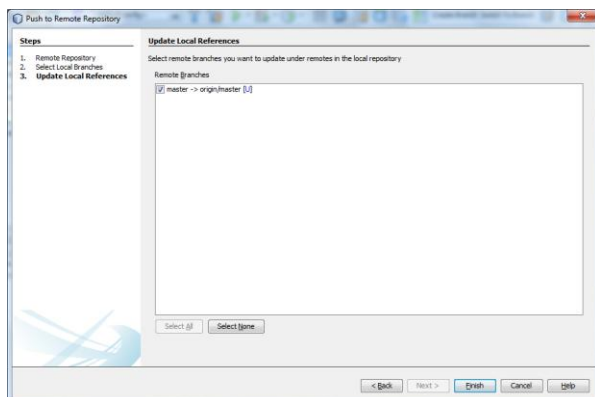
Después, hacemos un **push**



Como siempre ubicamos el repositorio remoto:



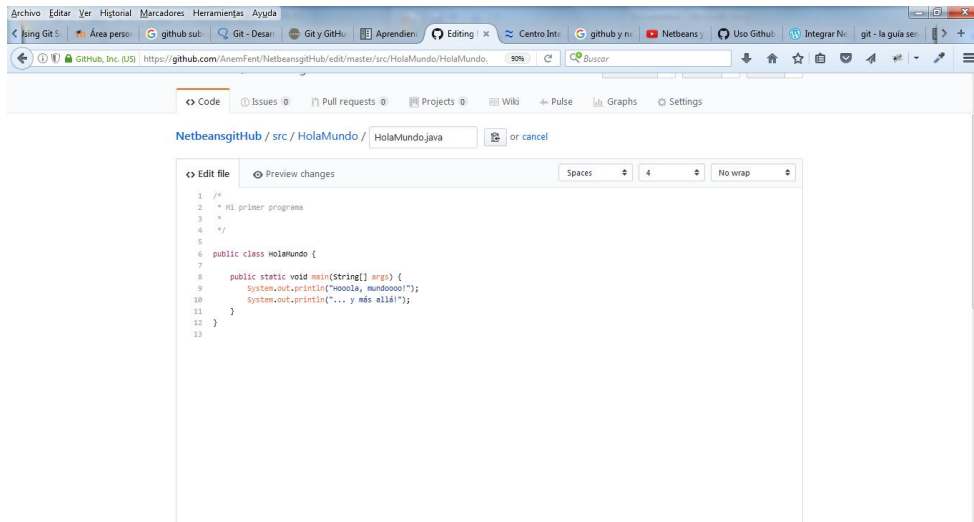
Seleccionamos Rama:



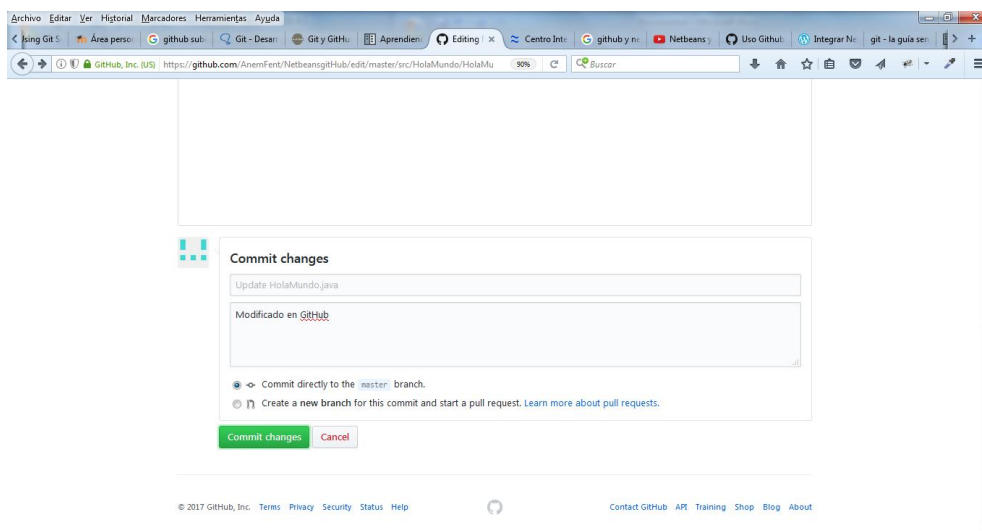
Vemos si se ha subido correctamente a GitHub.com. Efectivamente:

6. Proceso inverso: modificamos en github.com y lo traemos a local

Modificamos ficheros del repositorio remoto



Hacemos un commit



Ahora, desde NetBeans hacemos pull como antes....

...y Comprobamos que queda modificado en local