

TUTORIAL ECLIPSE GIT

El uso de Eclipse Git en nuestros proyectos es cada día más habitual. Git se ha convertido como software de control de versiones. Con Eclipse es muy sencillo integrar nuestro proyecto en un repositorio Git.

En este tutorial a través de un ejemplo vamos a integrar nuestro proyecto en un repositorio Git.

Vamos a utilizar Github.

CREAR REPOSITORIO

1. Ir a www.github.com
2. Crear una cuenta gratuita. Si ya tenéis entrad en vuestra cuenta.
3. Crear un nuevo repositorio haciendo clic en el botón NUEVO.



4. Ingresar datos e información inicial del repositorio:
 - a. Elegir un nombre para el repositorio.
 - b. Opcionalmente ingrese una descripción.
 - c. Indicar si desea que sea público o privado (en este tutorial usaremos uno público).
 - d. No marcamos ninguna opción porque queremos un vacío.
 - e. Botón Crear Repositorio.

Crea un nuevo repositorio

Un repositorio contiene todos los archivos del proyecto, incluido el historial de revisiones. ¿Ya tienes un repositorio de proyectos en otro lugar? [Importar un repositorio](#).

Propietario: / Nombre del repositorio:

Los grandes nombres de repositorios son breves y fáciles de recordar. ¿Necesitas inspiración? ¿Qué hay de la palmera super-duper-octo-palmera?

Descripción (opcional):

☒ Público
Cualquiera en Internet puede ver este repositorio. Tú eliges quién puede comprometerse.

☐ Privado
Tú eliges quién puede ver y comprometerse con este repositorio.

Inicialice este repositorio con:
Omita este paso si está importando un repositorio existente.

☐ Agregar un archivo README
Aquí es donde puede escribir una descripción larga de su proyecto. [Aprenda más.](#)

☐ Agregar .gitignore
Elija qué archivos no rastrear de una lista de patrones. [Aprenda más.](#)

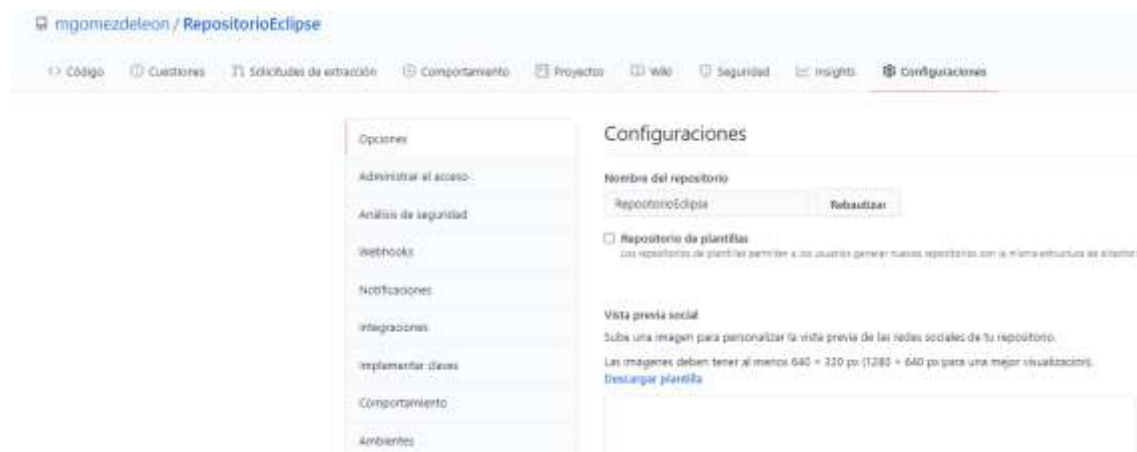
☐ Elija una licencia
Una licencia le dice a otros lo que pueden y no pueden hacer con su código. [Aprenda más.](#)

5. Aparece esta pantalla que nos confirma que el repositorio se ha creado.



6. Invitar a nuestros compañeros de equipo como colaboradores en el repositorio:

- Seleccionar pestaña Configuración
- En el menú de la izquierda elegir la opción Administrar el acceso



c. Pinchar en el botón Invita a un colaborador



d. Introducir el nombre de los colaboradores



- Una vez completada esta acción, los colaboradores recibirán un email para aceptar la invitación que les hemos enviado.



USO DE GIT EN IDE

Utilizamos Eclipse IDE for java Developers que ya cuenta con EGit incluido. En caso de no contar con esta versión de eclipse puede agregarse el plugin de EGit.

Para ver este ejemplo voy a utilizar varios usuarios diferentes (Marta, Ceinmark).

Con el usuario Ceinmark:

CREAR, INICIALIZAR Y CARGAR(PUSH) UNA VERSIÓN INICIAL DE NUESTRO PROYECTO

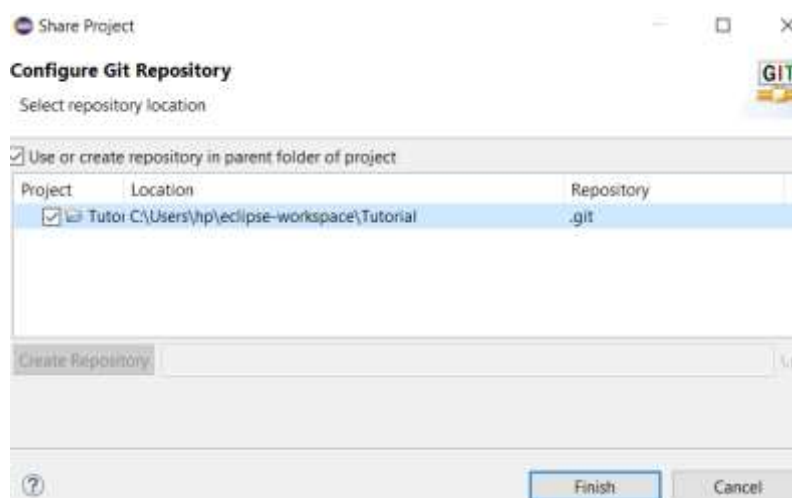
- Para este caso he creado usando Eclipse un proyecto llamado Tutorial, cuenta con una clase Hola que muestra por consola "Hola, soy Ceinmark".

```
Hola.java
1
2 public class Hola {
3
4     public static void main(String[] args) {
5         System.out.println("Hola, soy Ceinmark");
6     }
7
8 }
9
```

- Crear e inicializar el repositorio:
 - Haced clic botón derecho sobre el proyecto (Tutorial)
 - Opción Team > Share Project

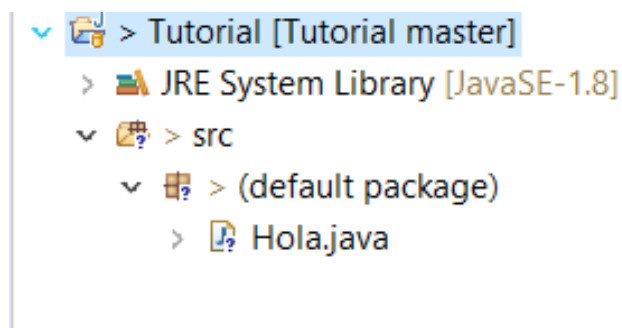


3. Seleccionar la ubicación de nuestro proyecto para el proyecto de Git y creamos un repositorio allí.
 - a. En esta ventana seleccionamos la opción Use or create repository in parent folder of Project
 - b. Seleccionar en la lista el proyecto y su ubicación
 - c. Hacer clic en Create Repository
 - d. Clic en Finish



4. Vemos en el Package Explorer que junto a los íconos del proyecto se muestra un pequeño cilindro amarillo y junto a la carpeta src, el package tutorial y el archivo Hola.java se muestra un ícono de ?

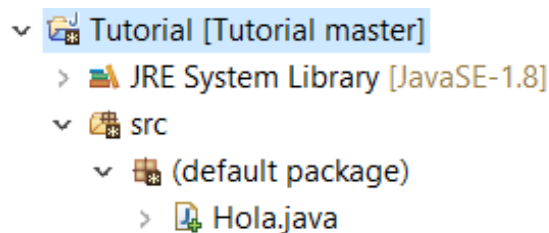
Esto indica que el proyecto tiene un repositorio git asociado, pero existen cambios sin agregar al index de git.



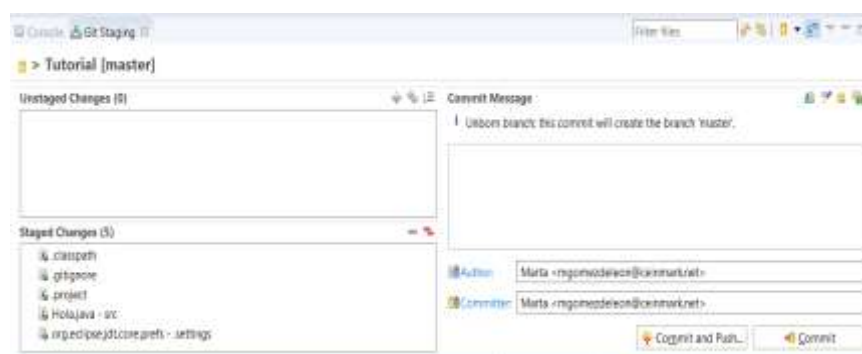
5. Agregar los cambios en index:
 - a. Clic con el botón derecho en el proyecto Tutorial
 - b. Opción Team > Add to Index



- Vemos en el Package explorer que los íconos de antes ahora cambiaron por un * blanco (**significa cambios en la estructura**) en el proyecto, carpeta source y package y un + verde (**significa agregado**) en el archivo Hola.java



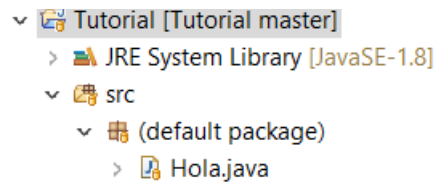
6. A continuación, commitear los cambios a git:
 - a. Haced clic botón derecho sobre el proyecto Tutorial
 - b. Vamos a la opción Team > Commit
 - c. Aparece la pestaña Git Staging



En este punto podemos ver a la izquierda en la parte inferior los archivos modificados que se encuentran en el index (Staged Changes) y en la parte superior los archivos modificados que NO se han agregados al index (Unstaged Changes). Al hacer commit sólo se incluirán los cambios que si se hayan agregado al index.

A su vez en Commit Message DEBEMOS agregar un comentario que nos de una idea de que cambio se realizó.

- Luego de esto podemos hacer clic en Commit and Push o Commit. Haced commit para mostrar cómo proceder si falla el push o no se cuenta con conexión en ese momento.
- Una vez que el presionamos Commit vemos en el Project/Package explorer.



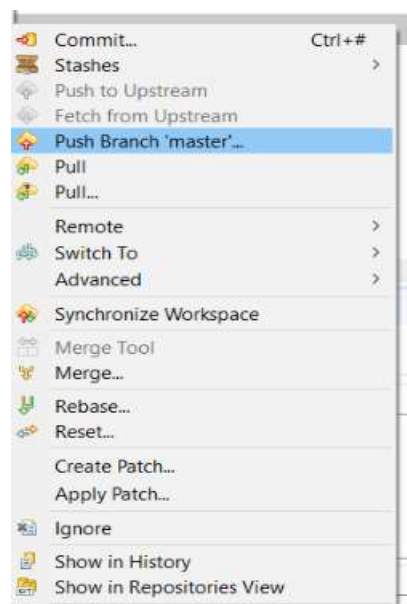
7. Ahora tenemos que hacer push para llevar los cambios del proyecto a Github.

- a. Nos dirigimos a la página del repositorio, allí seleccionamos la opción para la url del repo. Usaremos https por simplicidad, si tienen configurado una key ssh puede usar la opción ssh.
- b. Seleccionamos la opción HTTPS y hacemos clic en el botón a la derecha para copiar la url o la copiamos a mano.



- c. Hacemos clic derecho sobre el proyecto.
- d. Seleccionamos la opción Team > Push Branch 'master'...

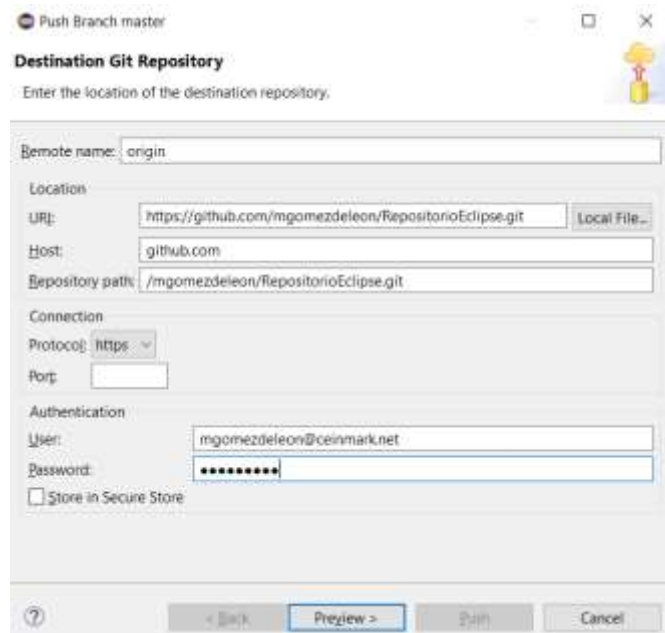
Nota: Sólo la primera vez, las siguientes veces , debemos usar Push to Upstream



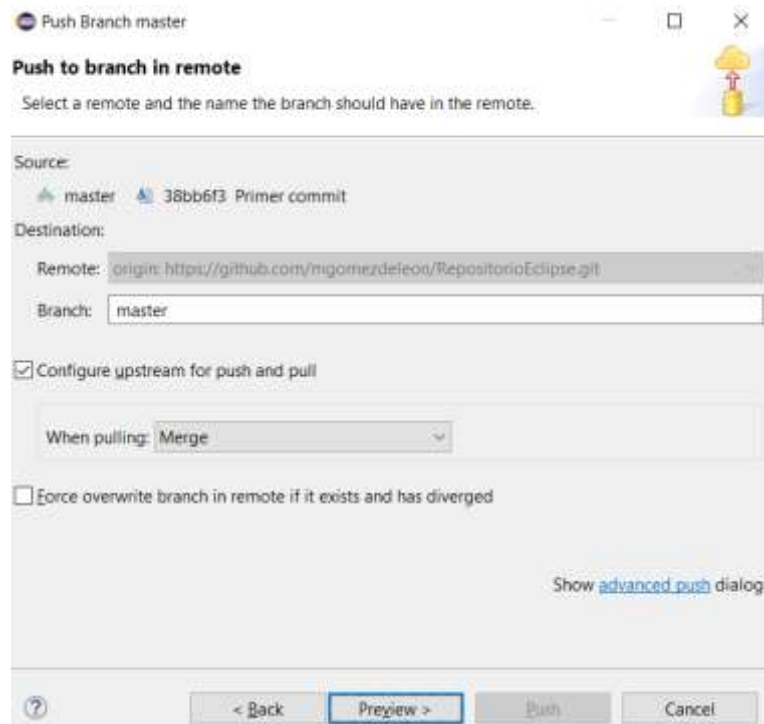
Aparece una ventana para configurar el repositorio remoto:

- En el Remote name dejamos origin por defecto.

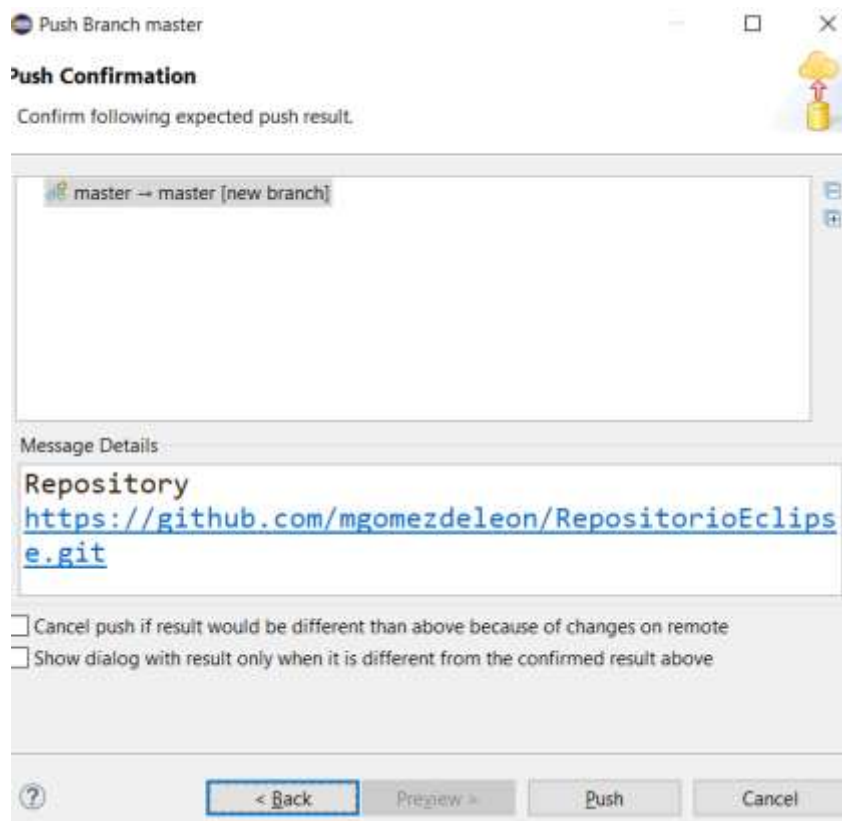
- Pegamos allí la url del repositorio que copiamos antes, se autocompletará casi toda la información.
- Ingresamos usuario y contraseña de github.
- Hacer click en Preview >



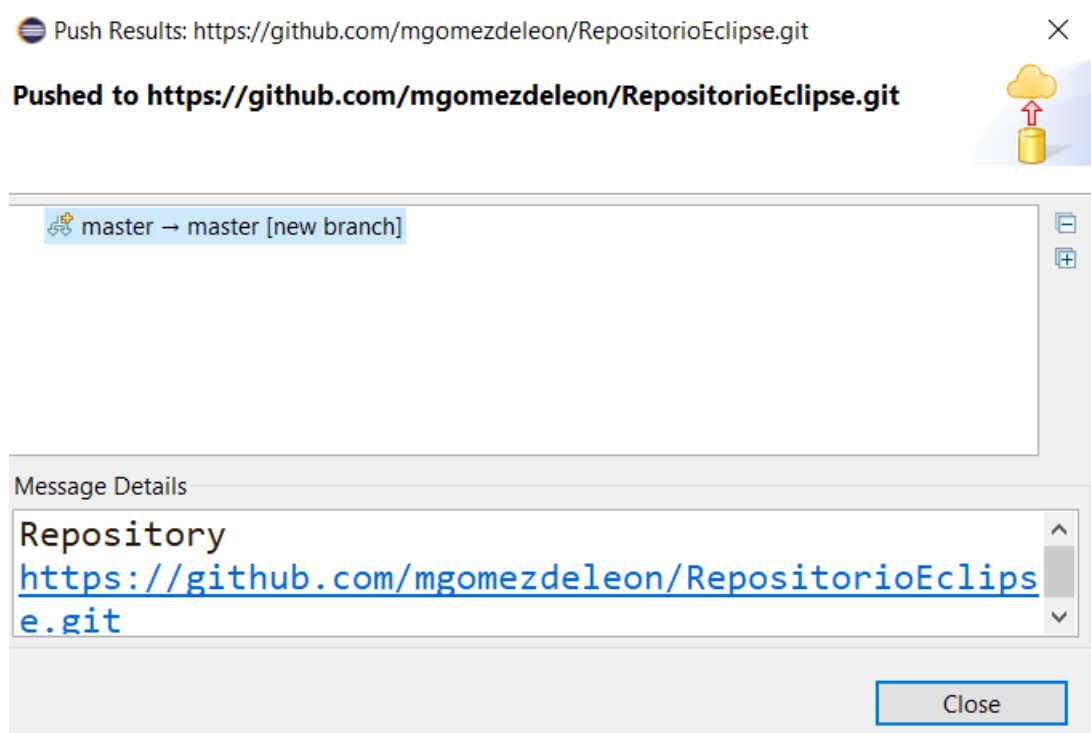
- Configurar como Branch(rama) remota para sincronizar el mismo nombre (en este caso master)
- Seleccionar la opción Configure upstream for push and pull.
- Elegir la acción default a realizar el pull como Merge
- **NO** seleccionar el Force overwrite branch in remote if it exists and has diverged para evitar perder los cambios realizados por otros.
- Pulsar en Preview



- Nos aparece la confirmación del push.



- Clic en el botón en Close.



Id a Github y comprobar que se ha subido bien el proyecto Tutorial:



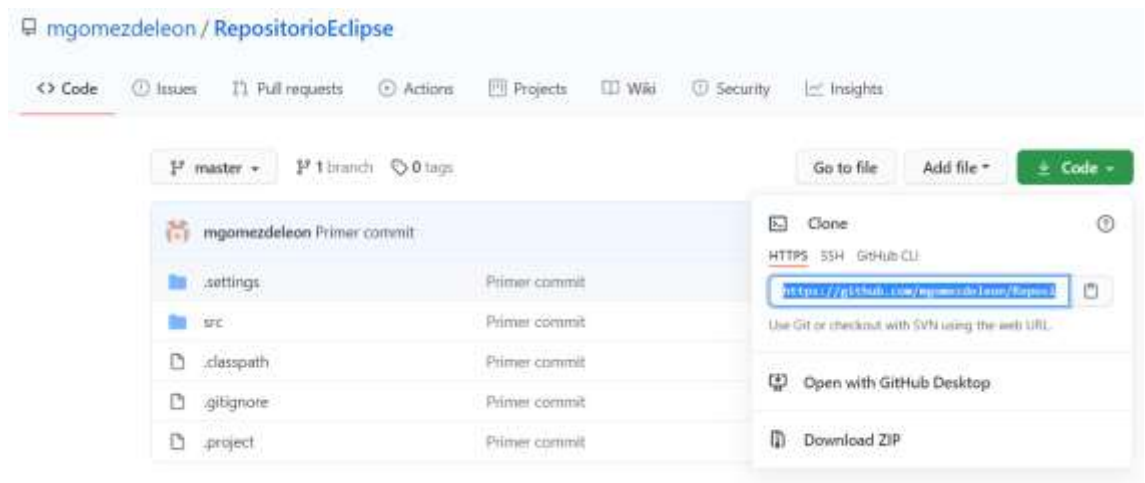
CLONAR EL REPOSITORIO, REALIZAR CAMBIOS Y CARGARLOS (PUSH)

Para este caso abrimos eclipse en un workspace diferente para simular otro usuario.

Con el usuario **Marta**

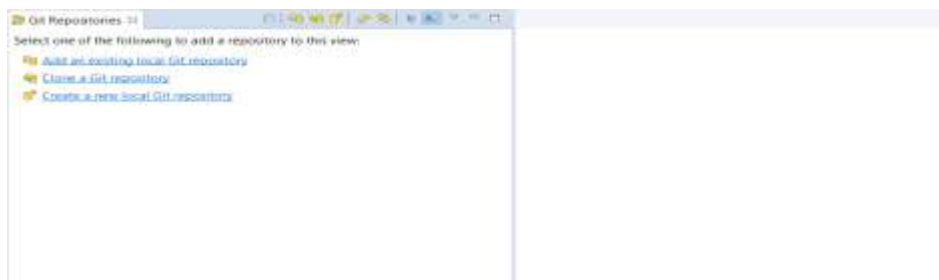
1. Entrar en el correo electrónico para aceptar la invitación como colaborador.
2. Ir a [www.github.com](https://github.com), acceder al repositorio y copiar la url del repositorio para clonar.

- Clic en el botón Code
- Elegimos la opción HTTPS.
- Copiamos el enlace.



3. Clonamos el repositorio en Eclipse:

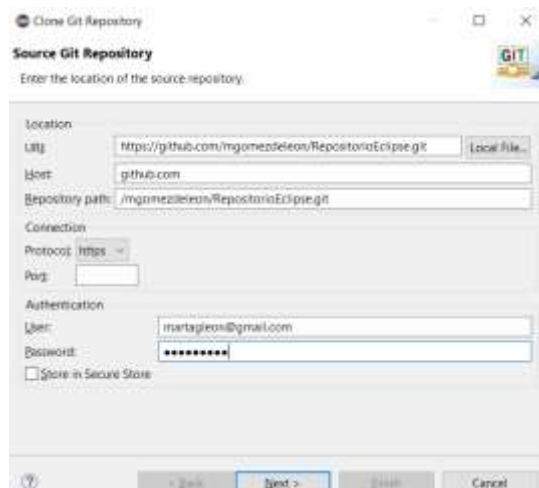
- Ir al menú Window > Perspective > Open Perspective > Other > Git
- Y aparece esta pestaña al lado izquierdo:



Haced clic en el botón de Clone a Git repository



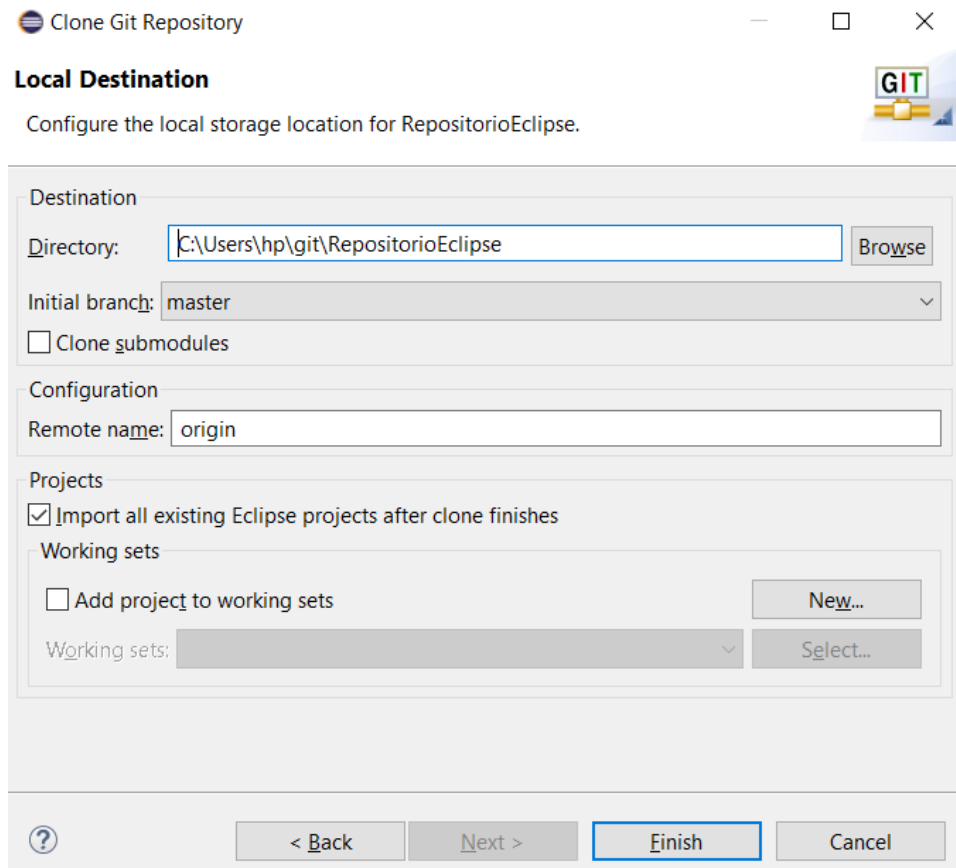
- Se abrirá la siguiente ventana
- Pegar la url que copiamos del repositorio donde indica URI
- Completar los datos de usuario y contraseña
- Hacer clic en Next



- Nos muestra la lista de branch que deseamos clonar.
- Dejamos seleccionada master.
- Clic en Next



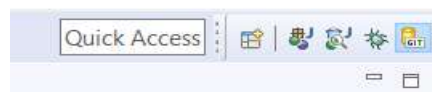
- En la siguiente ventana:
- En Destination > Directory reemplazamos la ubicación propuesta por el directorio donde se creará el proyecto, es decir la ubicación del workspace que utilizamos en eclipse
- Seleccionamos master como la rama inicial
- Para la definición del nombre del repositorio remoto (el que se encuentra en github) usaremos origin
- IMPORTANTE: seleccionamos la opción import all existing Eclipse projects after clone finishes, para asegurarnos que el proyecto se agregue al Project/Package explorer
- Clic en Finish



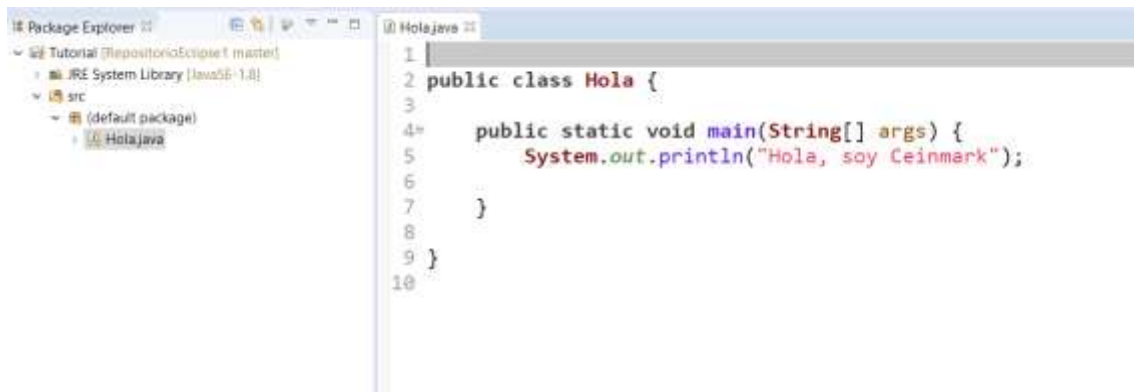
Nos aparece el repositorio agregado a la lista de repositorios de la pestaña Git Repositories:



En la parte superior derecha, elegimos la perspectiva de Java o Java EE para ver el proyecto y el código fuente

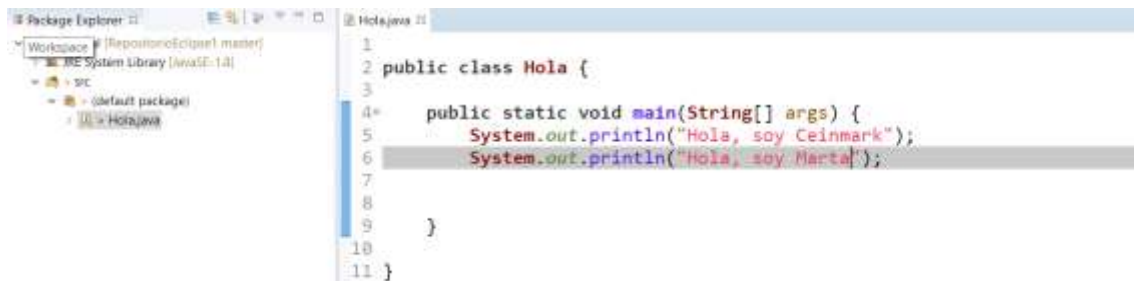


Ahora vemos nuestro proyecto:



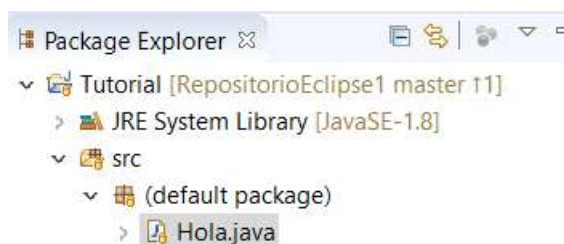
4. Realizar cambios:

- Editar y modificar el código. Hasta que no guardemos los cambios git no muestra cambios en los archivos del Package explorer ni la pestaña de git staging
- Guardar los cambios en el Package Explorer y aparece el signo de > .



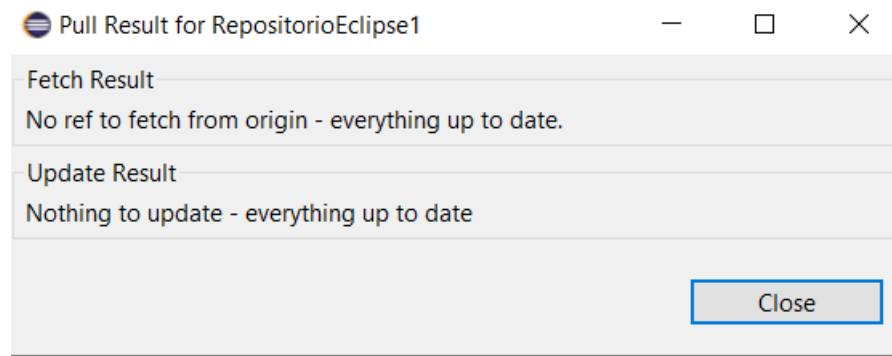
- Realizar los pasos siguientes (visto anteriormente)
 - a. Clic con el botón derecho en el proyecto Tutorial
 - b. Opción Team > Add to Index
 - c. Hacer clic botón derecho sobre el proyecto Tutorial
 - d. Vamos a la opción Team > Commit
 - e. Aparece la pestaña Git Staging
A su vez en Commit Message DEBEMOS agregar un comentario que nos dé una idea de que cambio se realizó.
 - f. Presionar Commit

En Package explorer el archivo y la estructura pasan a tener nuevamente un cilindro amarillo junto al ícono y junto al proyecto vemos una flecha en dirección ascendente con un 1 a la derecha. Esto indica que localmente hemos hecho un commit que no hemos pushado aún.

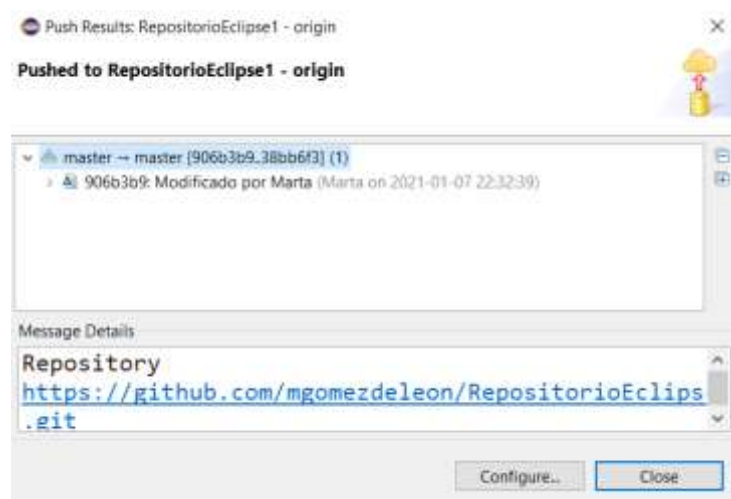


- g. Clic opción Team > Pull

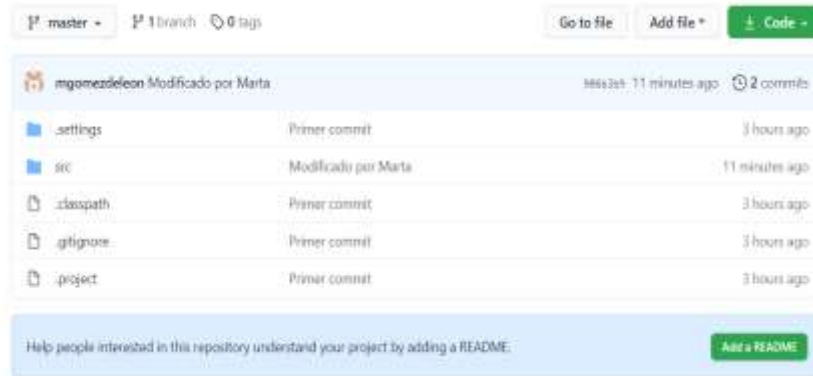
Nota: no es necesario realizar un pull antes de push, pero en caso que haya cambios en el repositorio remoto hecho por otro compañero si no hacemos pull antes del push veremos un mensaje de error y deberemos hacer pull antes del push.



- h. Clic opción Team > Push to upstream



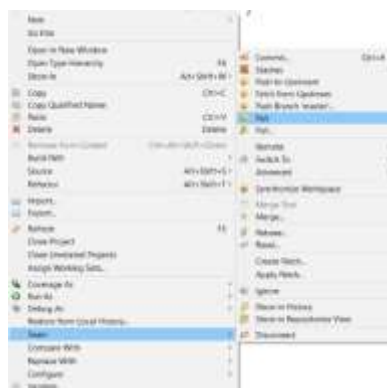
Si accedemos al Github, vemos que ha sido modificado:



OBTENER CAMBIOS REALIZADOS POR OTRO MIEMBRO DEL EQUIPO

Accedo a Eclipse en el workspace original para simular nuestro primer usuario (Ceinmark) y compruebo qué el código está sin modificar.

1. Sobre el proyecto botón derecho Team > Pull



Y aparece la siguiente ventana:



Al pinchar en Close, nos aparece el código modificado:

```

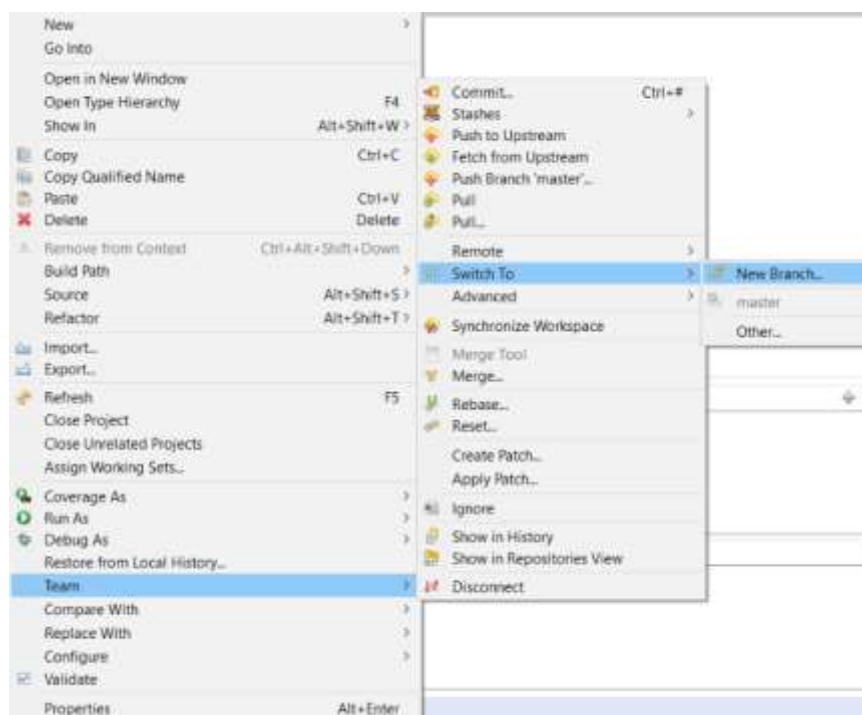
1
2 public class Hola {
3
4     public static void main(String[] args) {
5         System.out.println("Hola, soy Ceinmark");
6         System.out.println("Hola, soy Marta");
7
8     }
9
10 }

```

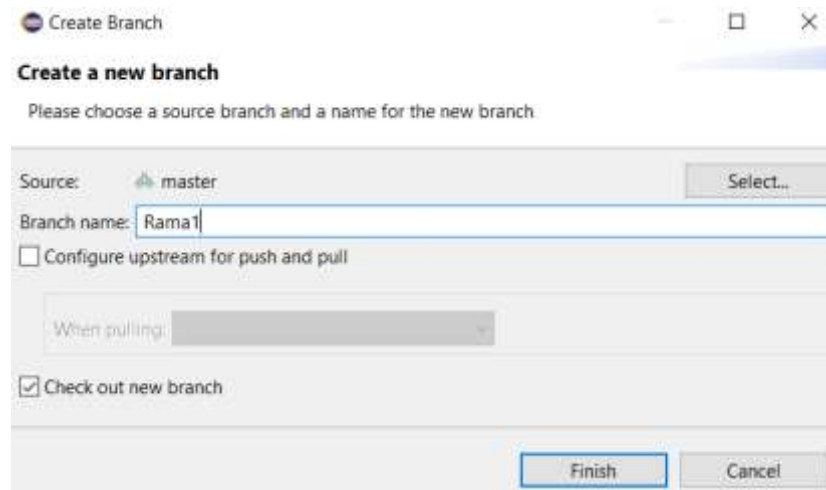
CREAR UNA NUEVA RAMA

Es útil para no trabajar con la rama maestra cuando estamos desarrollando nuevas funcionalidades de nuestro código. Para ello crearemos una nueva rama para cada una de las funcionalidades que estemos desarrollando, una vez superadas las fases de pruebas esta rama podrá ser unida a la rama maestra.

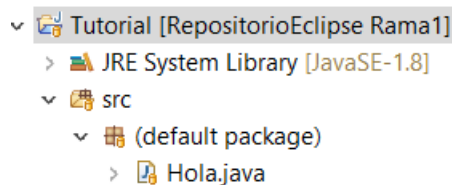
1. Partimos de nuestro proyecto en Eclipse
2. Botón derecho sobre nuestro proyecto Team > Switch To > New Branch



3. Ponemos un nombre a la nueva Rama:



4. Aparece en la parte de Package Explorer el nombre de la nueva rama (Rama1)

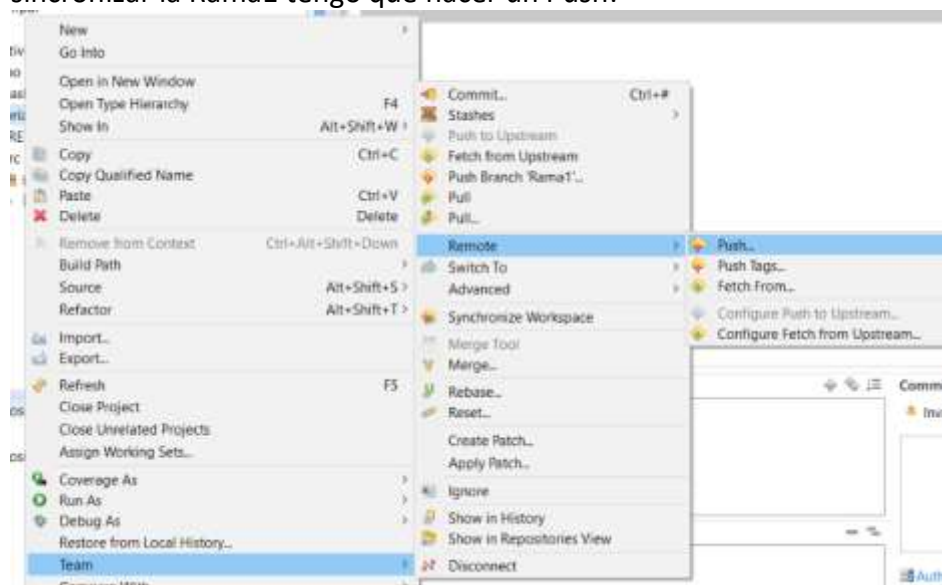


(La rama Maestra y la Rama Rama1 contiene el mismo código)

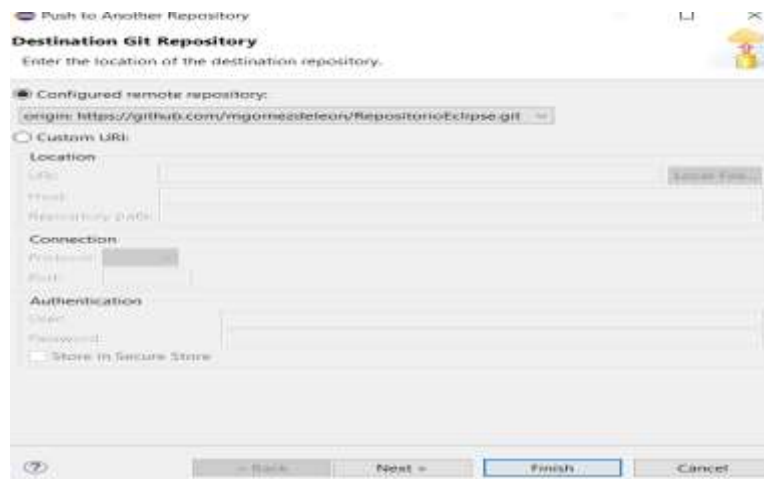
Nota: Si queremos volver a la rama maestra: Team > Switch to > master

Si vamos a Github vemos que sigue apareciendo sólo una rama que es la Master.

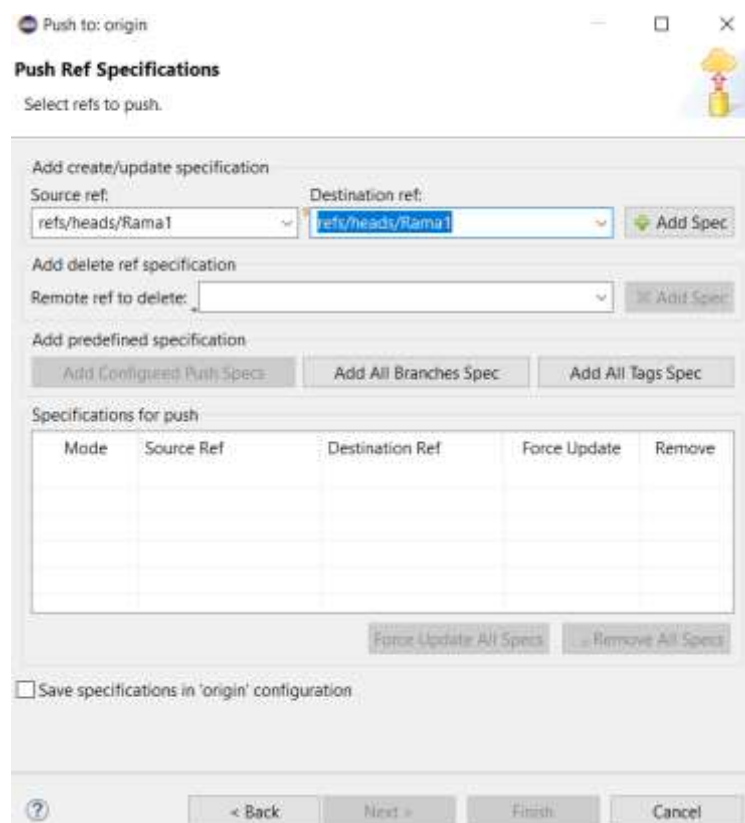
5. Si vamos a Github vemos que sigue apareciendo sólo una rama que es la Master. Para sincronizar la Rama1 tengo que hacer un Push:



6. Y nos muestra esta pantalla: y damos en Next



7. Introducimos el origen y el destino.
Y pinchamos en +Add spec.
Botón Next



8. Vamos al Repositorio Remoto Github y comprobamos si se ha subido:

Maestro ▾ 2 ramas 0 etiquetas Ir al archivo

mgomezdeleon Modificado por Marta	
.settings	Primer compromiso
src	Modificado por Marta
.classpath	Primer compromiso
.gitignore	Primer compromiso
.proyecto	Primer compromiso

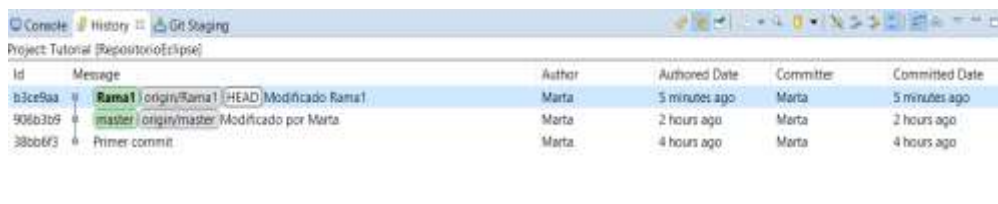
Avude a las personas interesadas en este repositorio a comprender su proyecto agregando un archivo

Hay dos ramas: Master y Rama1.

MODIFICAR LA NUEVA RAMA

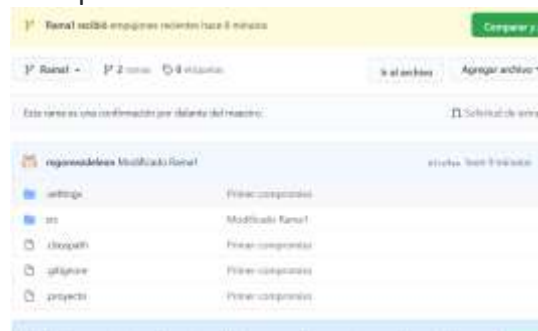
1. Ahora puedo trabajar sobre la Rama1 y hacer modificaciones en el código desde Eclipse y luego lo subiremos a GitHub (commit y push)

Nota: Para ver todos los commit que hemos realizado: Team > Show in History



Id	Message	Author	Authored Date	Committer	Committed Date
b3ce9aa	Rama1 origin/Rama1 HEAD Modificado Rama1	Marta	5 minutes ago	Marta	5 minutes ago
906b3b6	master origin/master Modificado por Marta	Marta	2 hours ago	Marta	2 hours ago
382b6f3	Primer commit	Marta	4 hours ago	Marta	4 hours ago

2. En Github, comprobamos que se han realizado los cambios en la Rama1:



Rama1 recibirá empujes recientes hace 5 minutos

Rama1 - 2 ramas 0 etiquetas Ir al archivo Agregar archivo

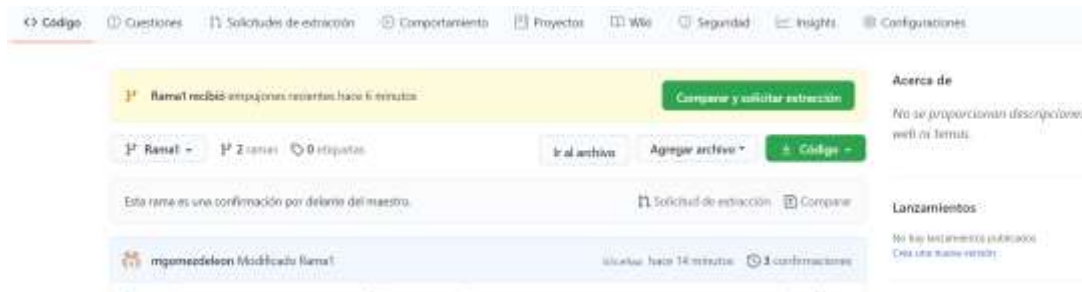
Este ramo es una confirmación por delante del maestro.

mgomezdeleon Modificado Rama1	
.settings	Primer compromiso
src	Modificado Rama1
.classpath	Primer compromiso
.gitignore	Primer compromiso
.proyecto	Primer compromiso

UNIR RAMAS (MERGE)

Si queremos unir la Rama Maestra con la Rama1:

1. En el repositorio Github, elegir la Rama1



2. Y pulsamos al botón Solicitudes de extracción:

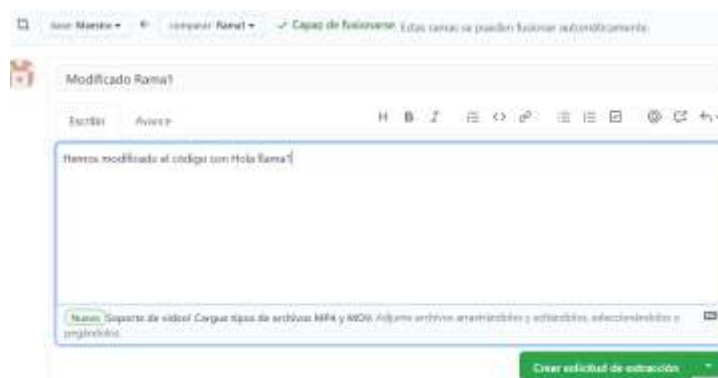


3. Clic Comparar y Nueva solicitar extracción:
Comparando cambios

Elige dos ramas para ver qué ha cambiado o para iniciar una nueva solicitud de extracción. Si es necesario, también puede comparar entre ramas.



4. Crear solicitud de extracción, y nos pide introducir un comentario, sobre lo que hace nuestra Rama1 y Crear solicitud de extracción.



5. Todavía no hemos realizado la unión (el Merge). Pinchamos en Solicitud de extracción(1):

The screenshot shows the GitHub interface for pull requests. The top navigation bar includes 'Código', 'Cuestiones', 'Solicitudes de extracción 1' (highlighted), and 'Comportamiento'. Below the navigation bar, there's a section titled 'Problemas de etiquetas y solicitudes de extracción' with the text 'Ahora, GitHub ayudará a los posibles colaboradores'. A search bar shows 'Filtros' and 'is:pr is:open'. Below the filters, there's a summary: '1 abierto' and '0 cerrado'. A list of pull requests follows, with the first one titled 'Modificado Rama1' by 'mgomezdeleon', marked as '# 1 abierto hace 8 minutos'.

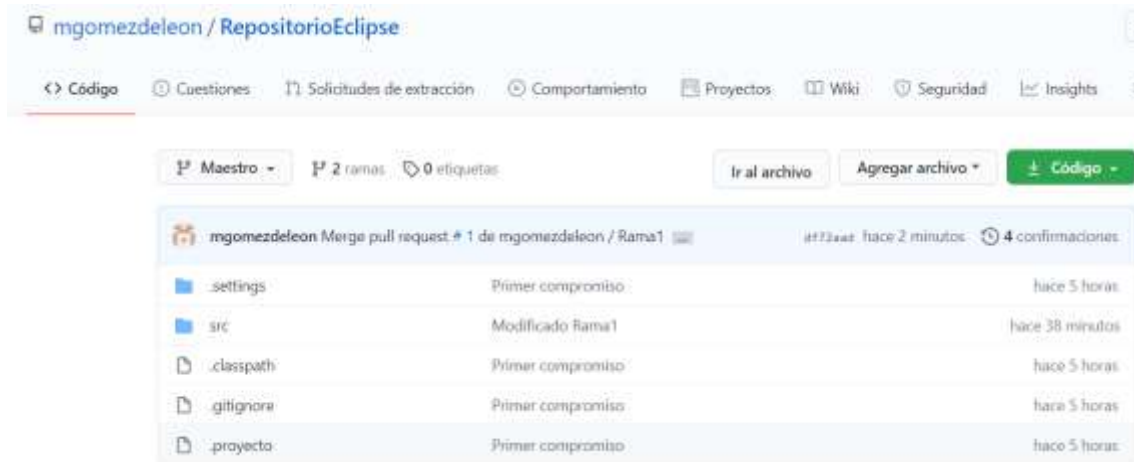
6. Seleccionamos Modificado Rama1 y combinar solicitar de extracción:

The screenshot shows the 'Compare' dialog in GitHub. It prompts the user to 'Agregar más confirmaciones presionando a la Rama1rama en mgomezdeleon / RepositorioEclipse'. The dialog contains two status messages: 'No se ha configurado la integración continua' (No continuous integration is configured) and 'Esta rama no tiene conflictos con la rama base.' (This branch has no conflicts with the base branch). At the bottom, there is a green button labeled 'Combinar solicitud de extracción' (Combine pull request) and a link to 'También puede abrir esto en GitHub Desktop o ver instrucciones de'.

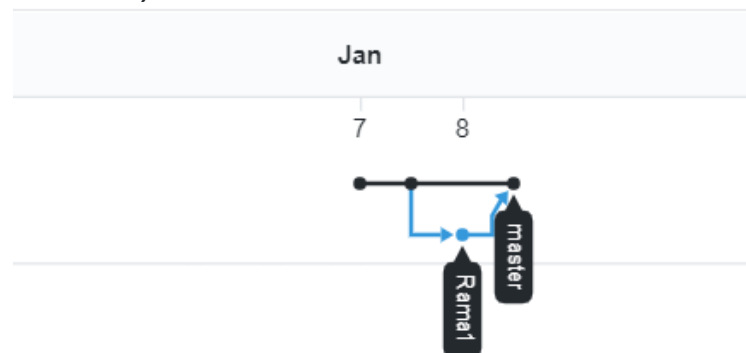
7. Merge Pull, Confirmar fusión

The screenshot shows the 'Merge pull request' confirmation dialog. It displays the title 'Merge pull request #1 from mgomezdeleon/Rama1' and the branch name 'Modificado Rama1'. At the bottom, there are two buttons: 'Confirmar fusión' (Confirm merge) and 'Cancelar' (Cancel).

8. Si nos vamos a la Rama Maestra, ya lo tenemos actualizado con la Rama1:

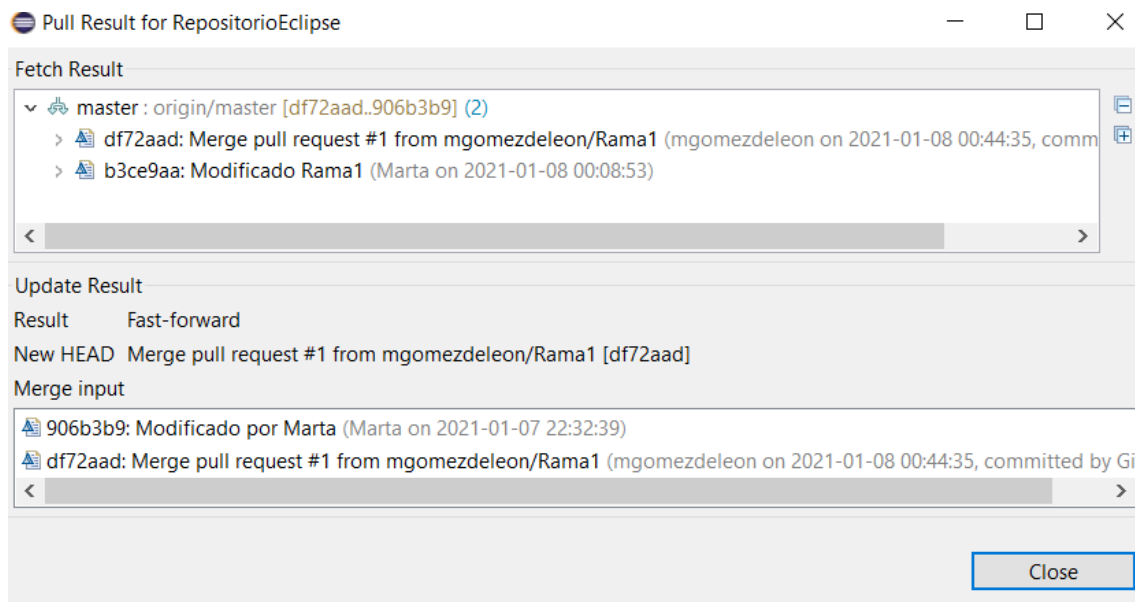


Nota: Si nos vamos a Insights (Estadística) y en el menú de la izquierda vamos a Red, nos muestra como de la rama maestra hemos creado una rama1, hemos modificado la rama1 y la hemos vuelto a unir con la maestra:



Si volvemos a **Eclipse**, comprobamos que nuestra rama1 no está unida con la maestra. Para eso tendremos que hacer:

1. Team > Switch to Master
2. Team > Pull



3. Al pinchar en Close, nos aparece ya el código:

