Actividad 4 - Utilizando sistemas de control de versiones

Brayan Steven Bonilla Castellanos

Juan Carlos Monsalve Gómez

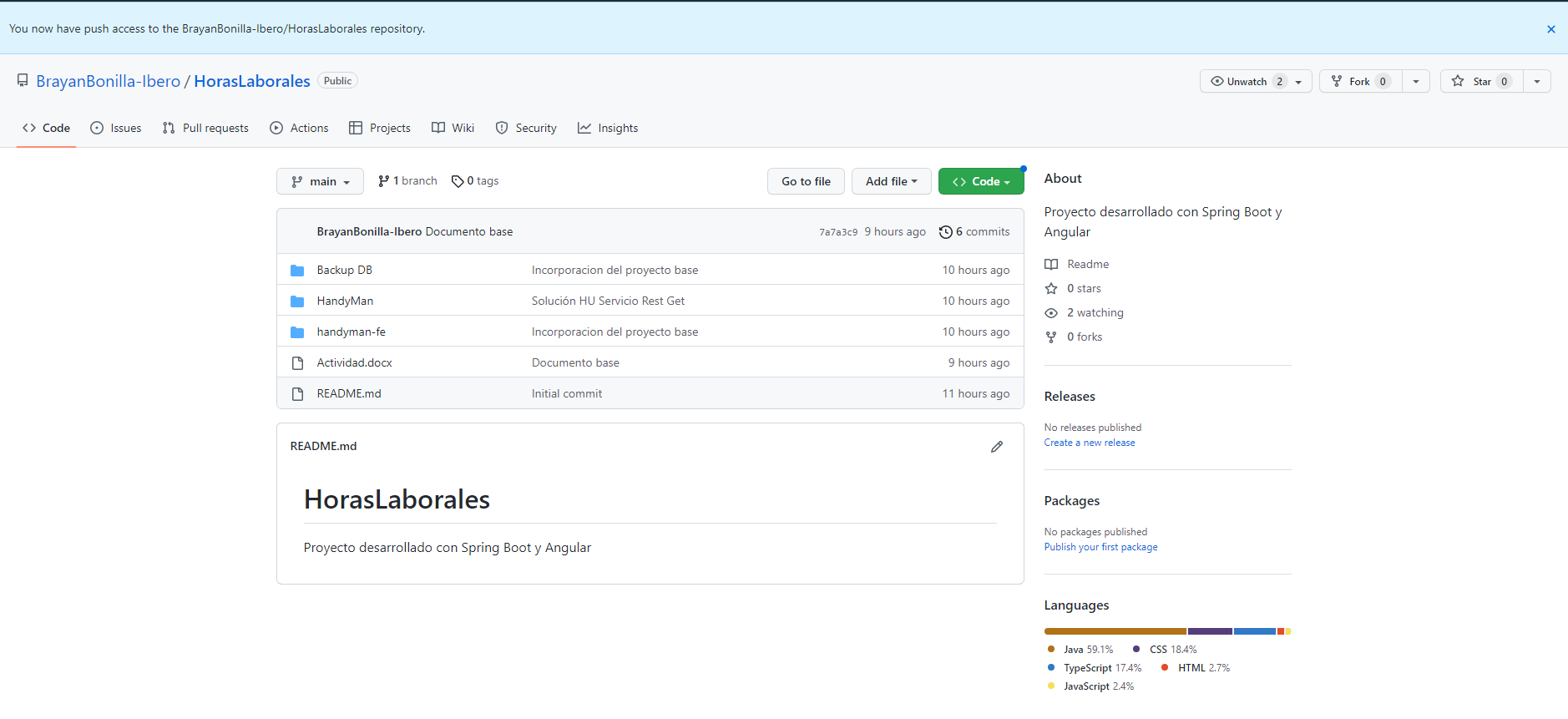
Corporación Universitaria Iberoamericana

Ingeniería de Software

Mantenimiento de Software

**Enlace Git**

[**https://github.com/BrayanBonilla-Ibero/HorasLaborales**](https://github.com/BrayanBonilla-Ibero/HorasLaborales)

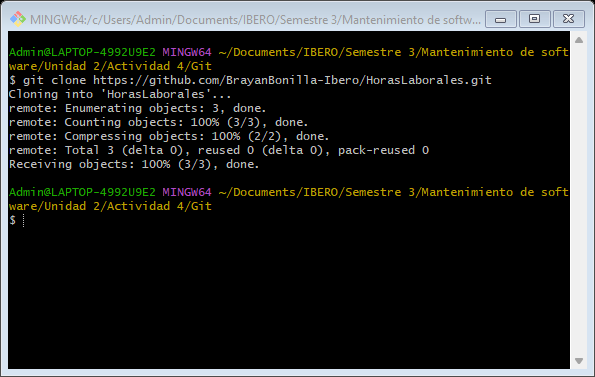


**Descripción del contexto**

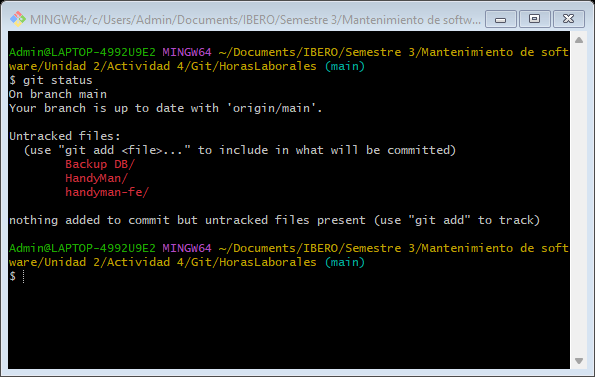
Se requiere implementar un sistema de información que permita el registro de horas laboradas por cada trabajador (Horas diurnas, horas nocturnas, horas dominicales, horas diurnas extras, horas nocturnas extras, horas dominicales extras).

**Flujo de trabajo**

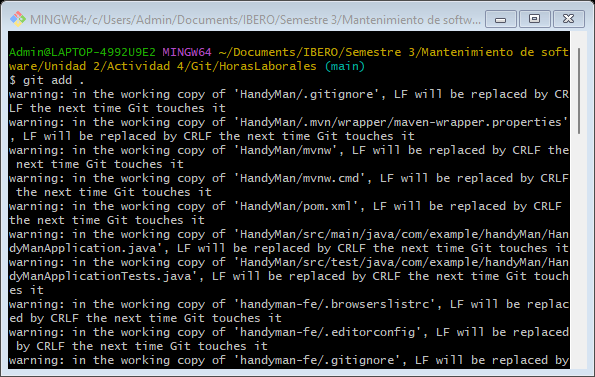
Ingresamos el comando **git clone**, para clonar el repositorio remoto al equipo local.



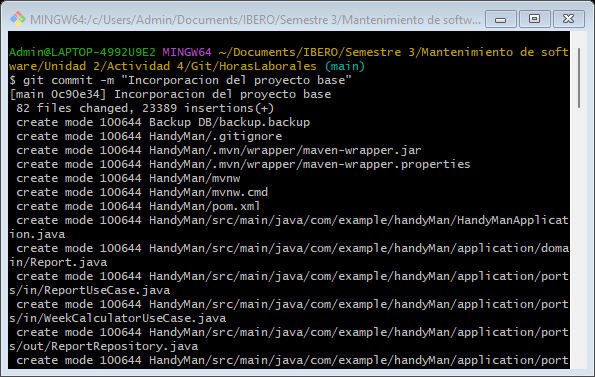
Una vez clonado el repositorio, incorporamos el proyecto base y verificamos los cambios realizados con el comando **git status**



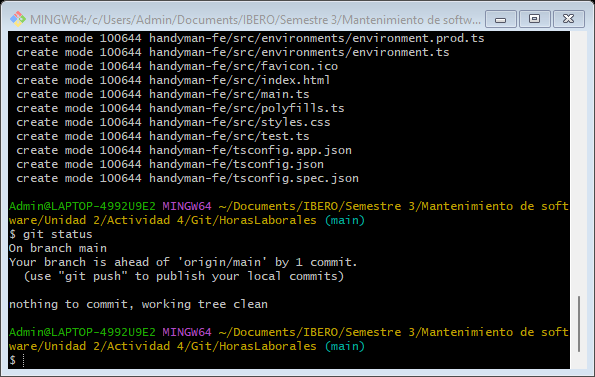
Procedemos a agregar los cambios realizados a un stage con el comando **git add .**



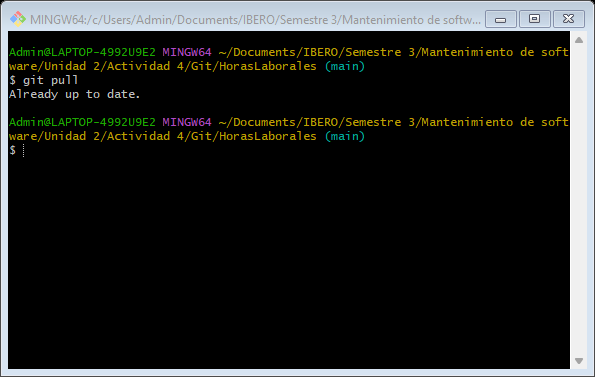
Una vez confirmado los cambios procedemos a realizar un commit, con el comando **git commit –m**



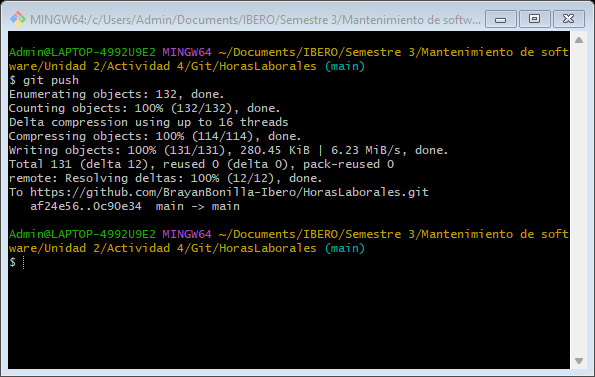
Verificamos que el commit se registrara con éxito, con el comando git status y debe registrar 1 commit pendiente por subir



Verificamos que no tengamos cambios pendientes por obtener con el comando **git pull**



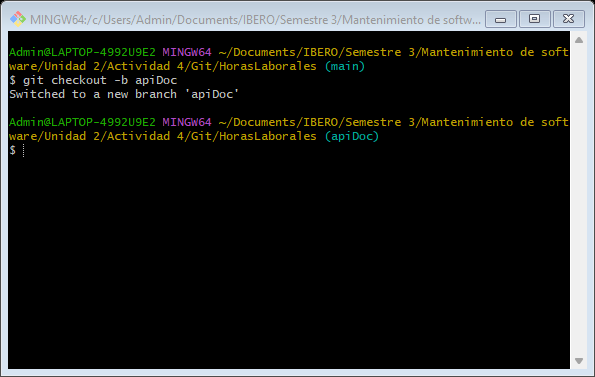
Procedemos a subir los cambios al repositorio remoto en la rama principal, con el comando **git push**



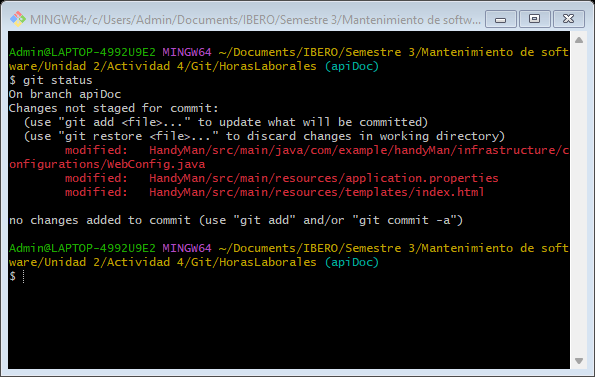
**Historias de usuario**

* **Se requiere la construcción de un api que contenga la documentación de los servicios desarrollados.**

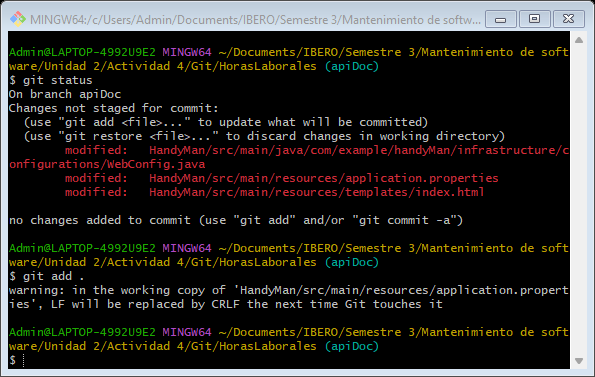
Para llevar a cabo esta tarea vamos a crear una rama específica para poder trabajar esta HU, con el comando **git checkout -b apiDoc**



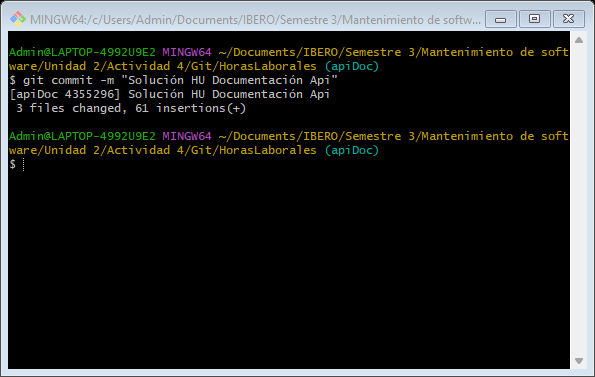
Una vez se trabaja la HU en el código fuente, procedemos a verificar los cambios, con el comando **git status**



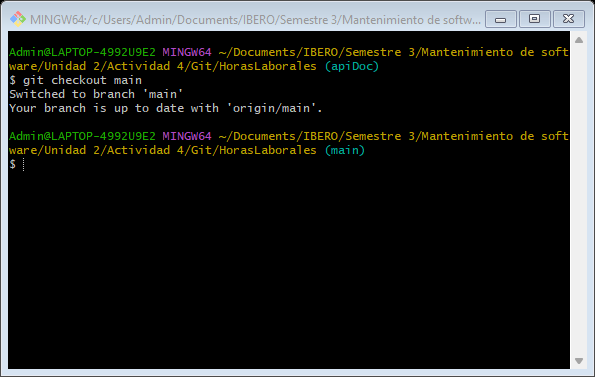
Agregamos los cambios al stage, con el comando **git add .**



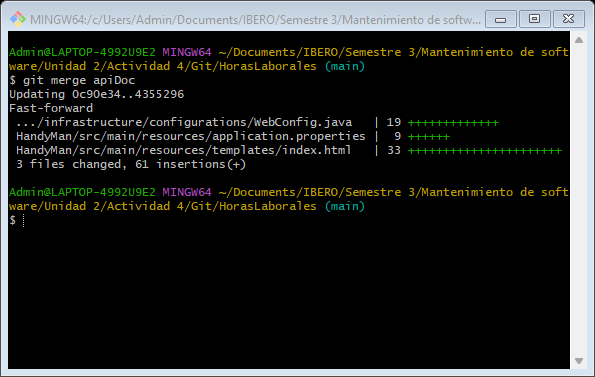
Realizamos el commit con los cambios realizados con el siguiente comando



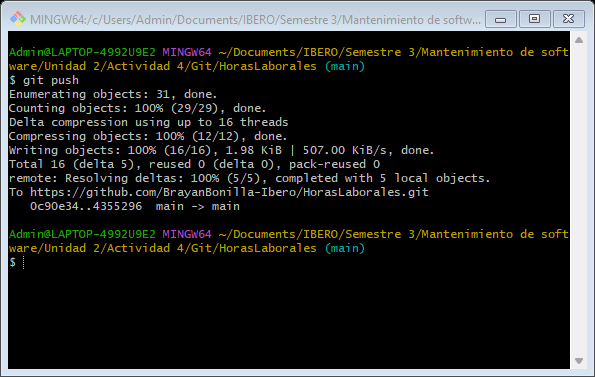
Una vez realizado el commit, procedemos a pasar a la rama principal con el siguiente comando



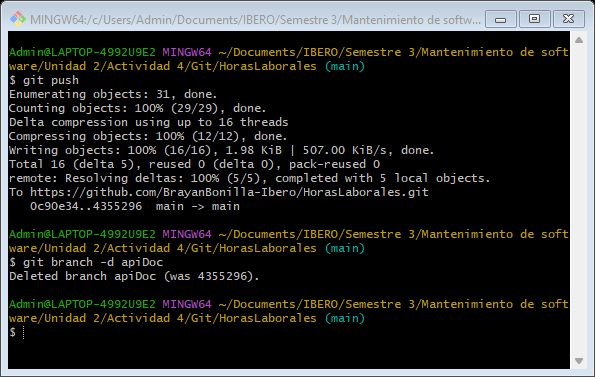
Una vez verificada la solución a la historia de usuario, procedemos a realizar el merge con la rama principal, con el siguiente comando



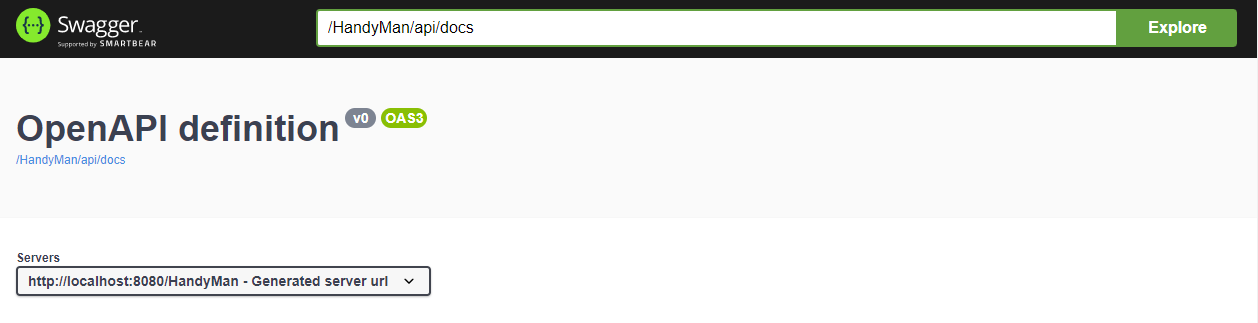
Subimos los cambios incorporados a la rama principal al repositorio remoto, con el siguiente comando



Como ya nos vamos a utilizar más la rama creada para trabajar la HU, procedemos a borrarla con el siguiente comando

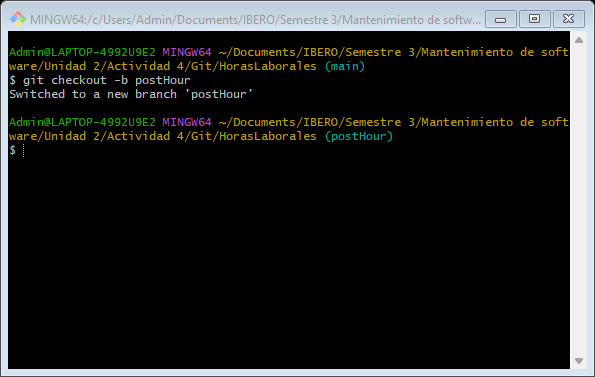


El resultado obtenido es el siguiente

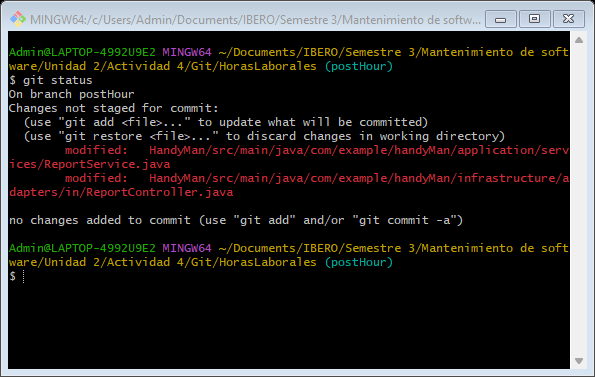


* **Se requiere crear un servicio Rest de tipo Post que permita guardar el registro de las horas por trabajador.**

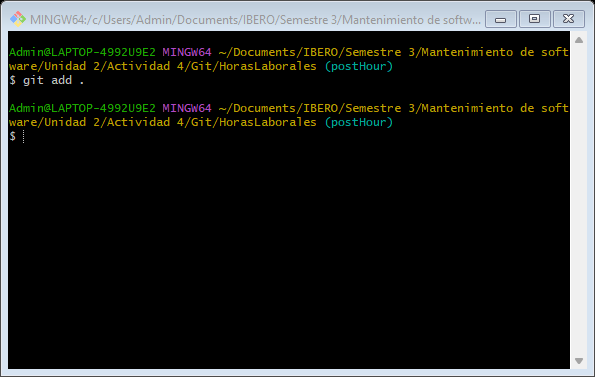
Para llevar a cabo esta tarea vamos a crear una rama específica para poder trabajar esta HU, con el comando **git checkout -b postHour**



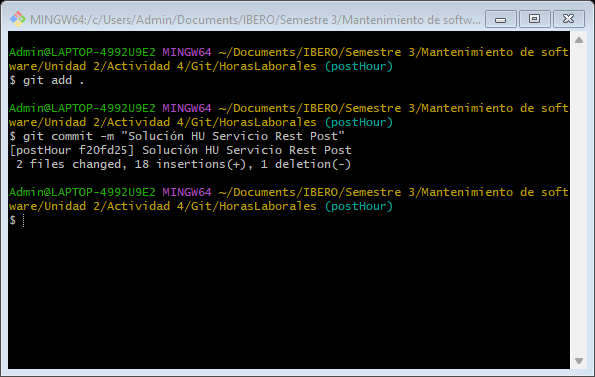
Una vez trabaja la HU en el código fuente, procedemos a verificar los cambios, con el comando git status



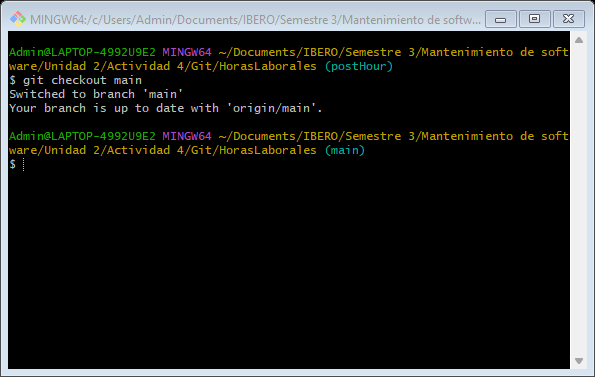
Agregamos los cambios al stage, con el comando **git add .**



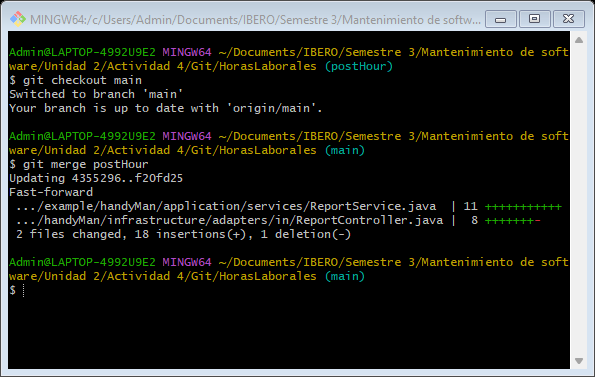
Realizamos el commit con los cambios realizados con el siguiente comando



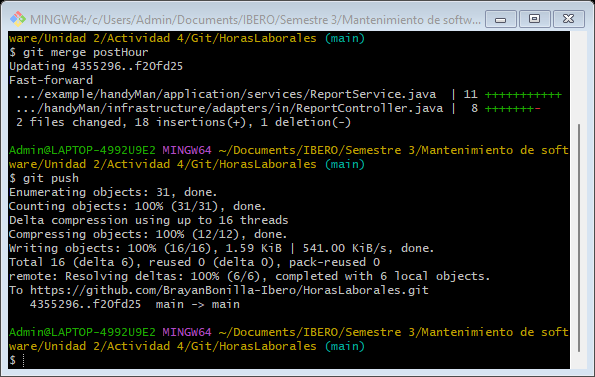
Una vez realizado el commit, procedemos a pasar a la rama principal con el siguiente comando



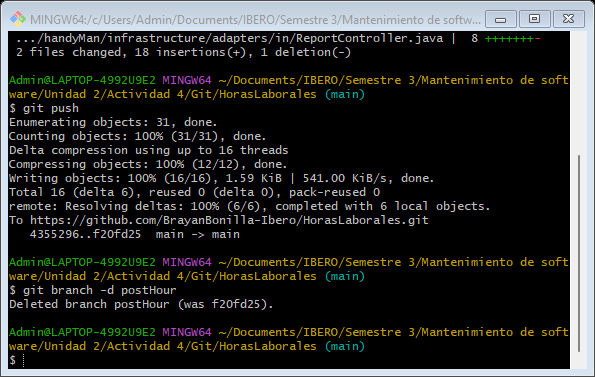
Una vez verificada la solución a la historia de usuario, procedemos a realizar el merge con la rama principal, con el siguiente comando



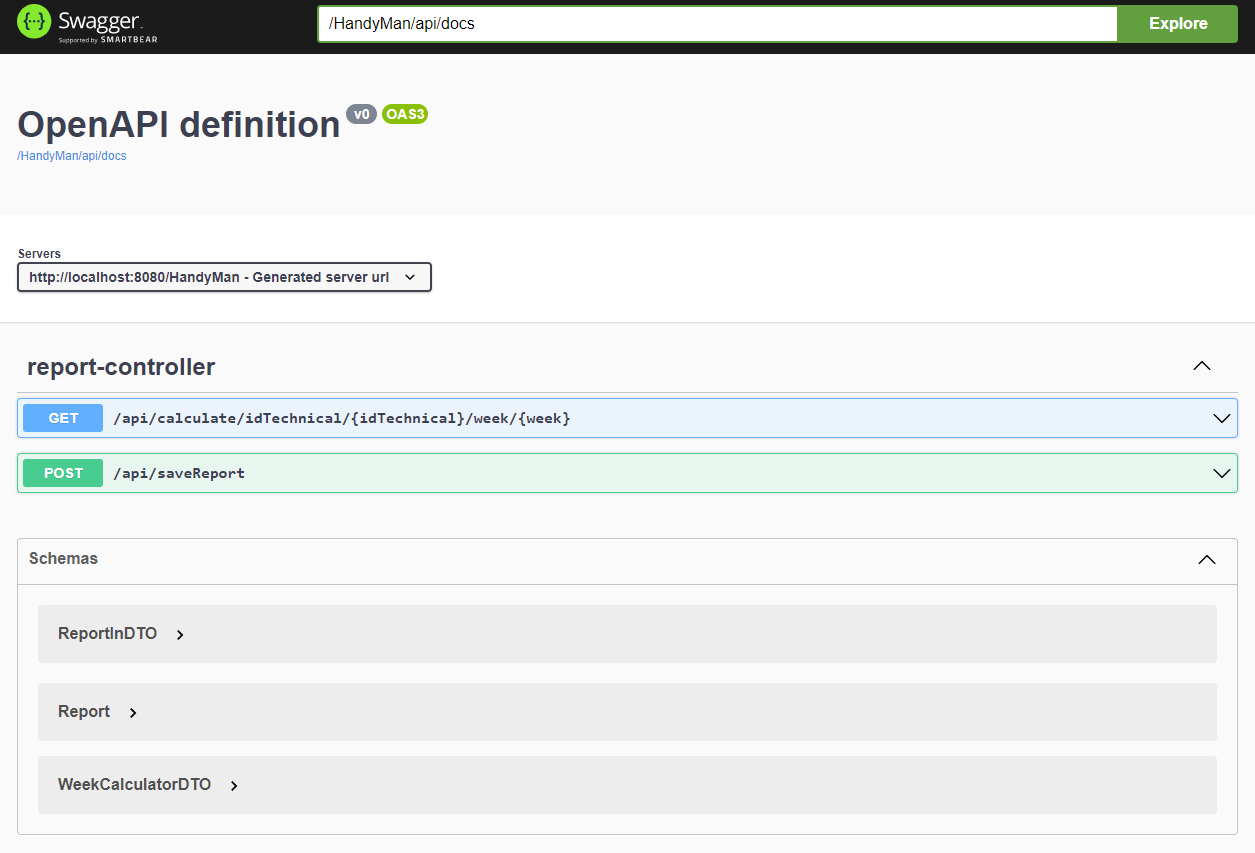
Subimos los cambios incorporados a la rama principal al repositorio remoto, con el siguiente comando



Como ya nos vamos a utilizar más la rama creada para trabajar la HU, procedemos a borrarla con el siguiente comando

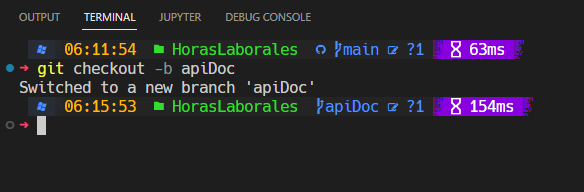


El resultado obtenido es el siguiente

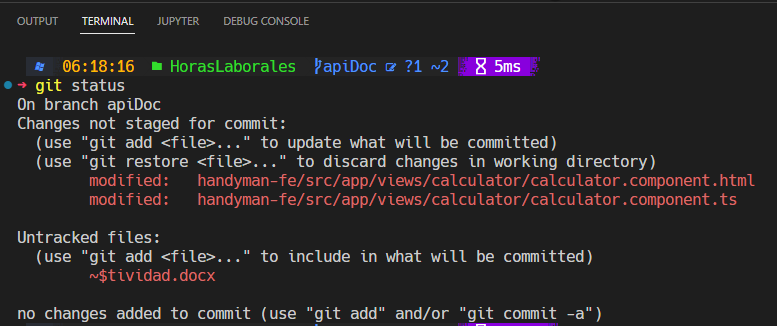


* **Se requiere diseñar la vista al usuario que le permita registrar sus horas laboradas.**

Para llevar a cabo esta tarea vamos a crear una rama específica para poder trabajar esta hu, con el comando **git checkout -b apiDoc**



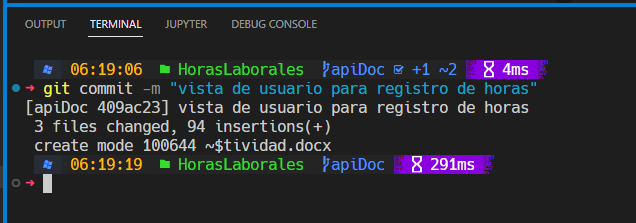
Una vez trabaja la HU en el código fuente, procedemos a verificar los cambios, con el comando **git status**



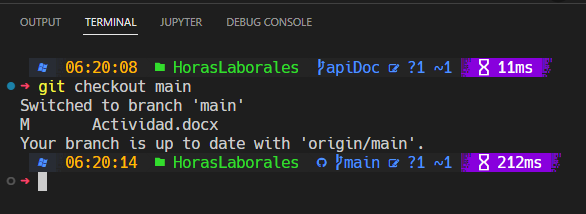
Agregamos los cambios al stage, con el comando **git add .**



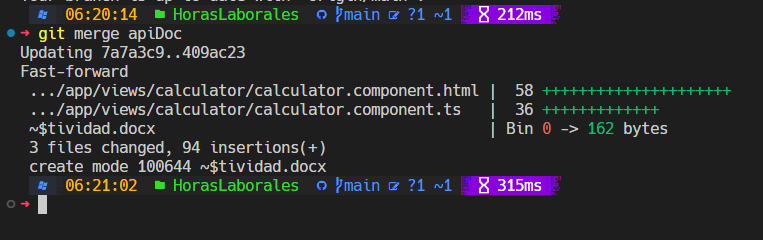
Realizamos el commit con los cambios realizados con el siguiente comando



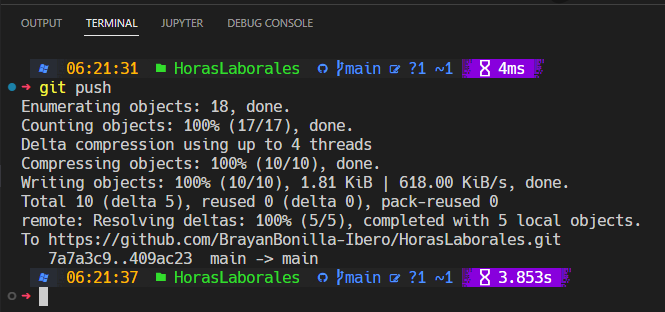
Una vez realizado el commit, procedemos a pasar a la rama principal con el siguiente comando



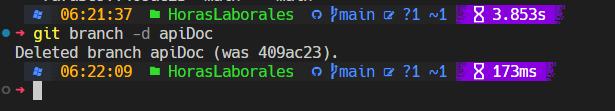
Una vez verificada la solución a la historia de usuario, procedemos a realizar el merge con la rama principal, con el siguiente comando



Subimos los cambios incorporados a la rama principal al repositorio remoto, con el siguiente comando



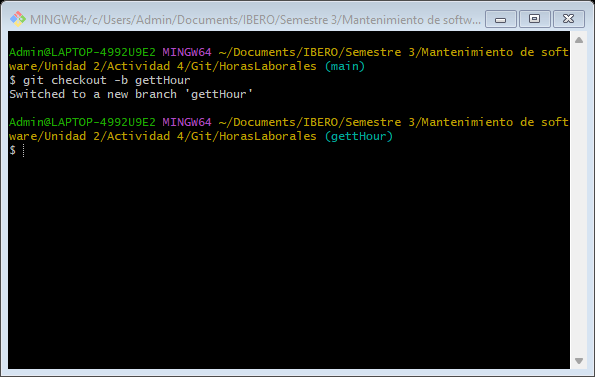
Como ya nos vamos a utilizar más la rama creada para trabajar la HU, procedemos a borrarla con el siguiente comando



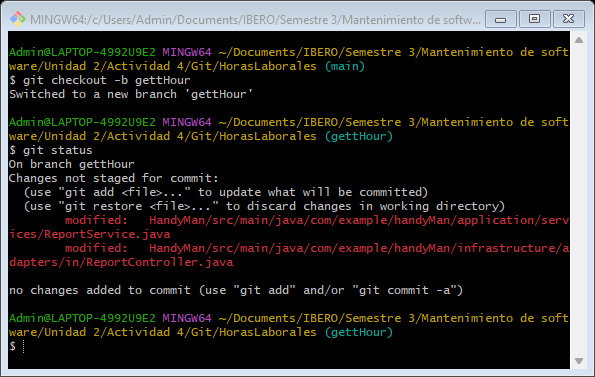
El resultado obtenido es el siguiente

* **Se requiere crear un servicio Rest de tipo Get que permita consultar las horas laboradas de un trabajador en una semana determinada.**

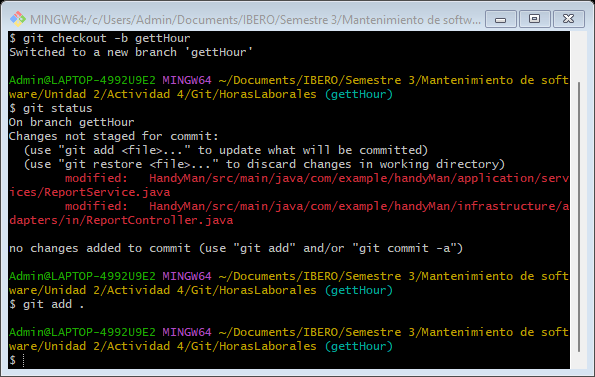
Para llevar a cabo esta tarea vamos a crear una rama específica para poder trabajar esta hu, con el comando git checkout -b getHour



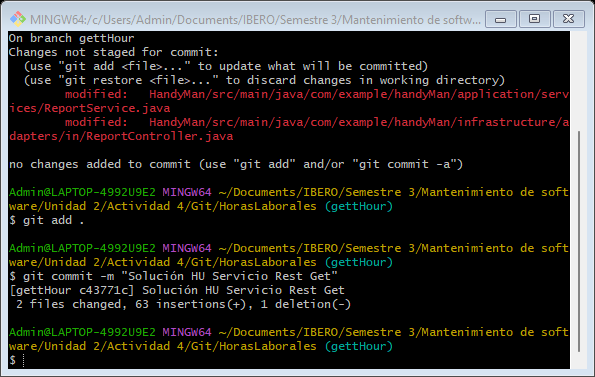
Una vez trabaja la HU en el código fuente, procedemos a verificar los cambios, con el comando git status



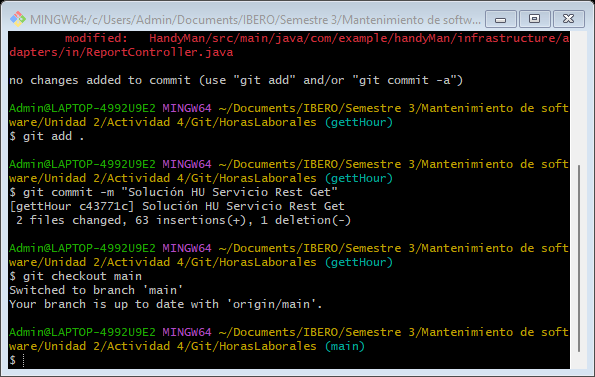
Agregamos los cambios al stage, con el comando git add .



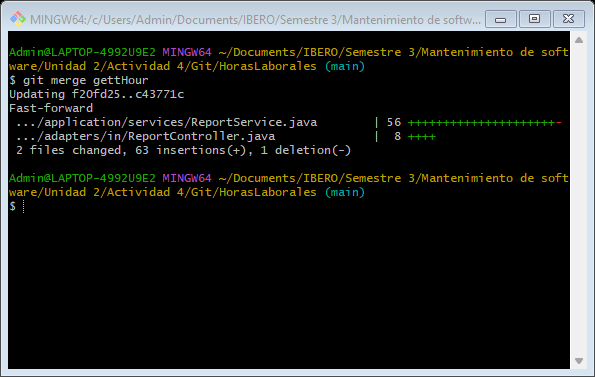
Realizamos el commit con los cambios realizados con el siguiente comando



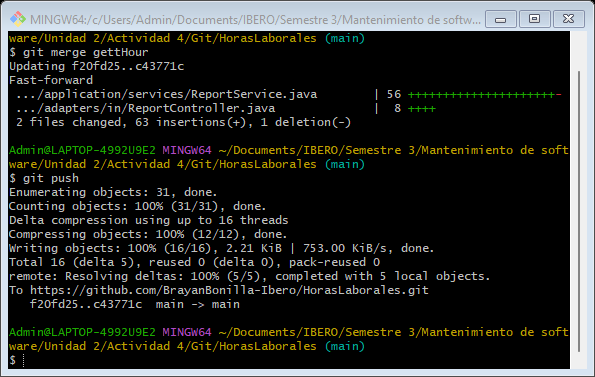
Una vez realizado el commit, procedemos a pasar a la rama principal con el siguiente comando



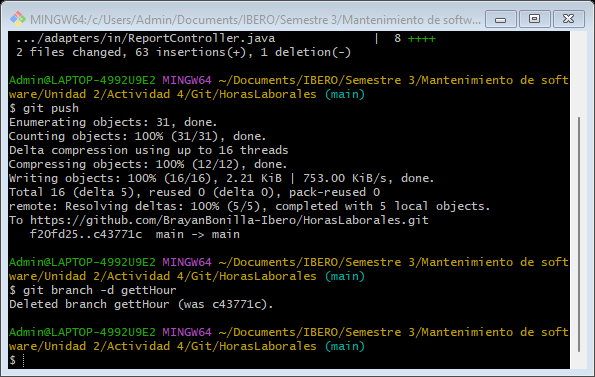
Una vez verificada la solución a la historia de usuario, procedemos a realizar el merge con la rama principal, con el siguiente comando



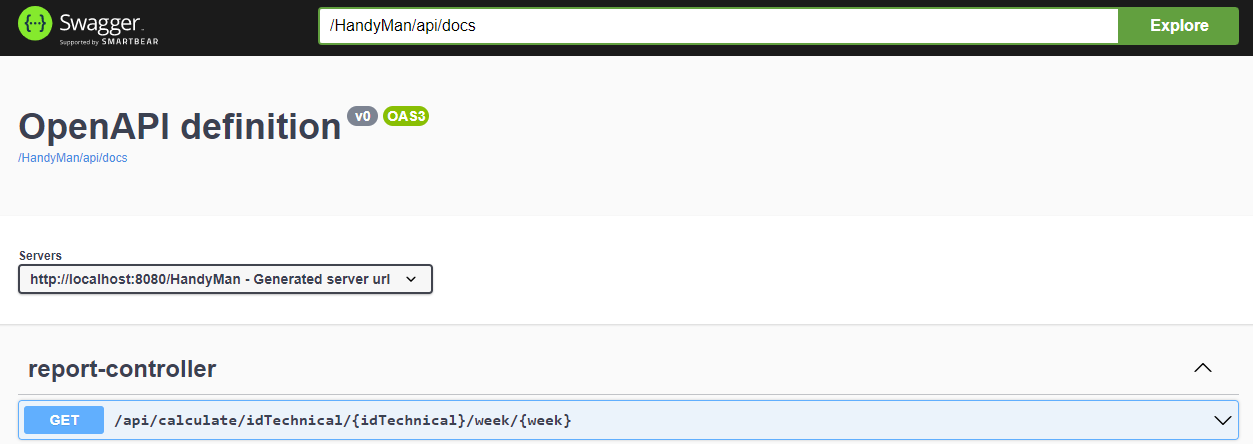
Subimos los cambios incorporados a la rama principal al repositorio remoto, con el siguiente comando



Como ya nos vamos a utilizar más la rama creada para trabajar la HU, procedemos a borrarla con el siguiente comando



El resultado obtenido es el siguiente



* **Se requiere diseñar la vista al usuario que le permita consultar las horas laboradas en una semana determinada.**

Para llevar a cabo esta tarea vamos a crear una rama específica para poder trabajar esta hu, con el comando git checkout -b apiDoc

Una vez trabaja la HU en el código fuente, procedemos a verificar los cambios, con el comando git status

Agregamos los cambios al stage, con el comando git add .

Realizamos el commit con los cambios realizados con el siguiente comando

Una vez realizado el commit, procedemos a pasar a la rama principal con el siguiente comando

Una vez verificada la solución a la historia de usuario, procedemos a realizar el merge con la rama principal, con el siguiente comando

Subimos los cambios incorporados a la rama principal al repositorio remoto, con el siguiente comando

Como ya nos vamos a utilizar más la rama creada para trabajar la HU, procedemos a borrarla con el siguiente comando

El resultado obtenido es el siguiente

