

Universidad De San Carlos De Guatemala
Facultad De Ingeniería
Escuela de Ciencias y Sistemas
Análisis y Diseño de Sistemas 2
Laboratorio

Hoja de Trabajo 1

Control de versiones

Luis Aroldo Morales Noriega
201020944
10/08/17

Control de versiones

• Parte I

1. ¿Qué es un sistema de control de versiones?

Es donde se gestionan cambios que se realizan sobre elementos de alguna configuración y se almacena el historial de esos cambios.

2. ¿Qué es un repositorio?

Es el lugar donde se van guardando los datos que se van actualizando los cambios.

3. En el contexto de sistemas de control de versiones, definir con sus propias palabras los siguientes conceptos:

- **Árbol:** Es el conjunto de directorios o archivos dentro del repositorio de un proyecto.
- **Revisión:** Es una versión de la información que va cambiando y actualizándose.
- **Reléase:** Es cuando el usuario termina una determinada versión.
- **Rama:** Es un módulo que se puede modificar en cierto tiempo.
- **Etiqueta:** Es la versión que se le da a cada fichero de los módulos para que cuando se busquen se puedan encontrar con ese nombre.
- **Versión:** Es un cambio terminado de la información que se está gestionando.
- **Commit:** Es cuando se realiza una copia de los cambios en otra copia local del repositorio.
- **Update:** La actualización que integra los cambios hechos en el repositorio en la copia de trabajo local.

- **Check out:** Es el que crea una copia de trabajo local desde el repositorio, puede ser una versión en específico o la última versión.
- **Merge:** Es una integración que une dos cambios sobre un fichero o ficheros en una revisión de ese fichero o ficheros.

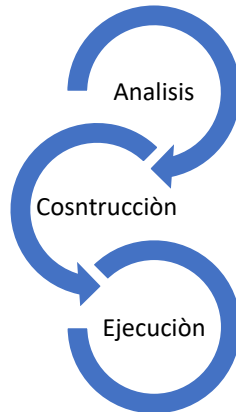
4. Diferencia entre trunk y branch

Branch es donde en un instante de tiempo se tiene dos copias que van cambiando independientes en su línea de desarrollo y trunk es donde se unifican las branch.

5. ¿Qué es y porque es necesaria la integración continúa?

A la hora de tener un proyecto en el que son 2 o más desarrolladores se puede tener un mejor control de los cambios, y así también unificar las partes de cada miembro en tiempo real.

6. Hacer un diagrama donde se explique el proceso de integración continúa

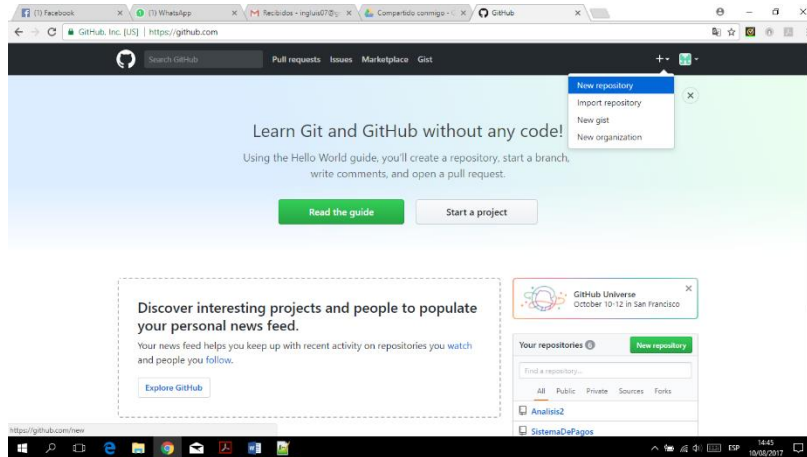


• Parte II

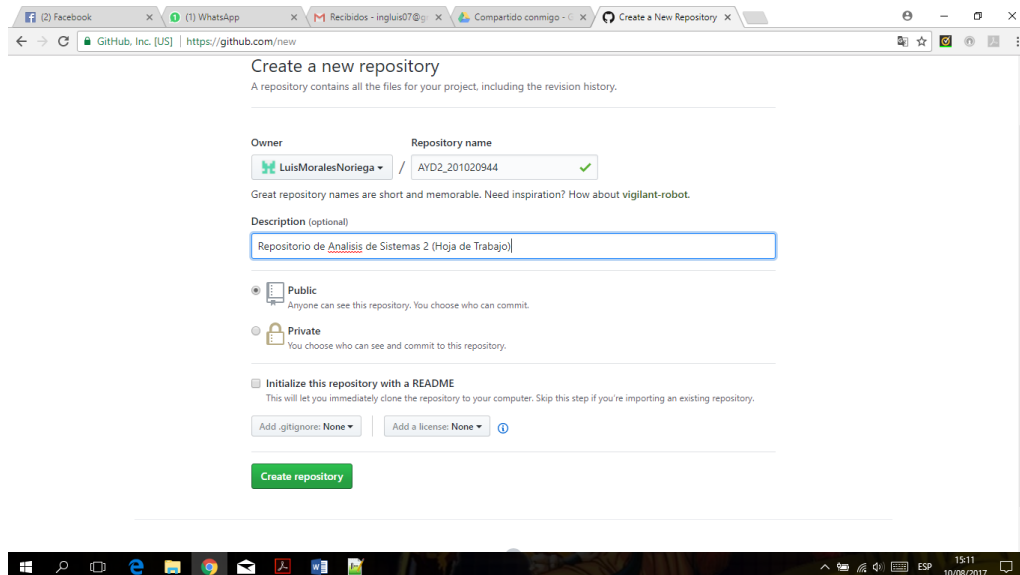
1. Crear un repositorio en GitHub llamado AYD2_#Carnet.

Para realizar esto seguimos los siguientes pasos:

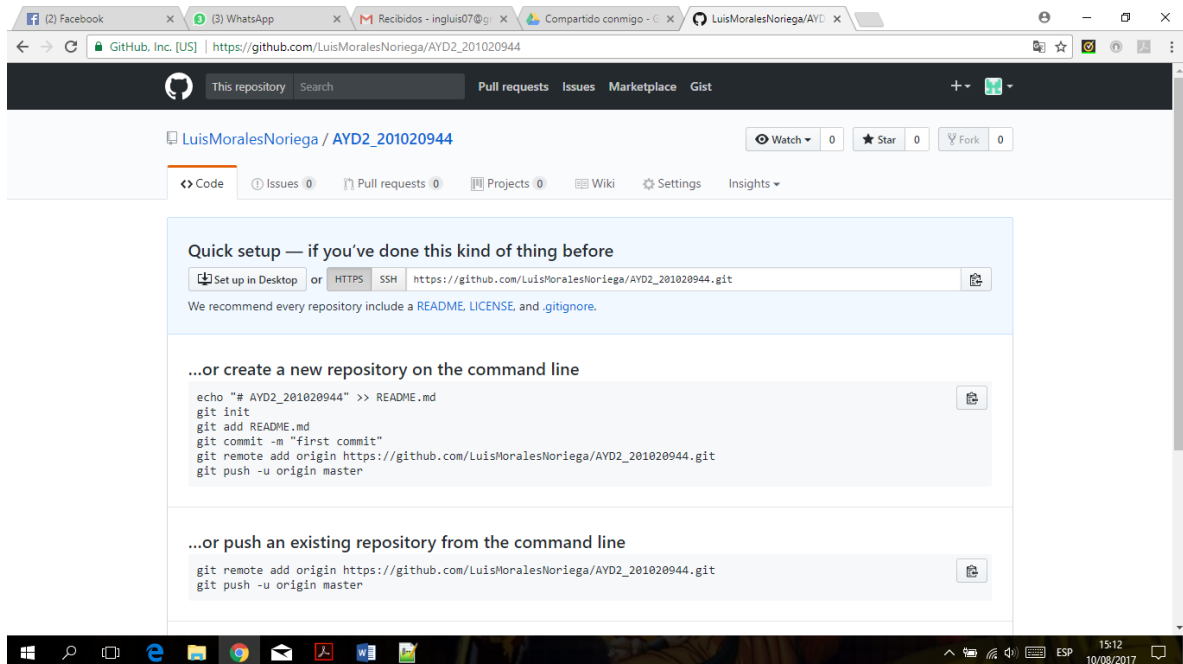
- a. Ingresar a la cuenta de GitHub e ir a la pestaña "+" luego "New Repository" como se ve en la imagen.



- b. Luego podemos llenar los datos que solicita, dándole el nombre de AYD2_201020944 y seleccionamos "Create repository".



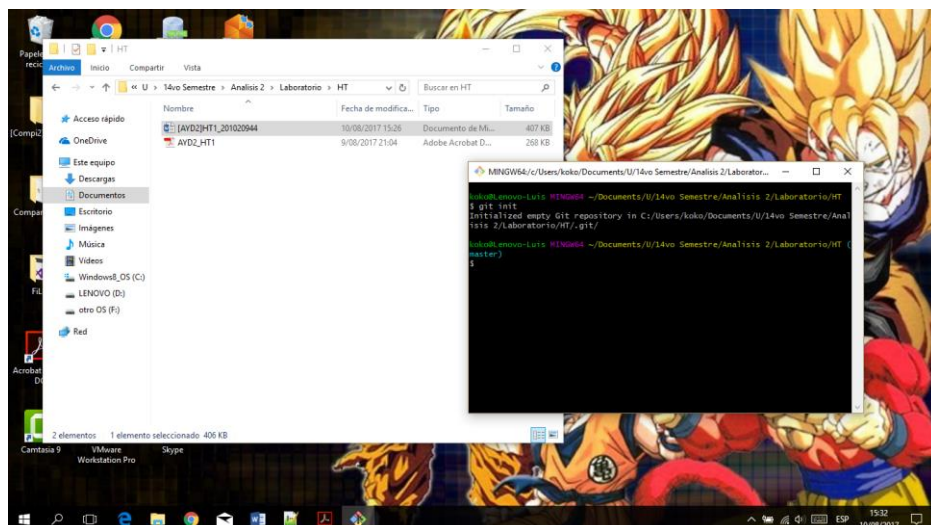
c. Nos muestra el repositorio creado



2. Crear un branch llamado HT1_Branch1 e incluir el archivo de la hoja de trabajo sin las respuestas

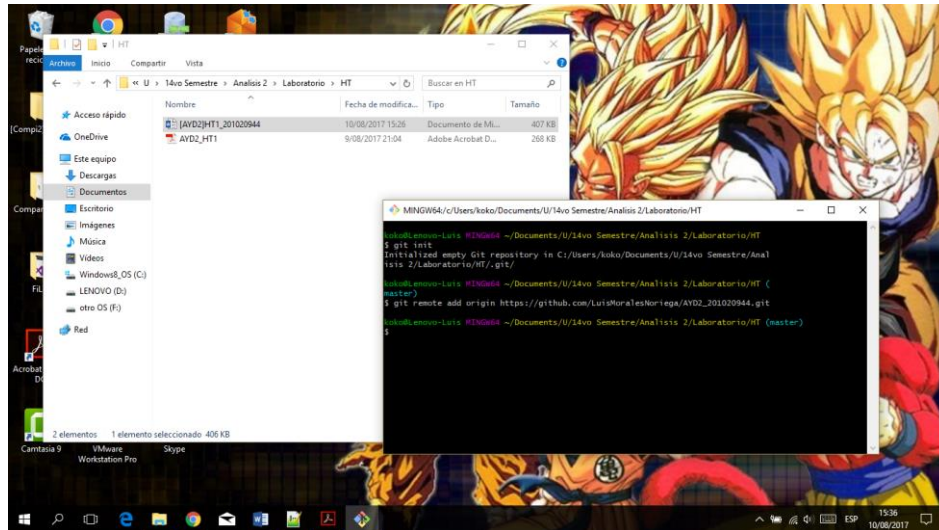
a. Iniciamos el repositorio en la carpeta master del proyecto con el comando

“git init”



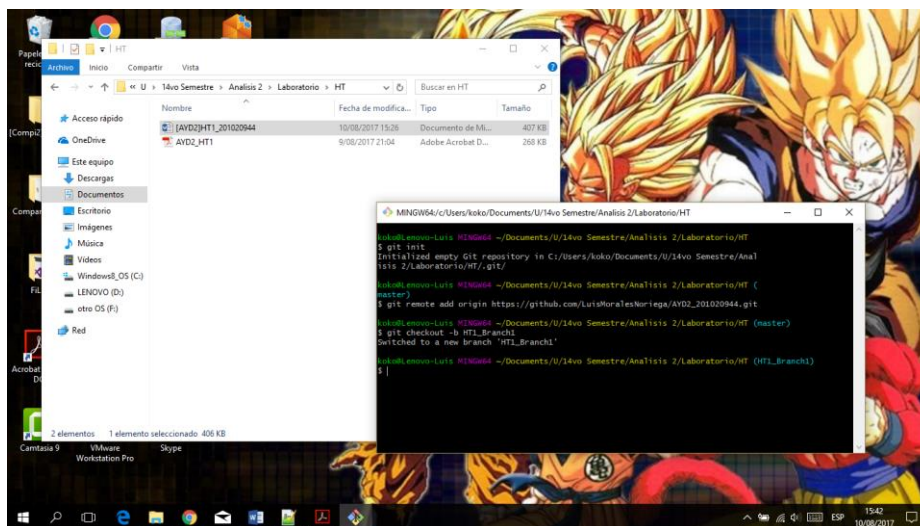
- b. Ahora agregamos el origen master de nuestro proyecto que fue el que se creó en GitHub con el siguiente comando:

Git remote add origin
https://github.com/LuisMoralesNoriega/AYD2_201020944.git



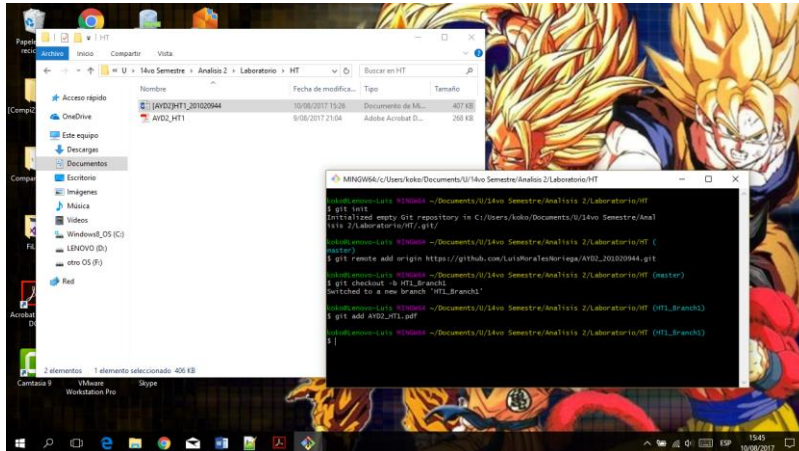
- c. Ahora creamos una rama con el nombre "HT1_Branch1" con el siguiente comando.

git checkout -b HT1_Branch1



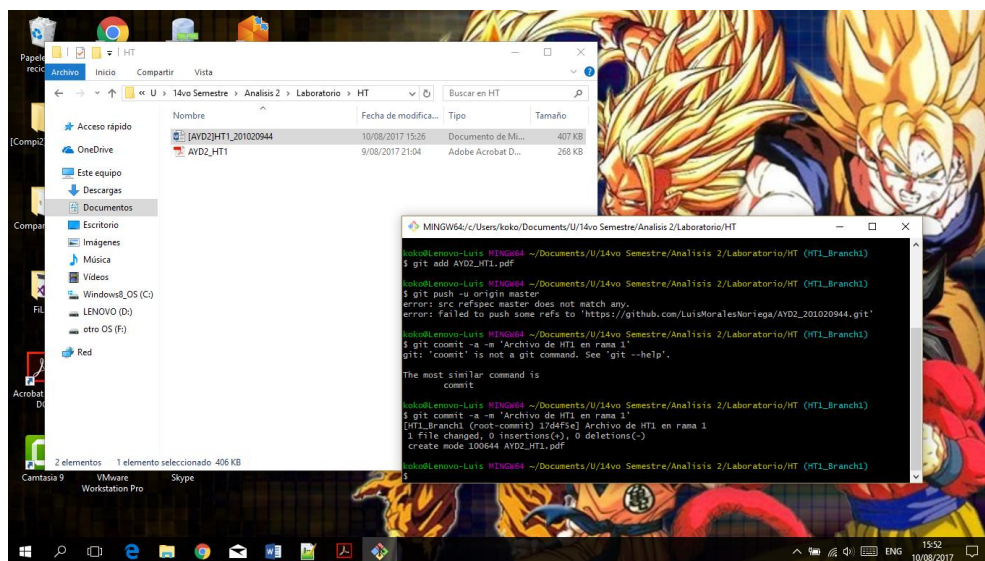
- d. Ahora agregamos el archivo de la hoja de trabajo, con el comando siguiente:

git add AYD2_HT1.pdf



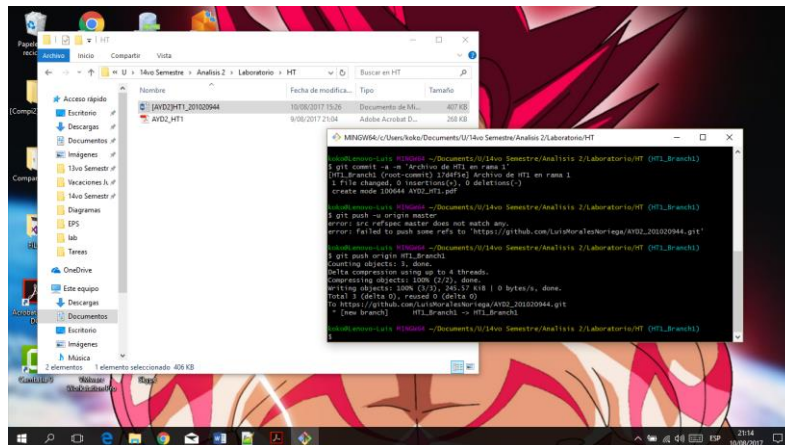
- e. Ahora, hacemos un commit para tener listo lo que se actualizara en el repositorio con el comando siguiente:

git commit -a -m 'Archivo de HT1 rama1'

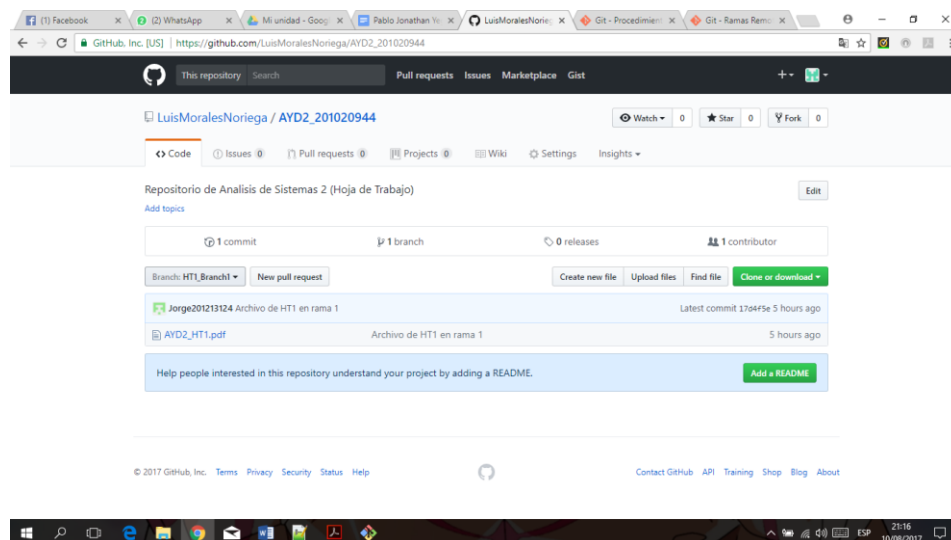


f. Por último, realizamos un push para ver los cambios en el repositorio.

git push origin HT1_Branch1



g. Ahora podemos ver en nuestro repositorio la rama creada.



3. Crear otro branch llamado HT1_Branch2 incluir el archivo de la hoja de trabajo con todas las respuestas.
 - a. Para realizar esta rama se utilizarán los pasos del punto 2 y la rama se muestra en nuestro repositorio con el archivo de respuestas.

