Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Introducción a la Programación y Computación II

Laboratorio

Escuela de Vacaciones, Junio 2015

Auxiliar: Reimer Chamalé

GIT

Paula María Vásquez Cifuentes

7 de Junio, 2015

# CONTROL DE VERSIONES

Es una herramienta que se utilizan para llevar un control del software. Una de sus características es que permite la complejidad del software. Sin embargo no incluye consideraciones del entorno del desarrollo.

Repositorio: se le llama asi al lugar en donde se guarda el código. Es un sitio centralizado en donde llegan o se depositan los cambios realizados en el código. Este permite realizar un back-up, restablecerse en otro lugar. Una de las ventajas de trabajar en un repositorio es que se guardan los cambios, pero se pueden entrar a los cambios anteriores, lo cual permite obtener una mejor comparación entre ideas sin perder información.

Otros términos importantes en el manejo de versiones son: Servidor, cliente, copia local y trunk. Esta ultima palabra no es tan común como las otras y se refiere a la versión final del código. La palabra diff, se utiliza también para referirse a control de versiones.

Branch, es otro de los beneficios de utilizar esta herramienta de trabajo, esto quiere decir que duplica el código en folders separados permitiendo que cada uno lleve una ruta distinta de cambios, pero en ningun momento se pierde la información anterior. Un merg es el punto en que se realizó un branch. También se puede realizar durante el proceso de aporte al código se pueden realizar etiquetas para su mejor diferenciación.

# ¿QUÉ ES GIT?

GIT es un sistema de control de versiones que evita perder información o ciertas características de las versiones anteriores, que se pudieran necesitar después.

## ¿Cómo funciona GIT?

La principal diferencia entre GIT y cualquier otro VCS es cómo GIT modela sus datos. Otros VCS guardan los cambios sobre una versión base, tal como se muestra en la figura I. En cambio GIT guarda las versiones, no solo los cambios de cada archivo, como se muestra en la figura II.

Figura I: **Modelación de datos de un VCS común**



Fuente: https://git-scm.com/

Figura II: **Modelación de datos de GIT**



Fuente: https://git-scm.com/

La mayoría de las operaciones en Git sólo necesitan archivos y recursos locales para operar. Guarda el historial de versiones en el disco local, entonces se puede trabajar y realizar una nueva versión sin internet y luego guardarla en la web cuando haya conexión. Por esta razón parece más rápido que otros VCS.

Git tiene tres estados principales en los que se pueden encontrar tus archivos: confirmado (committed), modificado (modified), y preparado (staged). Confirmado significa que los datos están almacenados de manera segura en tu base de datos local. Modificado significa que has modificado el archivo pero todavía no lo has confirmado a tu base de datos. Preparado significa que has marcado un archivo modificado en su versión actual para que vaya en tu próxima confirmación. Para guardar estos tres estados se utilizan diferentes secciones. La primera es el directorio de GIT, es un repositorio que se guarda en la nube y es lo que se descarga al utilizarlo en diferentes máquinas. Estos archivos ya están en un estado confirmado. Luego está el directorio de trabajo, es el repositorio que se guarda en el ordenador, el cual sirve para realizar modificaciones. Son los archivos en estado modificado. Por último está el área de preparación, la cual guarda los cambios realizados en los archivos para poder guardarlo en el directorio de Git. Cuando el archivo está en esta sección está preparado o en staged,

# PROGRAMAS QUE IMPLEMENTAN GIT

## GITHUB

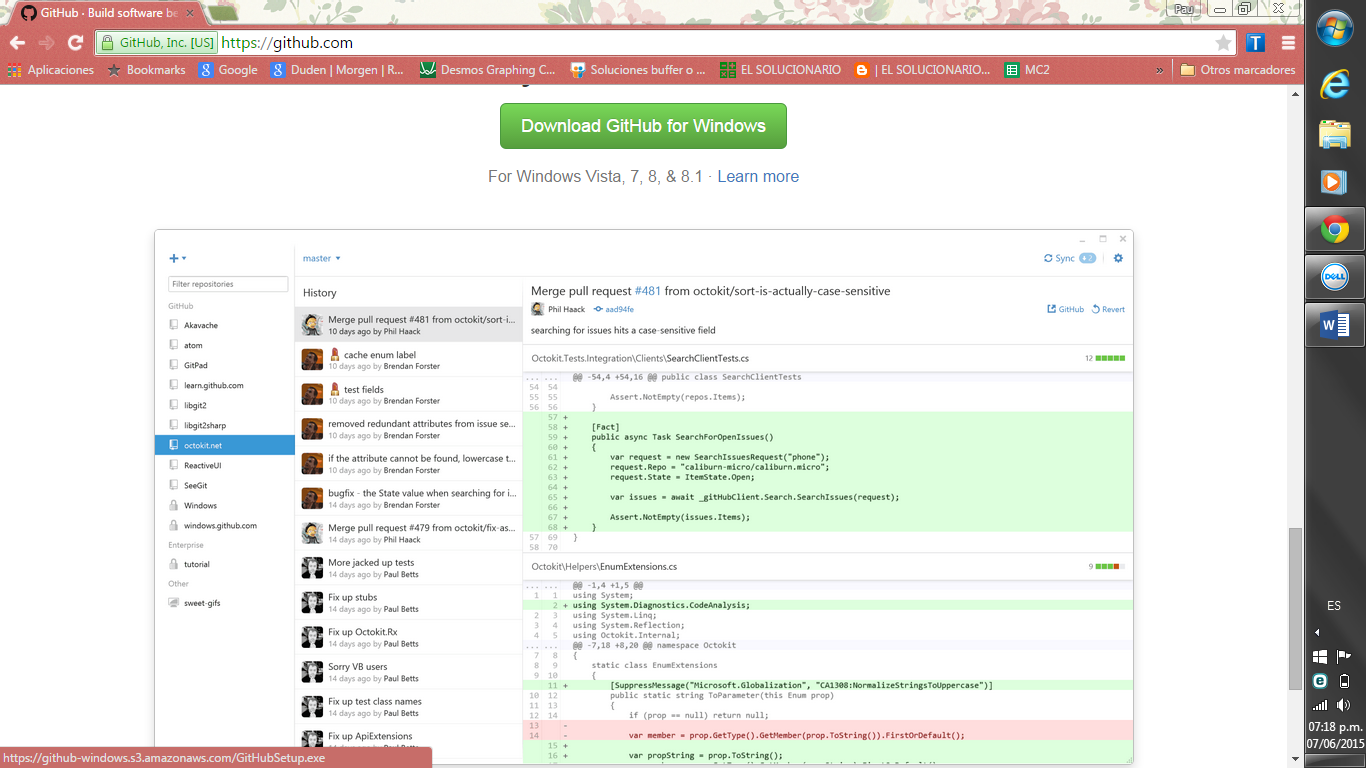
GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones GIT. GitHub presenta distintas herramientas para no solo trabajar de forma individual, si no colectiva tales como:

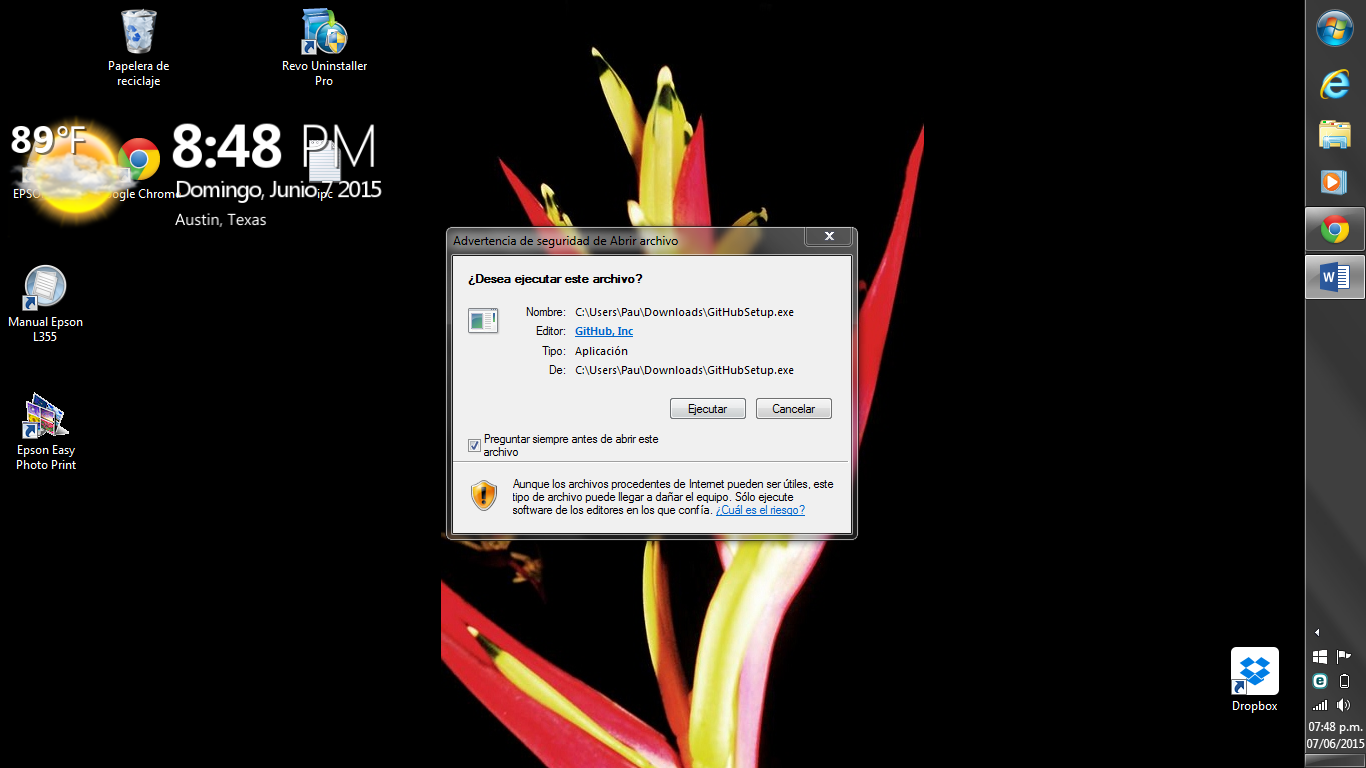
* Una wiki para el mantenimiento de las distintas versiones de las páginas.
* Un sistema de seguimiento de problemas que permiten a los miembros de tu equipo detallar un problema con tu software o una sugerencia que deseen hacer.
* Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero y debatir sobre determinados cambios realizados en un commit específico.
* Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

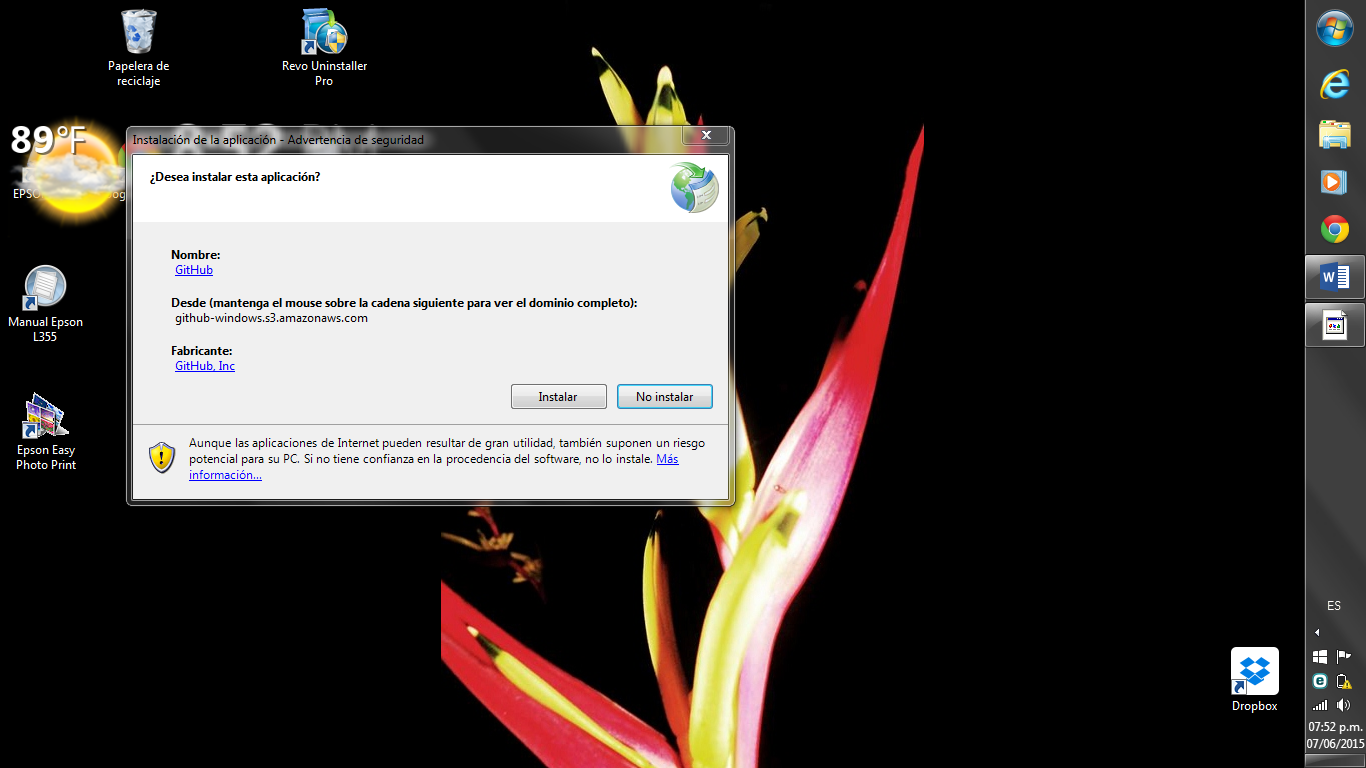
### Instalación de GitHub

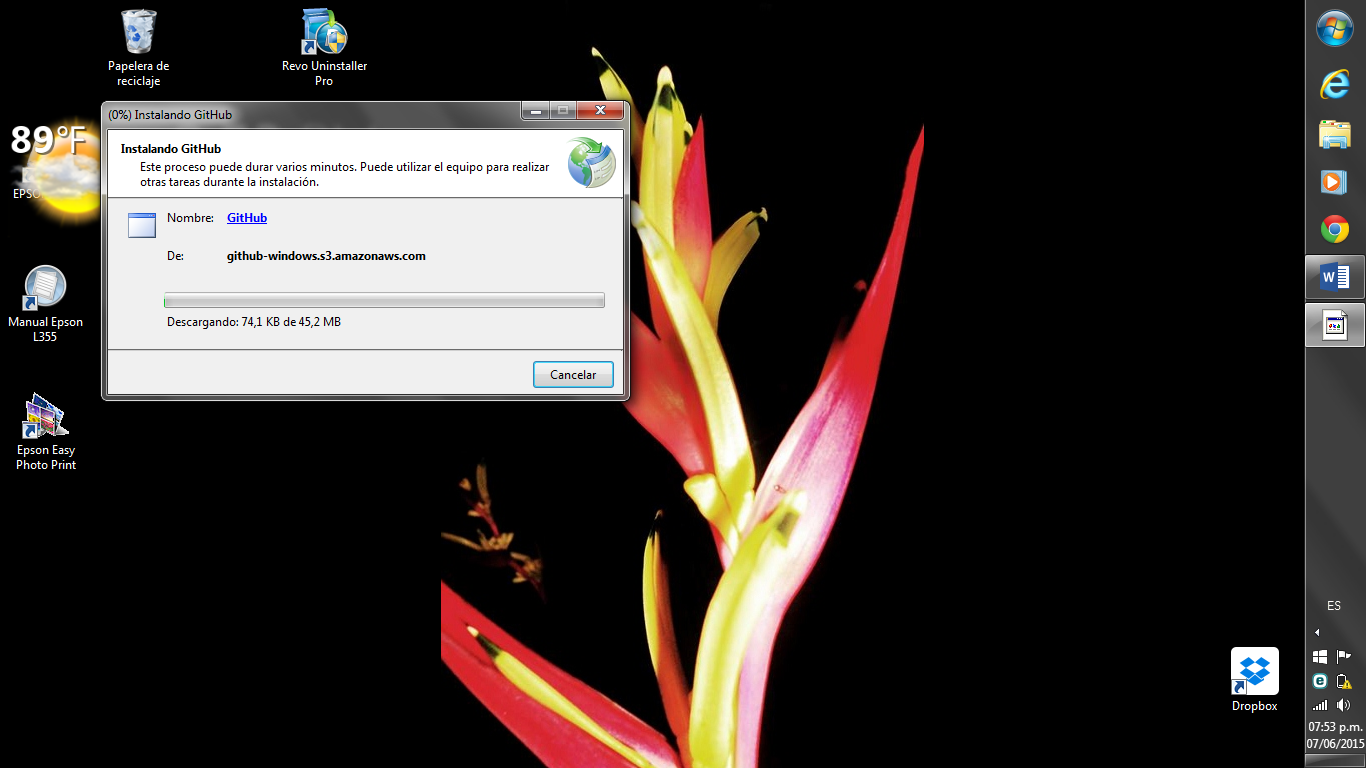
Los pasos para instalar GitHub en un ordenador son:

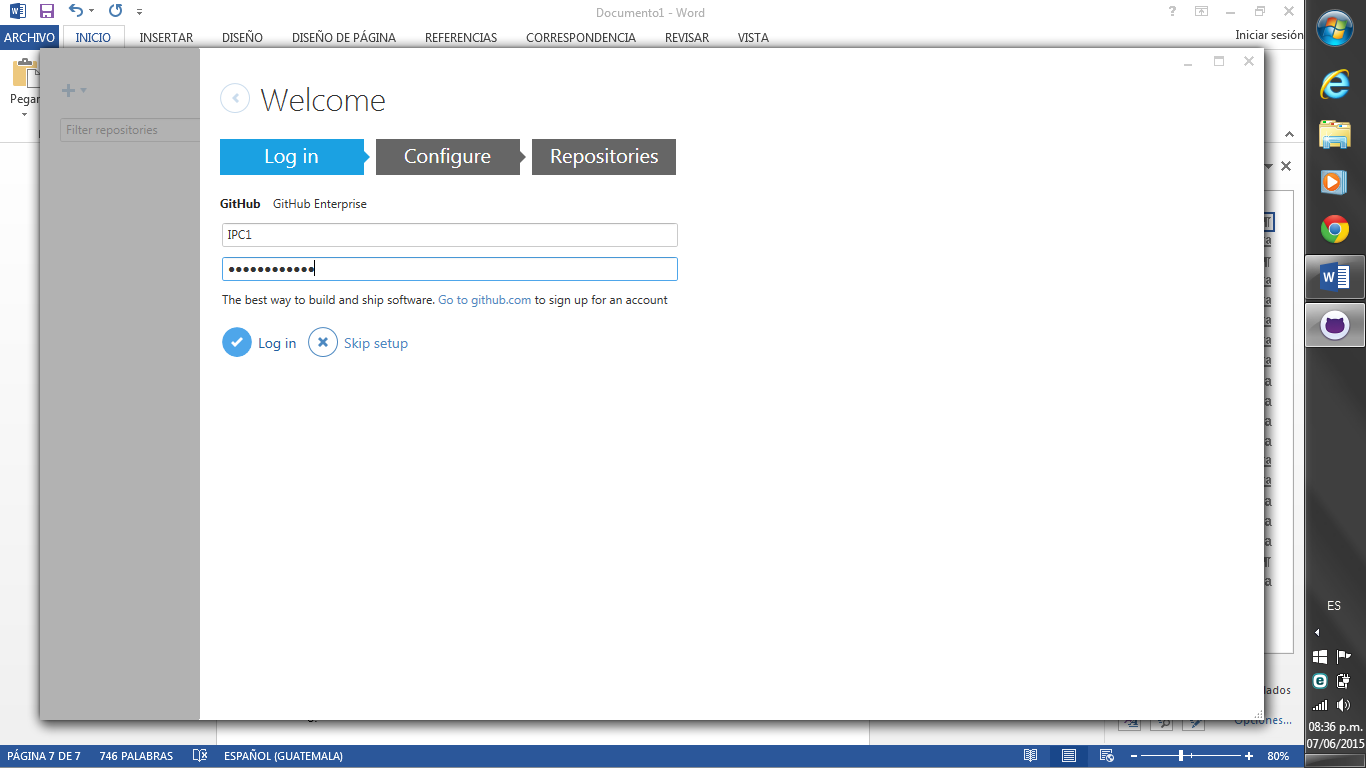
1. Entrara a github.com y crear una cuenta si aún no tienes una.

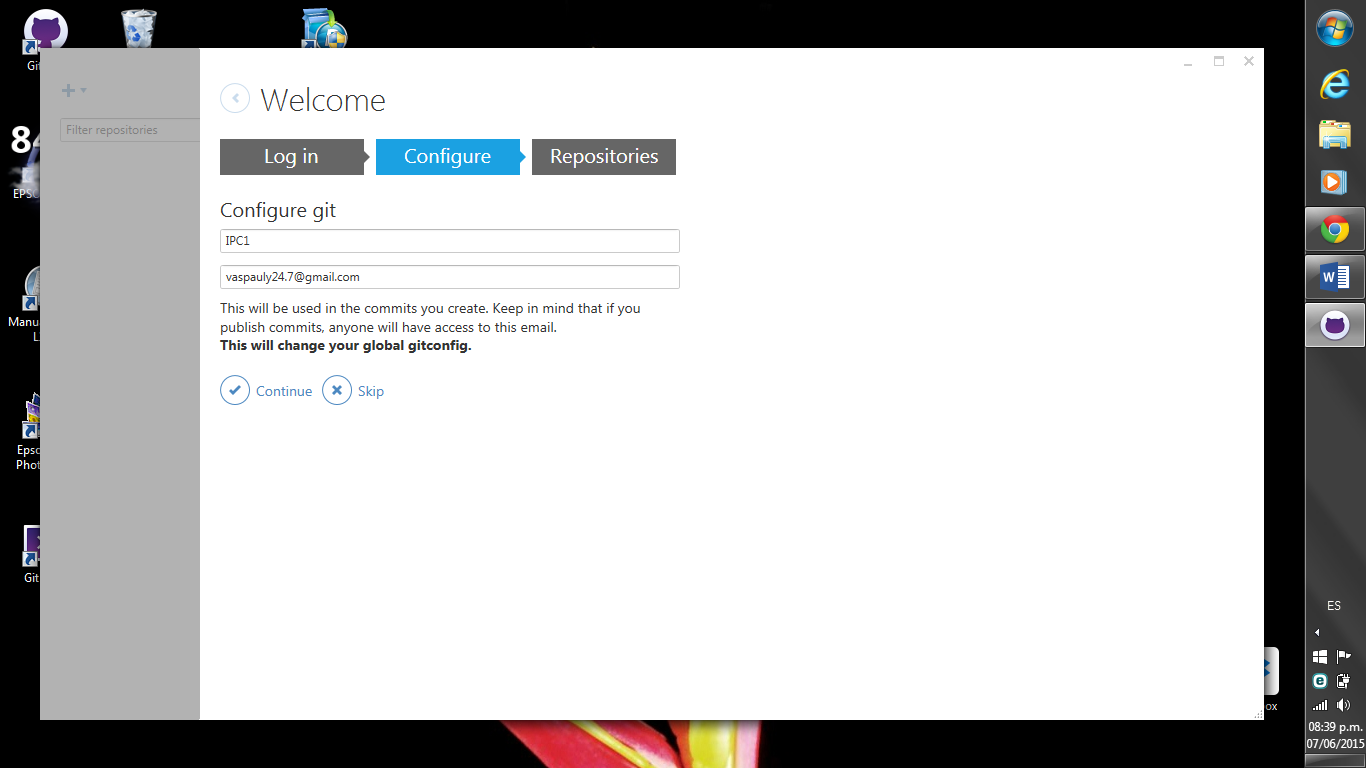
2.

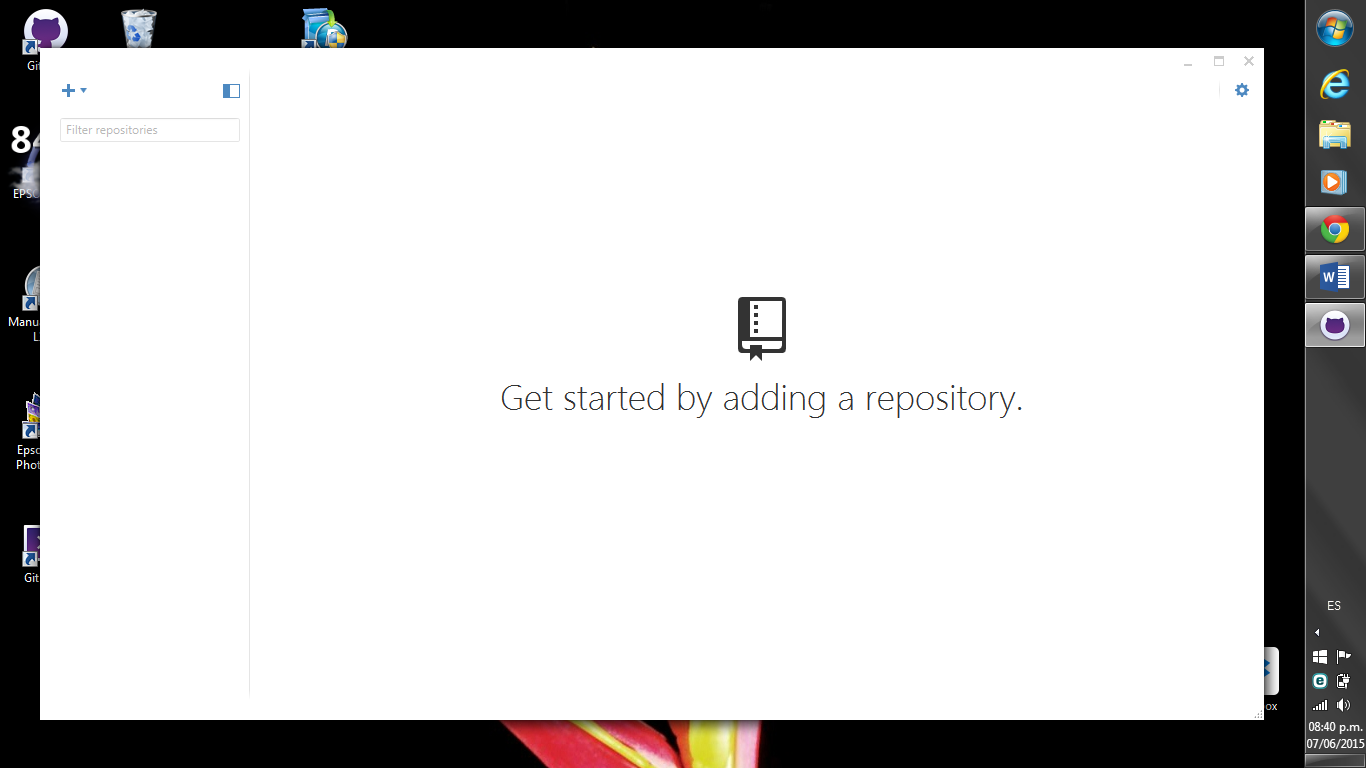
3.

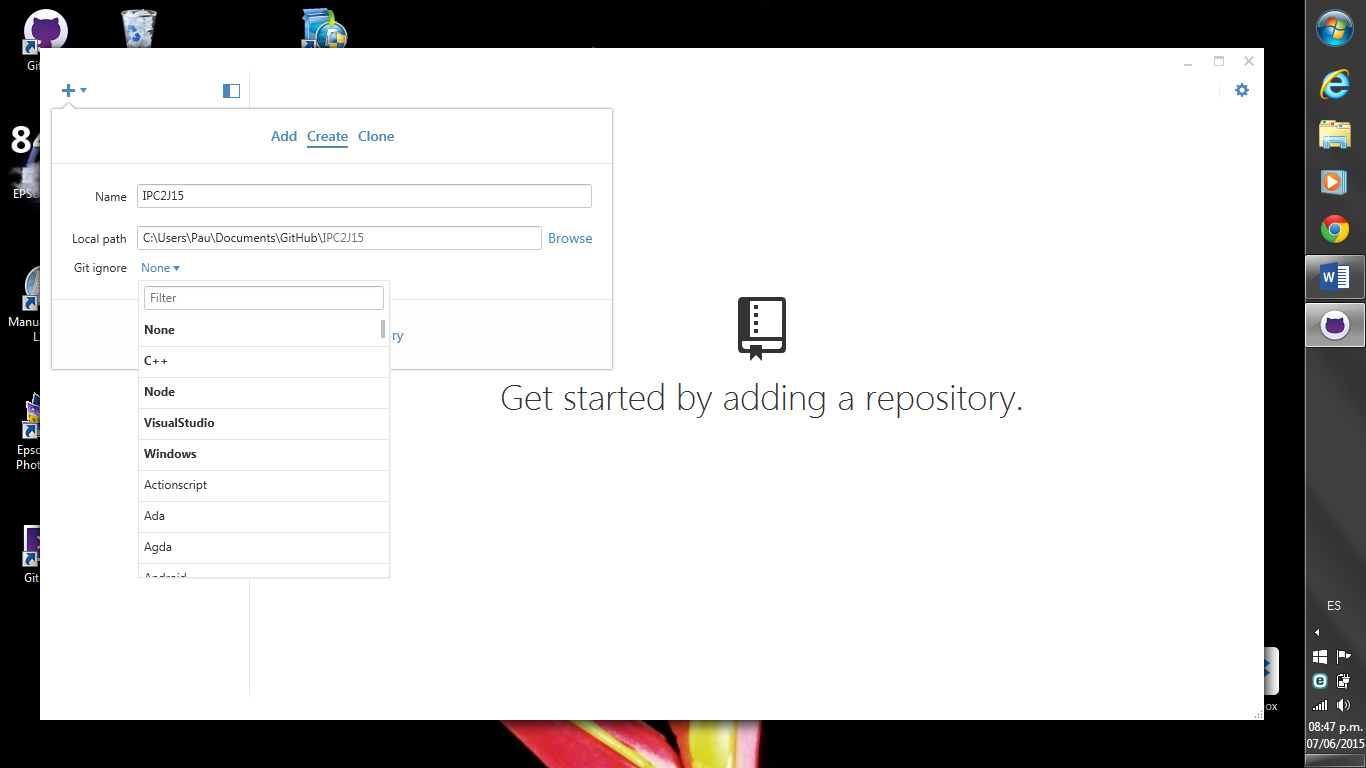
4.

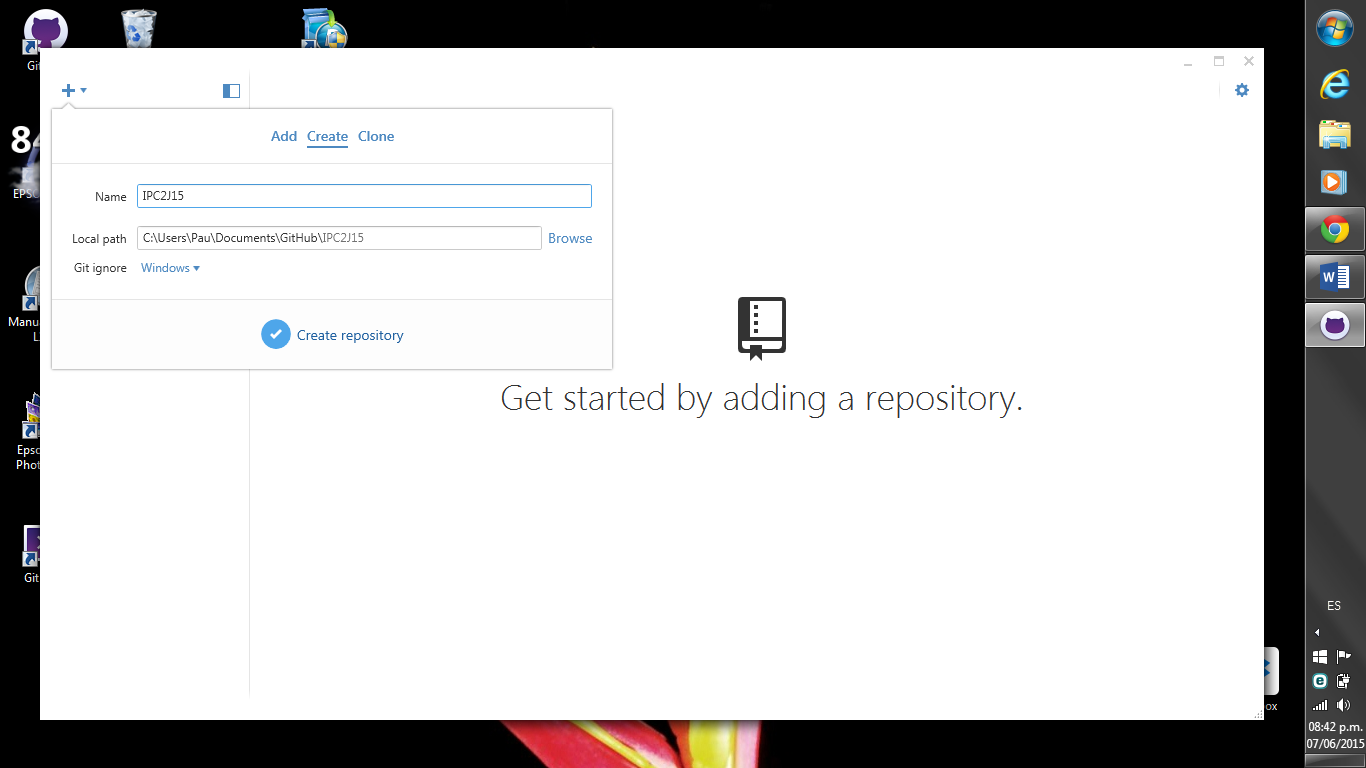
5. 

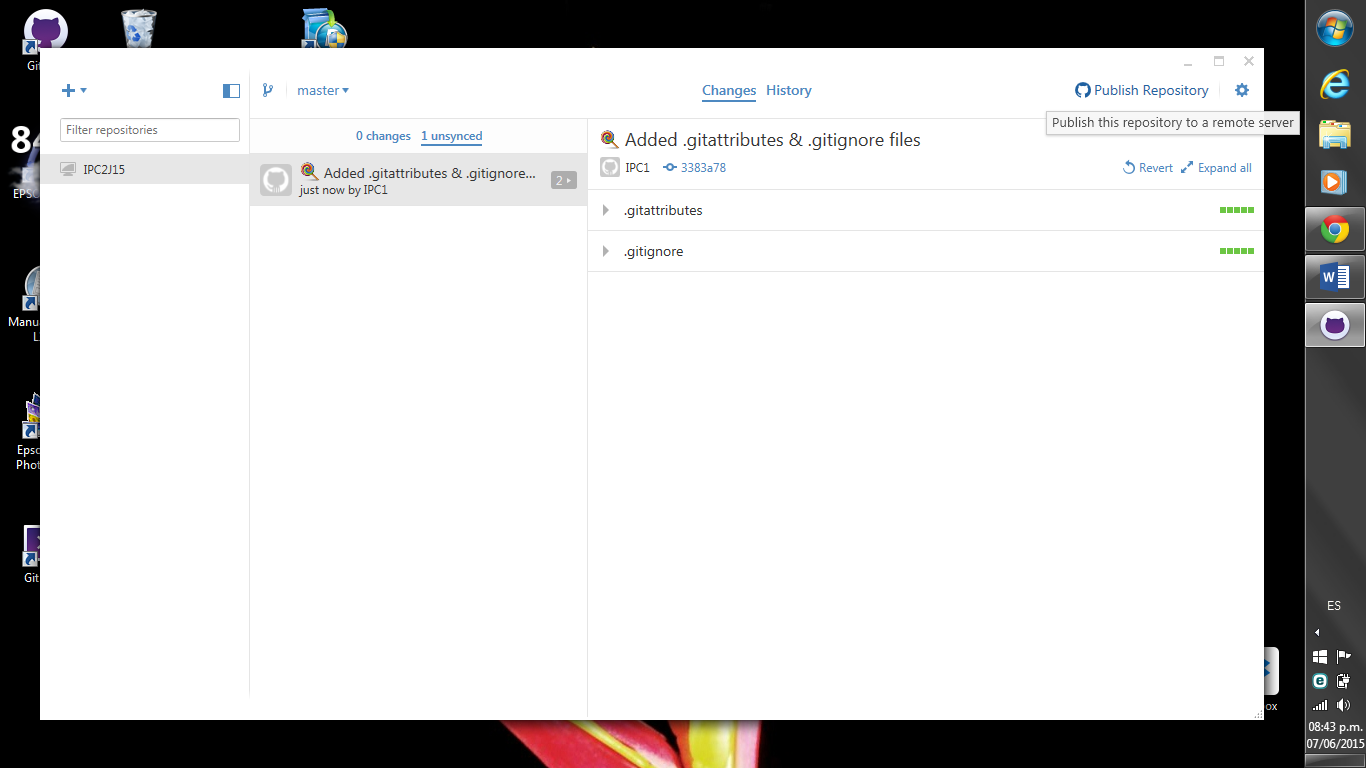
6. 

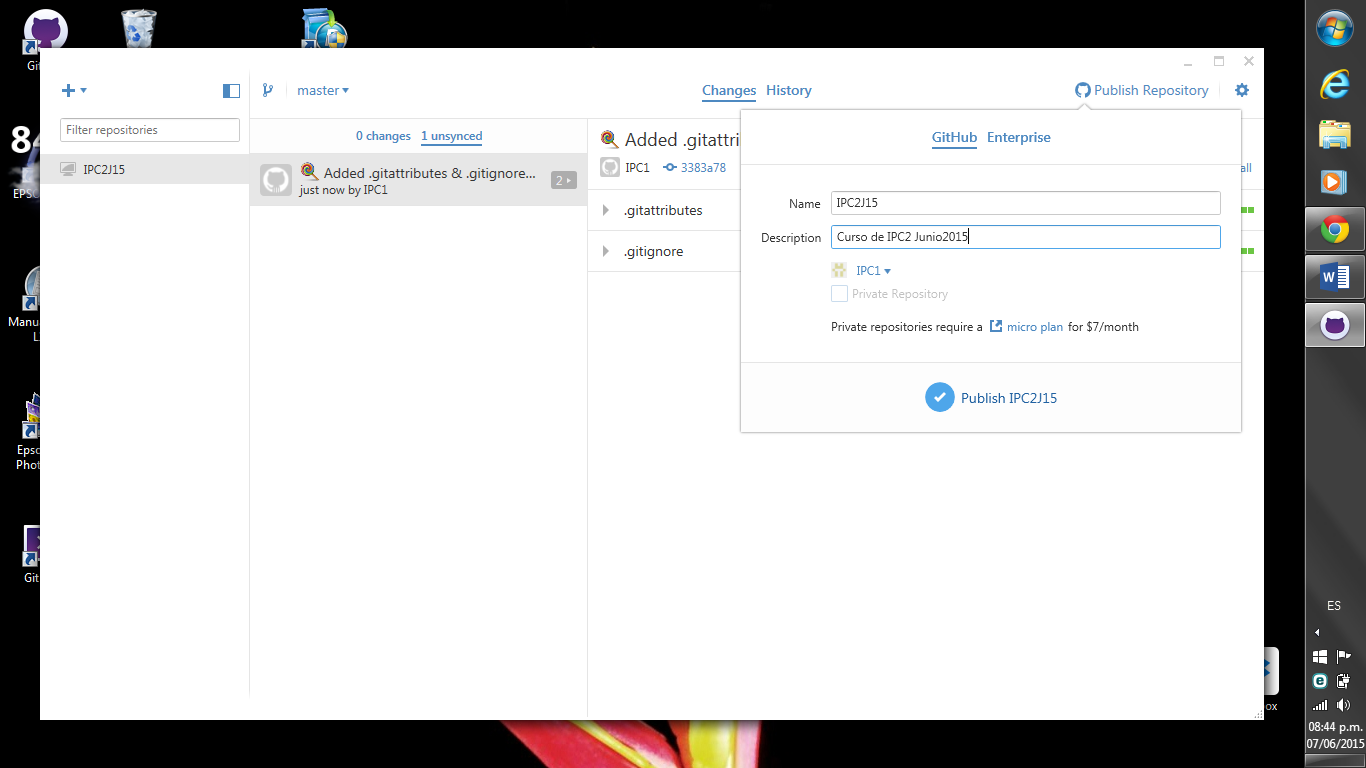
7. 

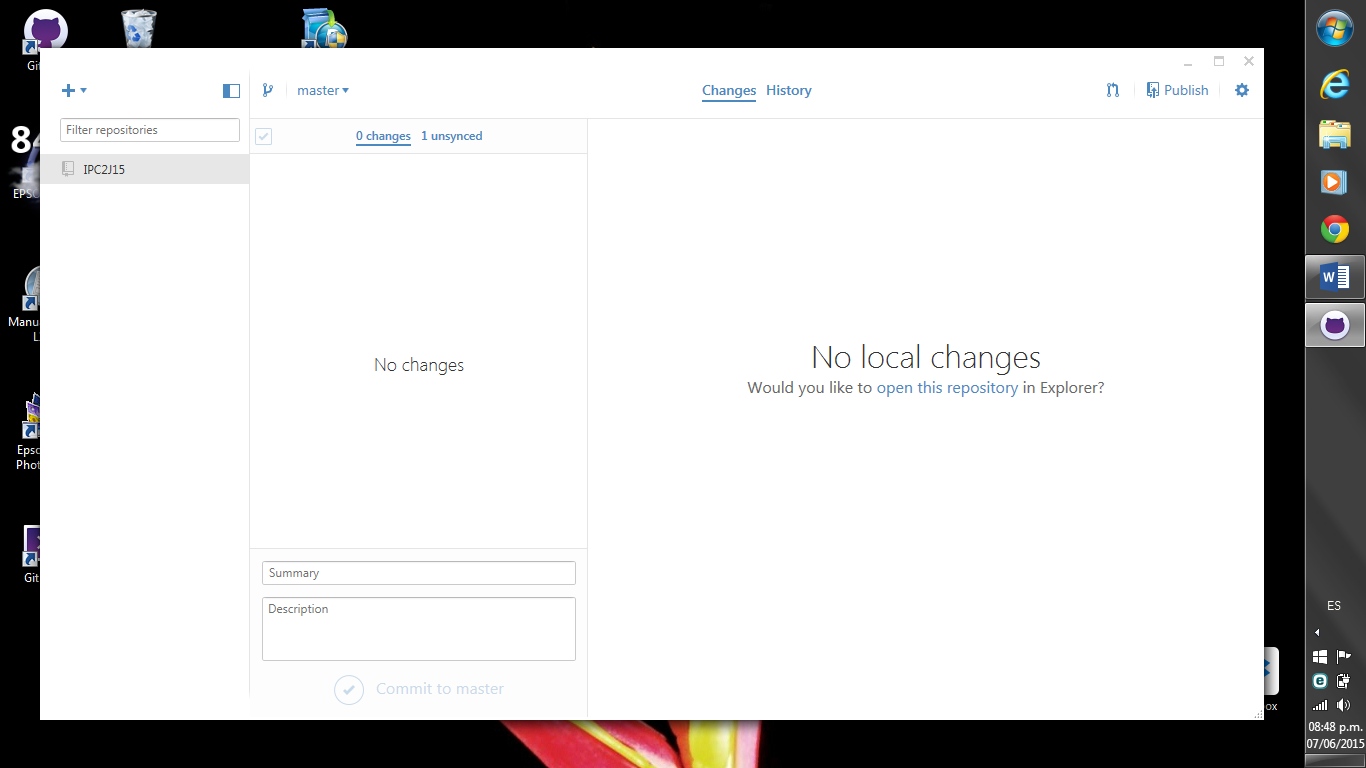
8. 

9. 

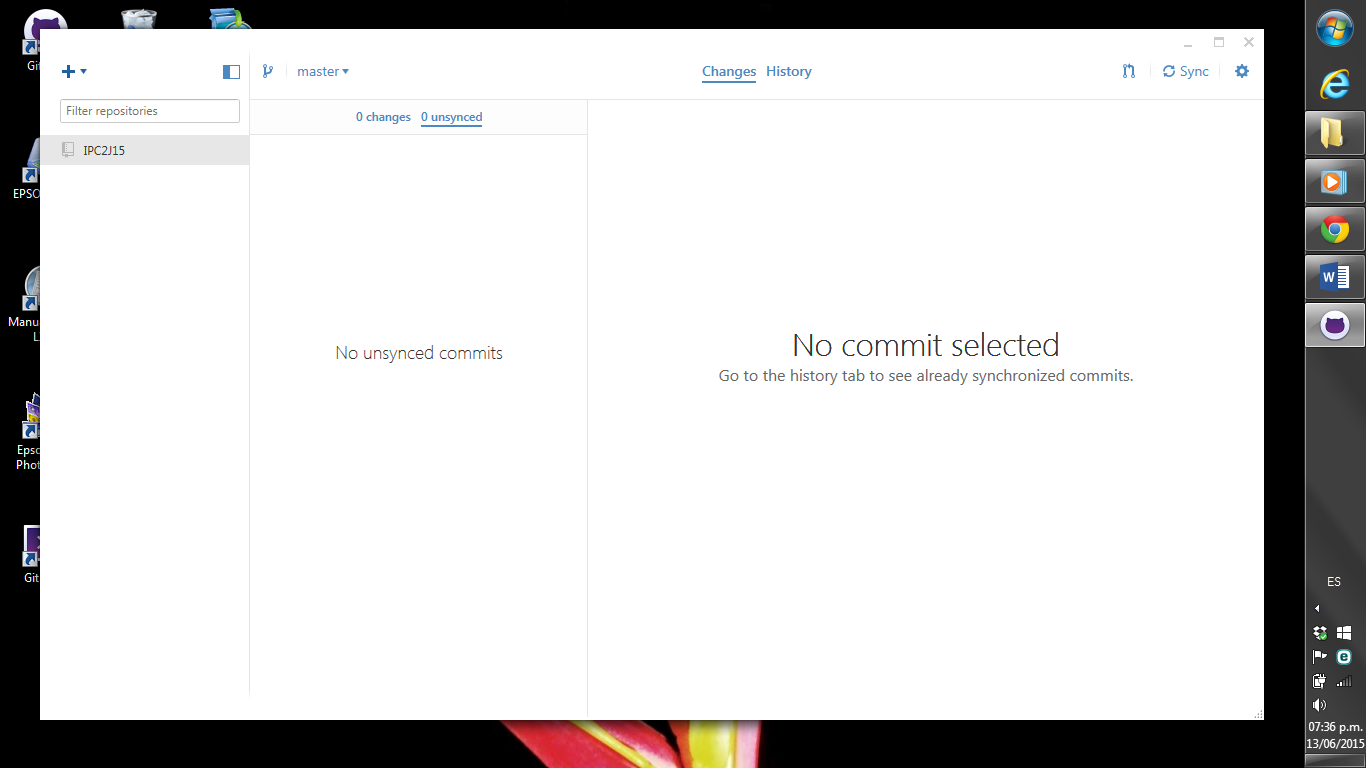
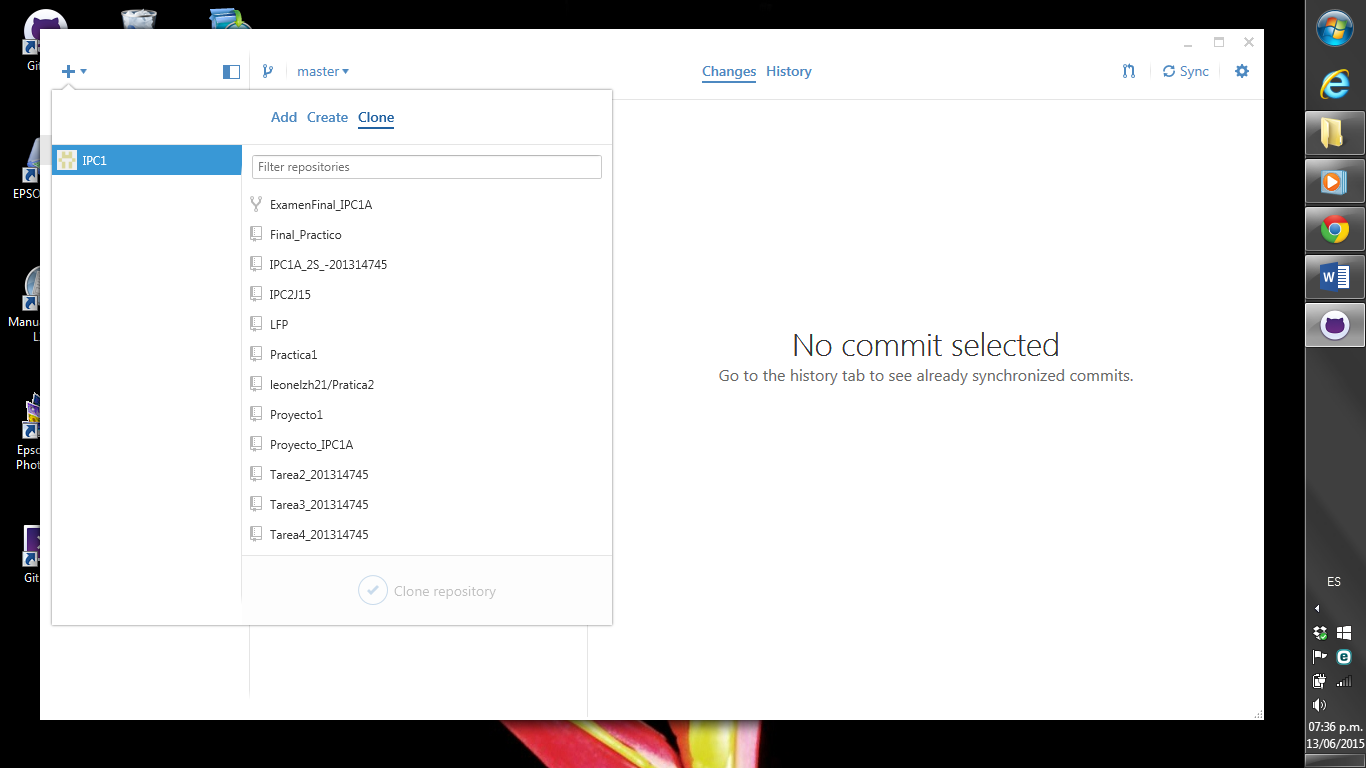
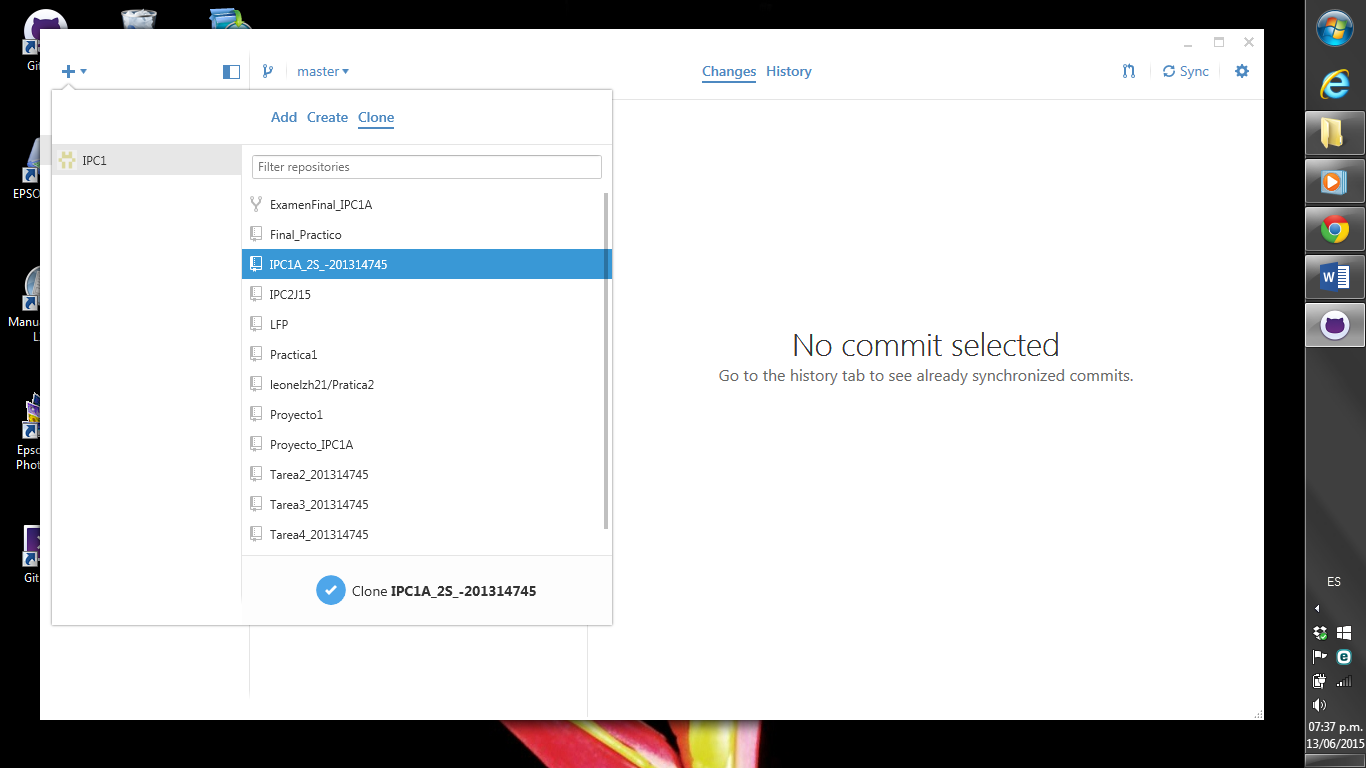
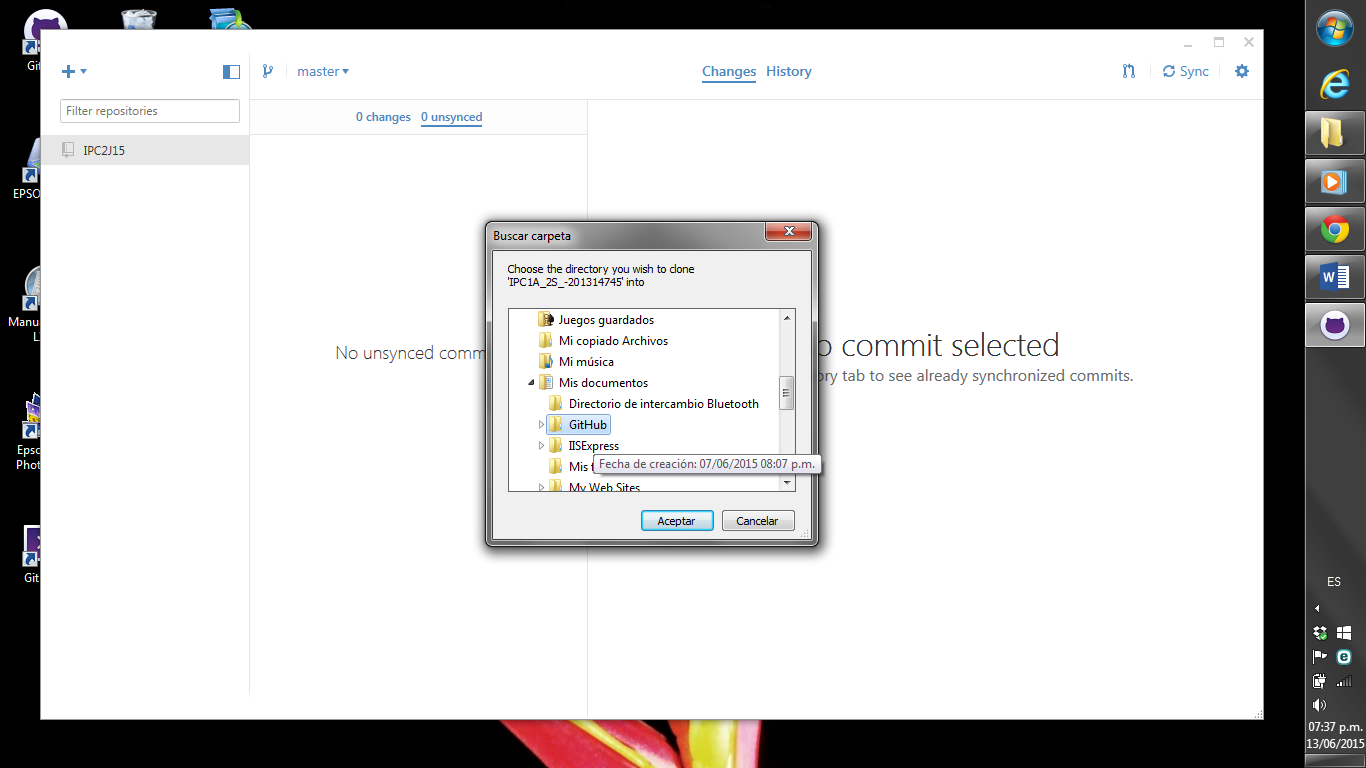
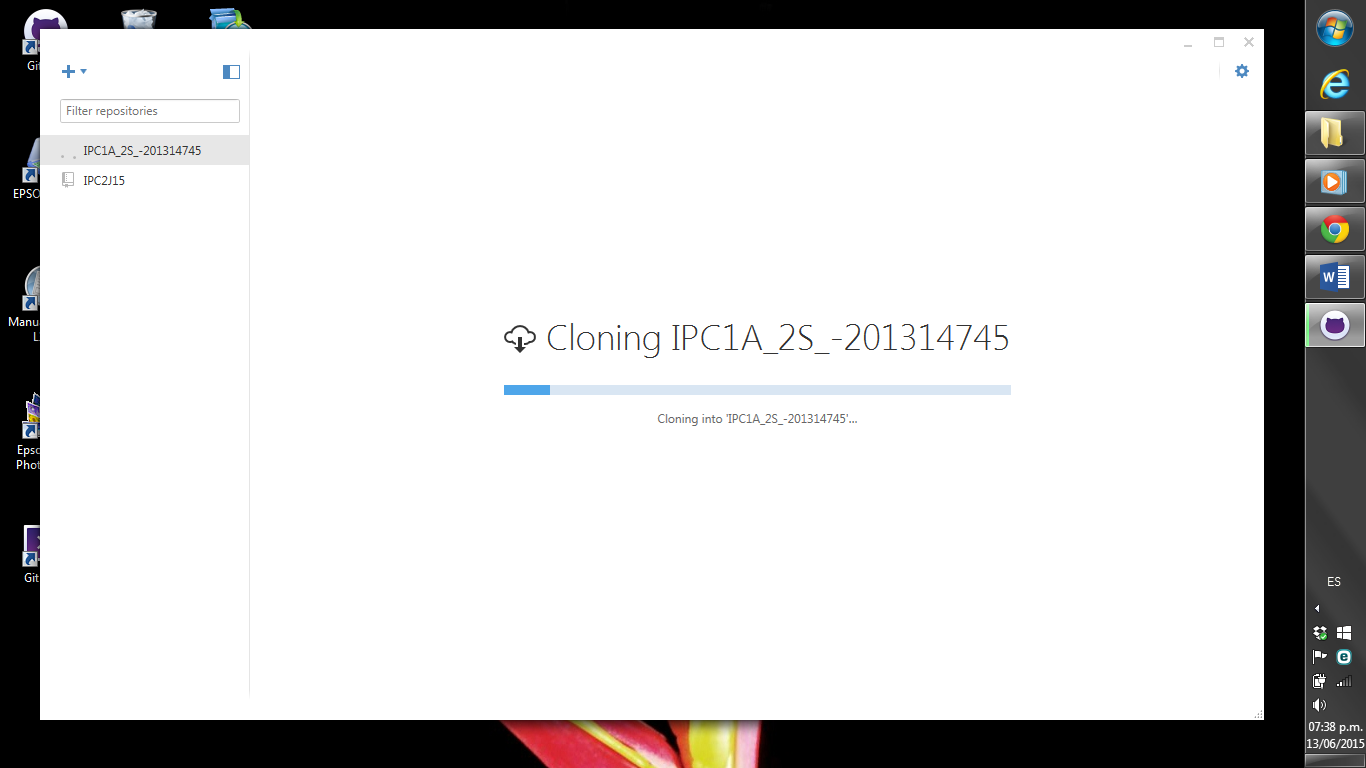
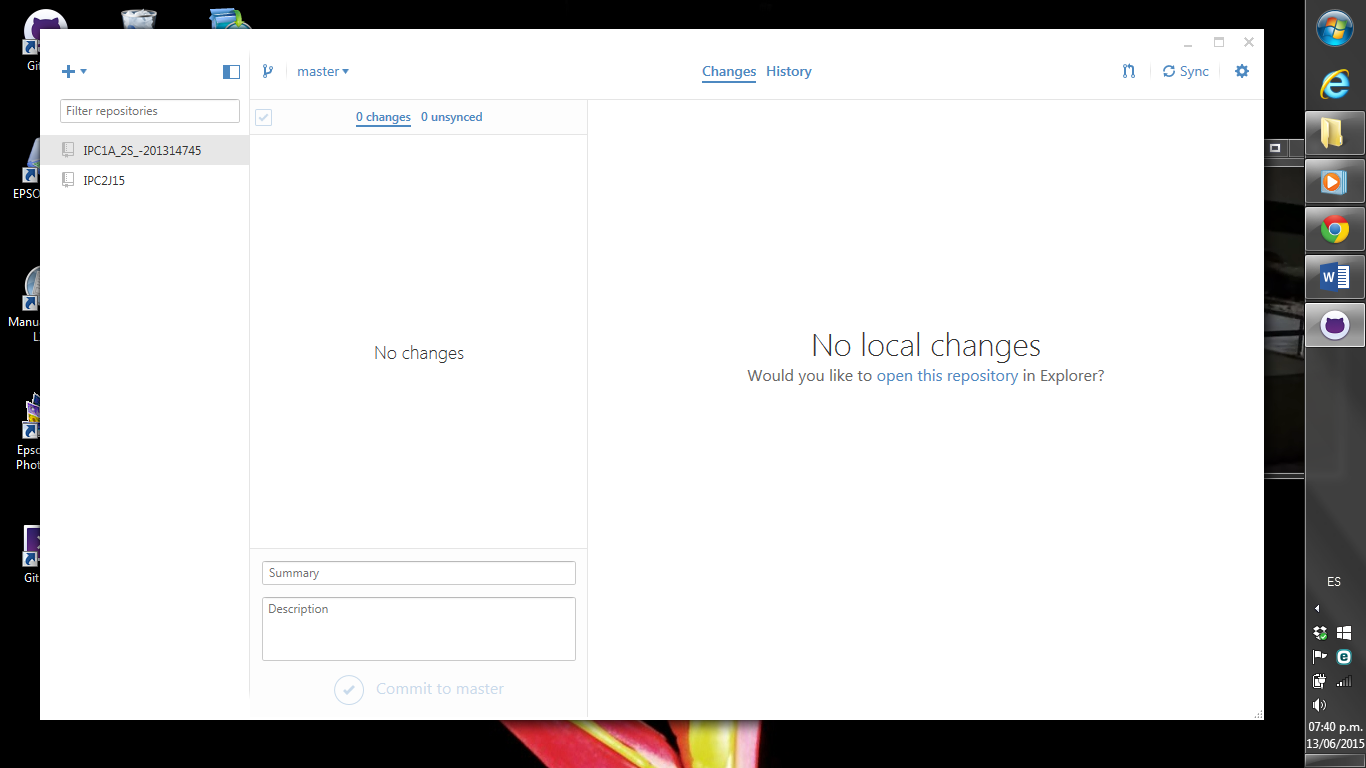
10. 

11. 

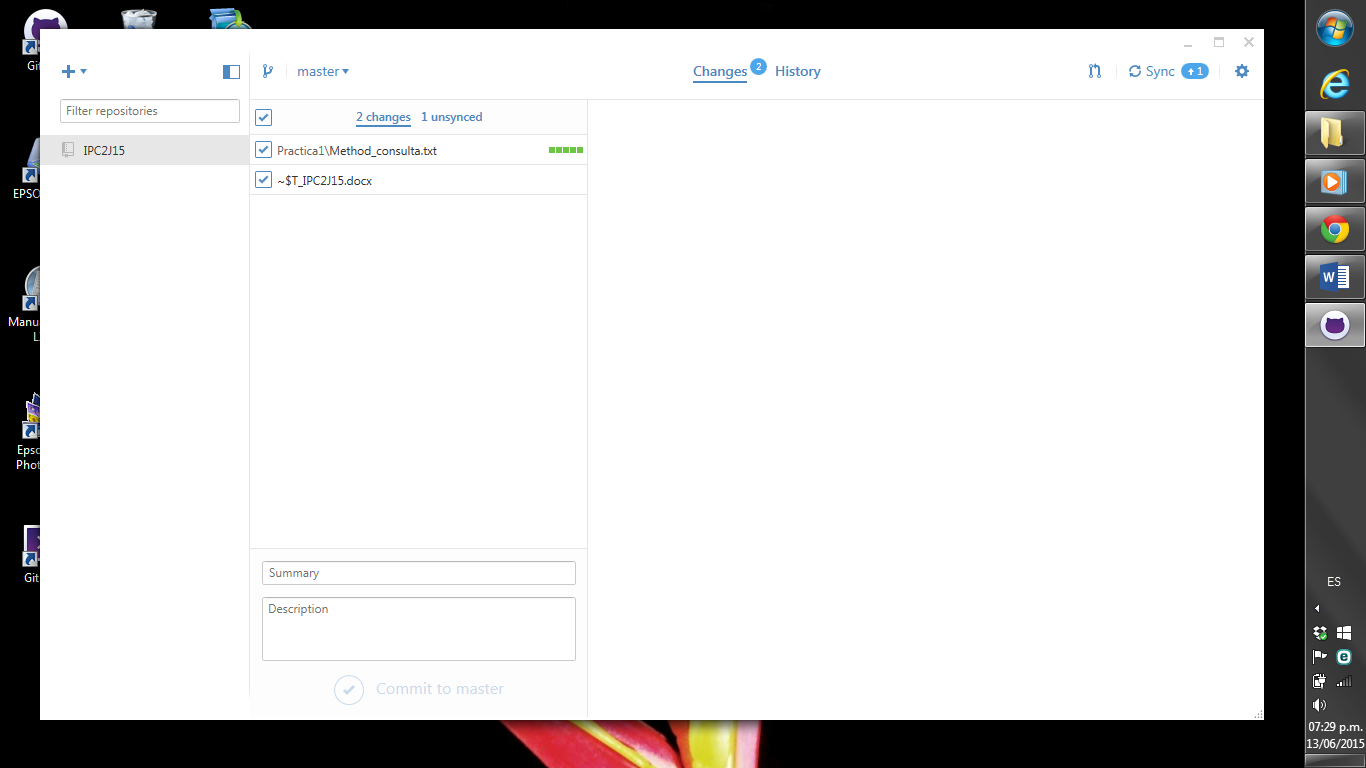
12. 

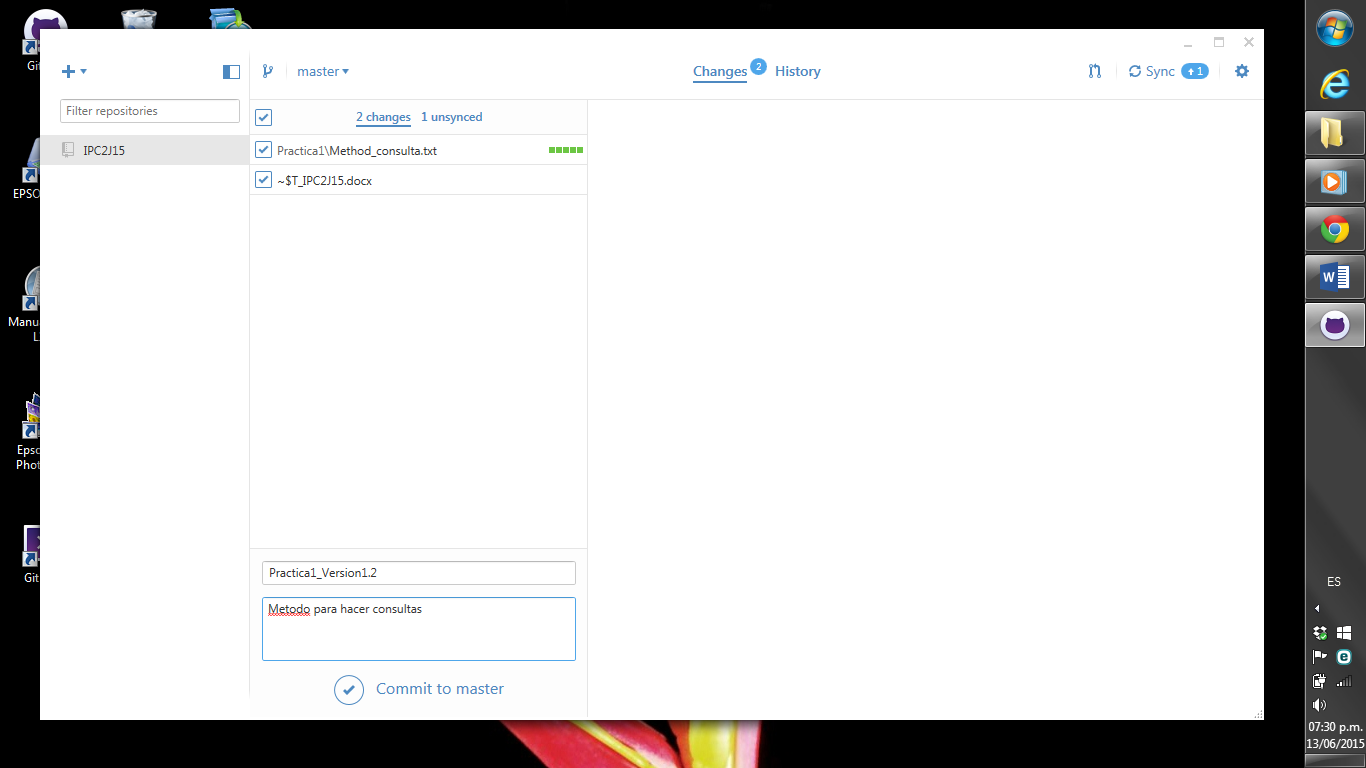
13. 

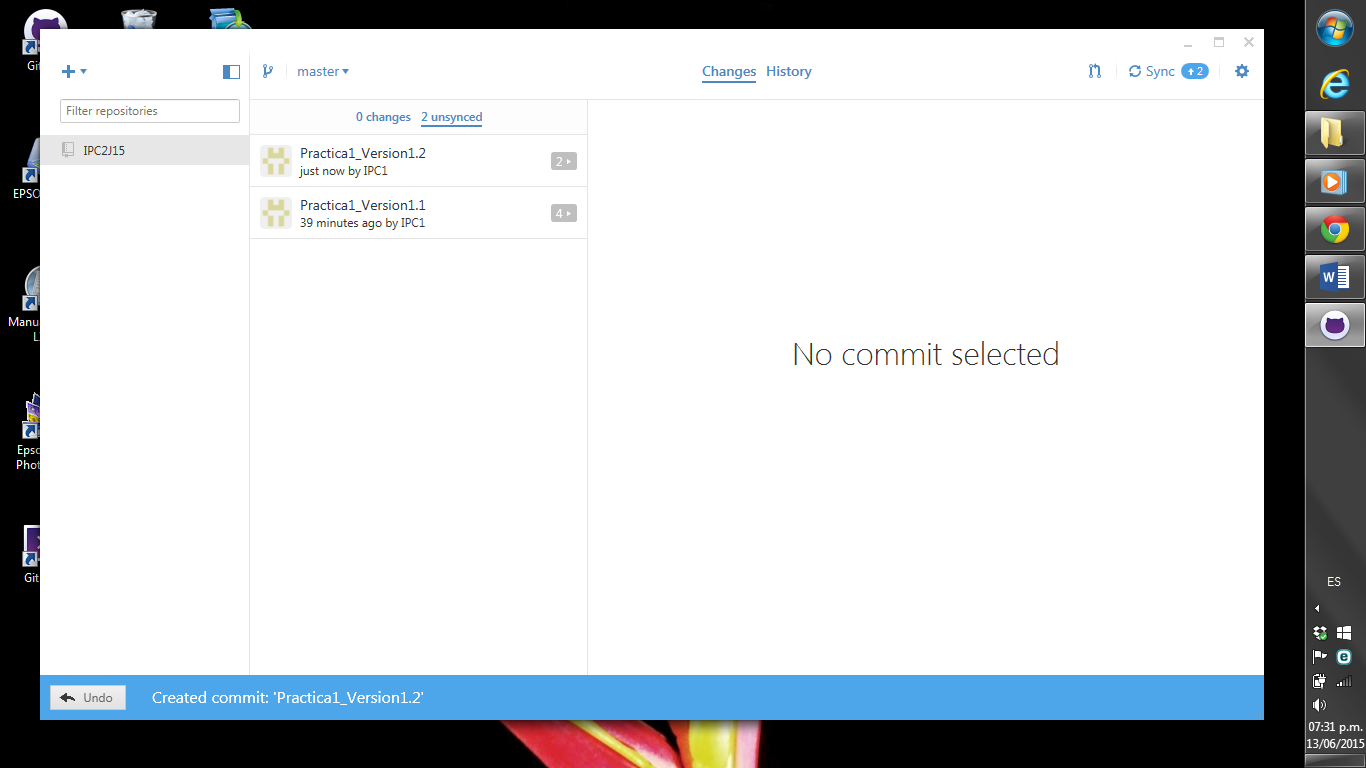
### Clonar repositorios

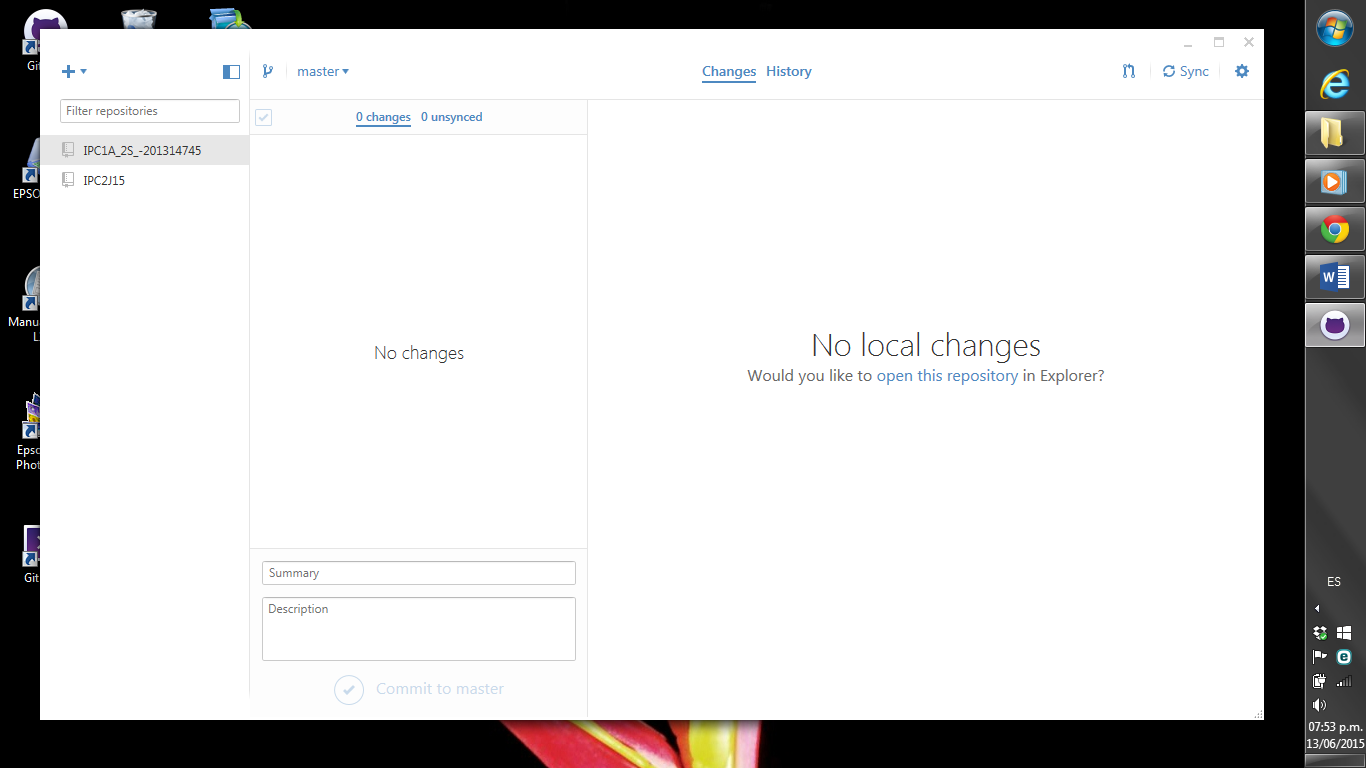
1. 
2. 
3. 
4. 
5. 
6. 
7. 

### Guardar y subir cambios

1. 
2. 

3. 

4.

5. 

1