

Universidad de San Carlos de Guatemala
Facultad de ingeniería
Escuela de Ciencias y Sistemas



Manual de uso de GitHub

Víctor Manuel Bámaca García
201313712

¿Qué es un sistema de control de versiones? Un sistema de control de versiones (CVS) te permite realizar un seguimiento de la historia de una colección de archivos y además incluye la funcionalidad de revertir la colección de archivos actual hacia una versión anterior. Cada versión podría considerarse como una fotografía del estado de la colección en un momento determinado de tiempo. La colección de archivos usualmente es código fuente de algún lenguaje de programación, sin embargo, un sistema de control de versiones funciona con cualquier tipo de archivo. Tanto la colección de archivos como su historia completa están guardadas en un repositorio (un repositorio es el nombre que recibe el lugar donde se aloja el código de un proyecto de desarrollo en algún lenguaje de programación).

¿Qué es un sistema de control de versiones distribuida? hay un servidor central para almacenar el repositorio y cada usuario puede hacer una copia completa del repositorio central mediante un proceso llamado “clonación”. Cada repositorio clonado es una copia completa del repositorio central y por ser una copia completa posee las mismas funcionalidades que el repositorio original, es decir, contiene la historia completa de la colección de archivos.

¿Qué es Git? Git es un sistema de control de versiones distribuida que se origina a partir del desarrollo del kernel de Linux, la principal diferencia entre Git y cualquier otro sistema de control de versiones es cómo Git modela sus datos. Conceptualmente, la mayoría de los demás sistemas almacenan la información como una lista de cambios en los archivos. Git modela sus datos más como un conjunto de instantáneas de un mini sistema de archivos. Cada vez que confirmas un cambio, o guardas el estado de tu proyecto en Git, él básicamente hace una foto del estado de todos tus archivos en ese momento, y guarda una referencia a esa instantánea.

Objetos de Git

- **Hay 4 tipos de objetos en git:**

1. **Blob:** su función es almacenar datos de archivos, la mayor parte del tiempo en un archivo en general.
2. **Tree:** Tecnicamente es como un directorio que permite hacer referencia a un conjunto de otros tree y/o blobs.
3. **Commit:** apunta a un determinado *tree(archivo)*, marcando como era en un momento determinado. Contiene información sobre ese momento determinado, los cambios del autor desde el último commit, el commit anterior (conocido como *parent*), etc. También se puede entender un commit, de una forma más imprecisa y coloquial, como la modificación o el conjunto de modificaciones a uno o varios archivos del repositorio. Otra forma de entenderlo también sería, como una "foto" de uno o varios archivos del repositorio en un momento determinado.
4. **Tag:** es una forma de marcar un commit como específico de alguna forma. Se usa normalmente para marcar algunos commits como reales específicos o algo destacables en esas líneas.

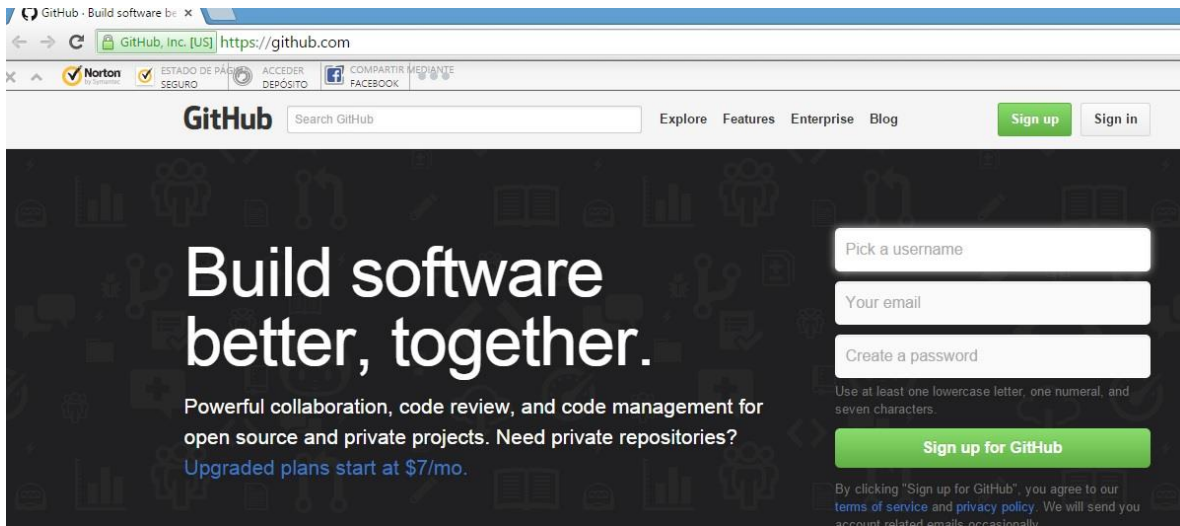
¿Qué es GitHub?

GitHub es una plataforma de desarrollo colaborativo de software para **alojar proyectos** usando el sistema de control de versiones Git. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago. También se pueden obtener repositorios privados (de pago) si se es estudiante.

GitHub no sólo ofrece alojamiento del código si no muchas más posibilidades asociadas a los repos como son, forks, issues, pull requests, diffs, etc. Se verán todos con detalle más adelante.

Realizar Registró en la página Oficial de Github:

1. Ingresar a la página github.com






2. Dar click en el botón “Sign Up”



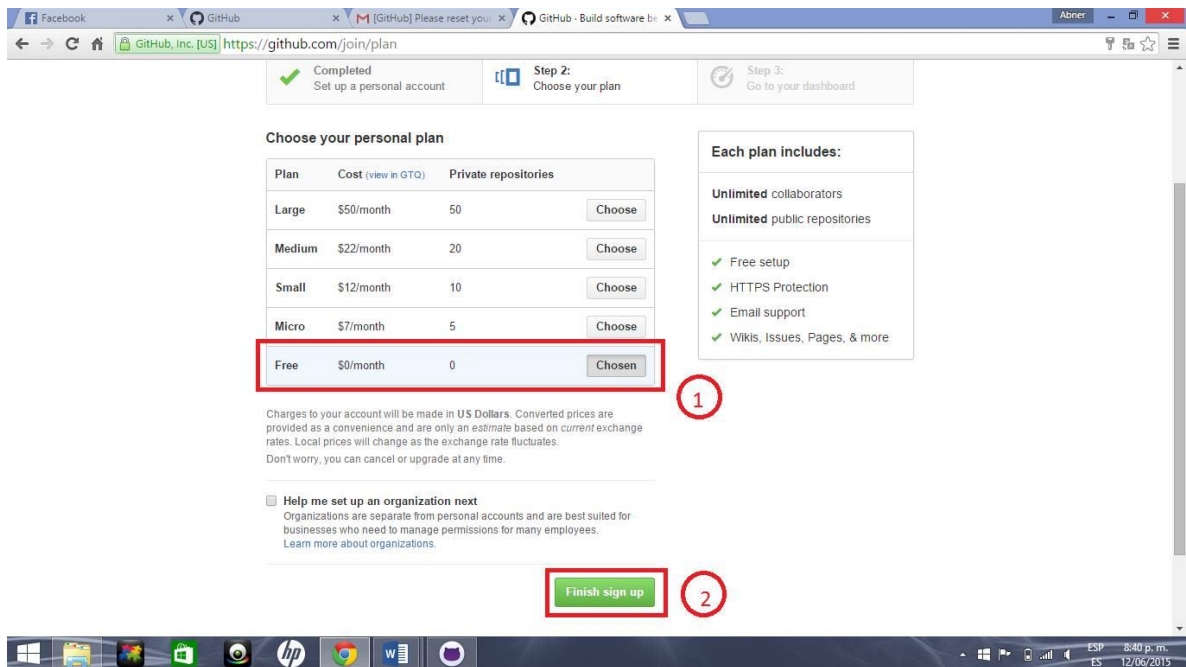
3. Ingresar los datos correspondientes que github le solicita.

Join GitHub

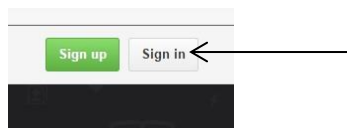
The best way to design, build, and ship software.

 Step 1: Set up a personal account	 Step 2: Choose your plan	 Step 3: Go to your dashboard
<h3>Create your personal account</h3> <p>Username</p> <input type="text"/> <p>This will be your username — you can enter your organization's username next.</p> <p>Email Address</p> <input type="text"/> <p>You will occasionally receive account related emails. We promise not to share your email with anyone.</p> <p>Password</p> <input type="password"/>		
<h3>You'll love GitHub</h3> <ul style="list-style-type: none">Unlimited collaboratorsUnlimited public repositories✓ Great communication✓ Friction-less development✓ Open source community		

4. Elegir el plan de costo con el cual desea trabajar (1) en este caso será gratuito. Dar click en finalizar registro (2).



5. Ya creado su registro puede ingresar a la pagina dando click en el botón sign in



6. Ingresar los datos para logiarse a la pag. De github.

Sign in

Username or Email

Password (forgot password)

Sign in

7. Pag. Principal después de ingresar a la cuenta.

Nombre de Username

Victor95

Welcome to GitHub! What's next? (on 6 Feb 2014)

- Create a repository
- Tell us about yourself
- Browse interesting repositories
- Follow @github on Twitter

ProTip! Edit your feed by updating the users you follow and repositories you watch.

Your repositories + New repository

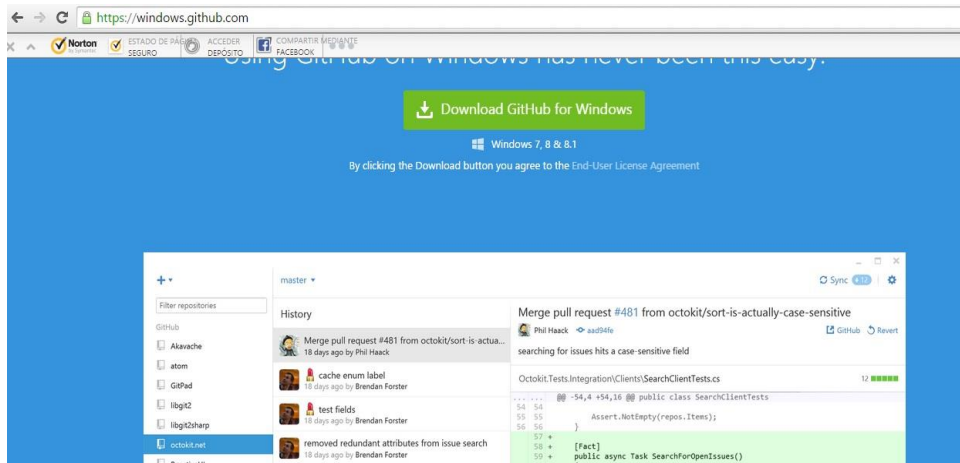
Find a repository...

All Public Private Sources Forks

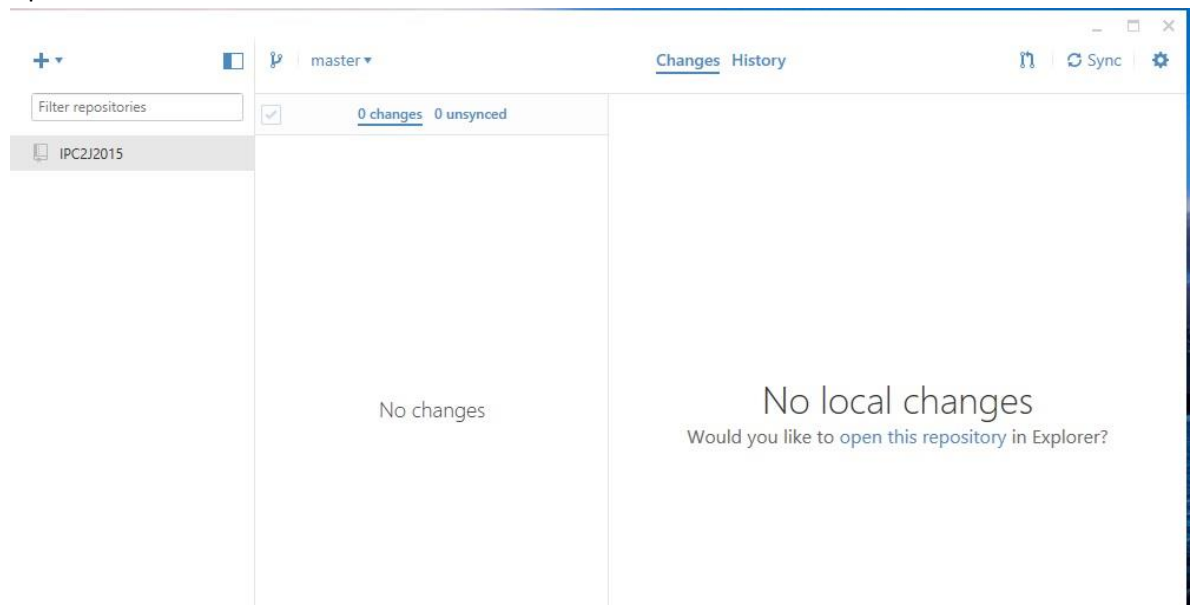
- IPC2J2015
- AlgoritmosEnJava
- victor

Aparecen todos los repositorios que fueron realizados.

8. Instalar la aplicación de Github para escritorio esto permite crear clonaciones y repositorios de una forma más práctico y fácil.
- Link para descargar dicha aplicación <https://windows.github.com/> teniendo en cuenta que esta versión es para Windows.

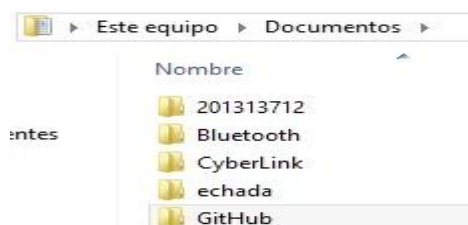


9. Aplicación de GitHub.

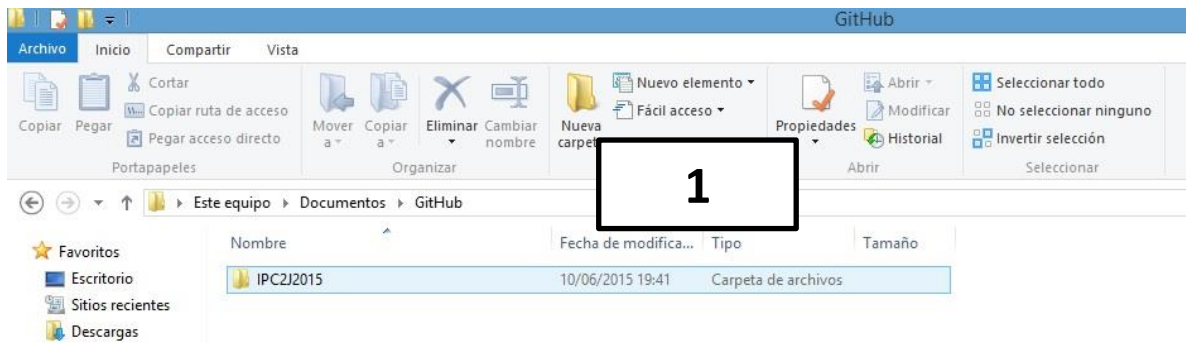


¿Cómo crear un repositorio desde mi aplicación para subir archivos a la página de GitHub?

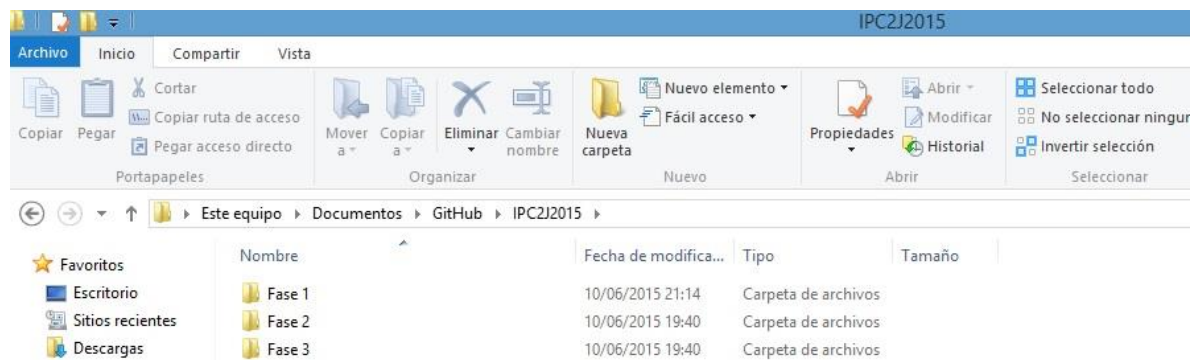
Cuando se termina de instalar la aplicación se crea una carpeta en mis documentos con el nombre de GitHub.



1. Creación de carpeta donde mejor le convenga o crear dentro de la carpeta de GitHub que se explicó anteriormente, este será el repositorio, pero antes hacemos un recordatorio de asegurarse de descargar la aplicación de GitHub para Windows o para el sistema operativo con el que cuente.



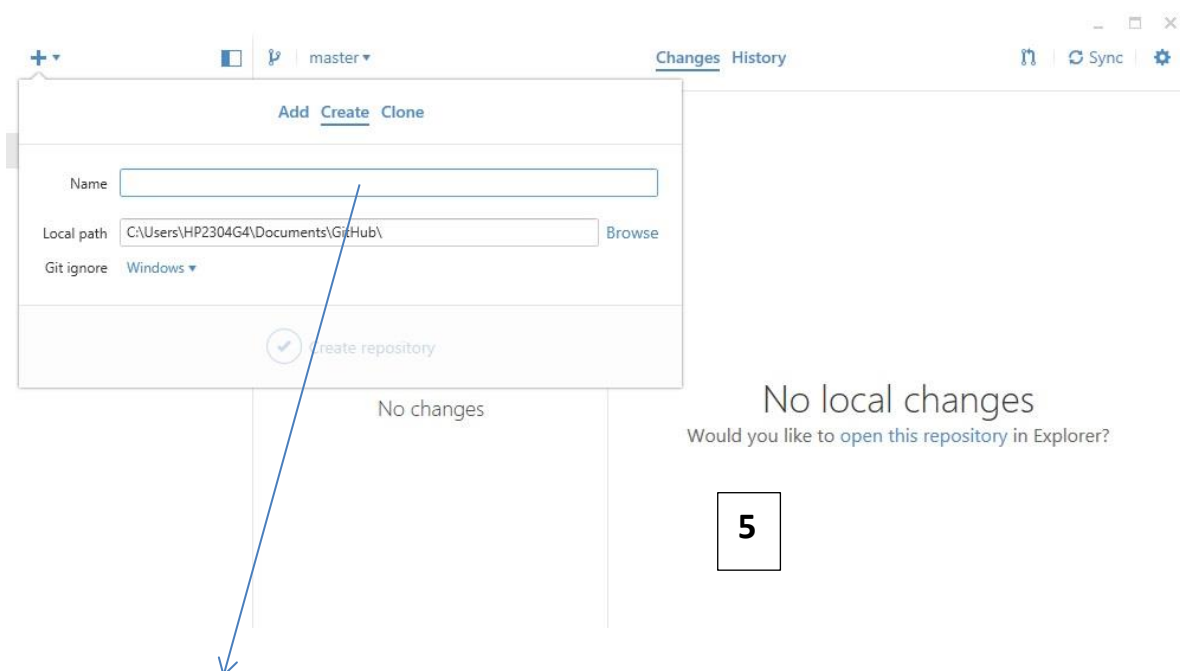
2. Segundo se creara las carpetas que irán dentro del repositorio y la vez agregarle un archivo .txt con cualquier información para poder realizar este paso ya que si no posiblemente no aparezcan las carpetas que fueran creadas por no contener ningún archivo adentro.



3. Abrir la aplicación GitHub y cuando se abra por primera vez pedirá iniciar sesión entonces hacer ese paso para que no haya ningún problema.
4. Arrastrar la carpeta Creada con anterioridad hacia la aplicación GitHub.

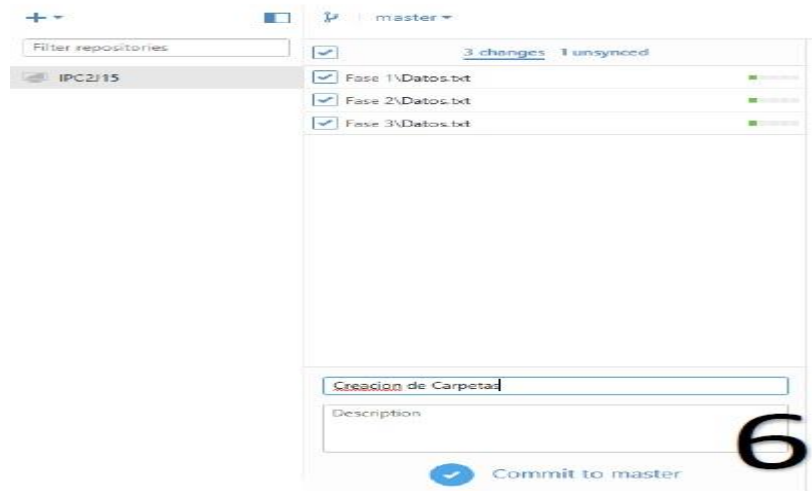


5. Cuando se arrastre la carpeta solo tomara referencia de la ubicación de la carpeta y desplegara lo siguiente que será el crear el Repositorio. Se Coloca el nombre que normalmente agarra el nombre de la carpeta con el que se creó con anterioridad. Cuando ya se tenga todo entonces se le da "Create Repository".



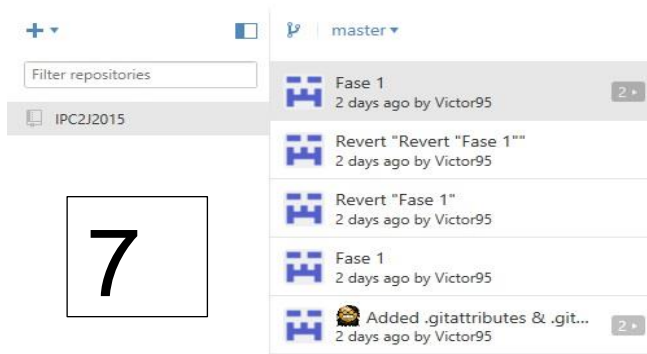
Se ingresa el nombre con el cual desea para crear el repositorio.

6. La aplicación tomara por si sola los objetos, archivos que contengan las carpetas y se mostrar cómo se puede apreciar. Ya con esto En la parte se debe de hacer un Commit Esto se refiere como a un tipo de nombre de guardado con el que todos tendrá acceso a la hora de que el repositorio sea público que normalmente Se pone así. Presionar “Commit to master”.

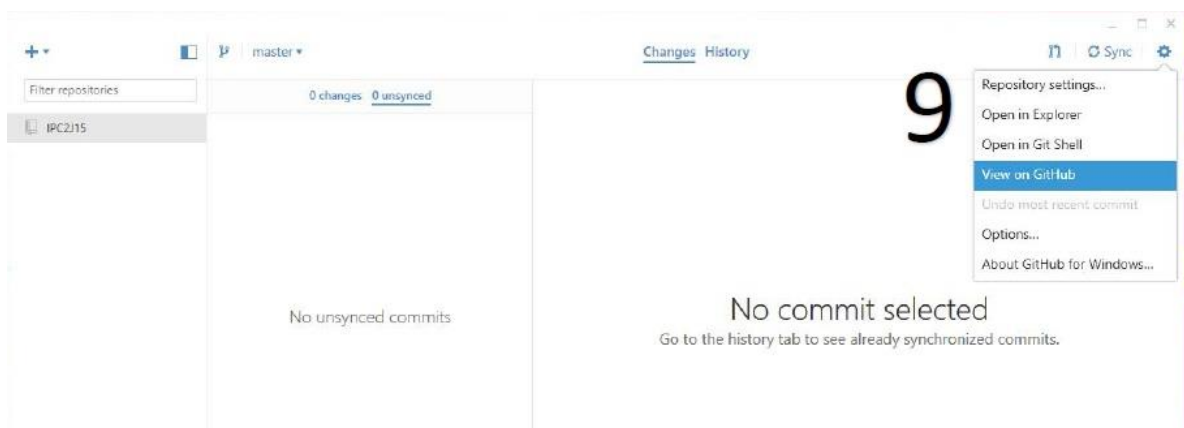


7. En esta parte mostrara las carpetas creadas que son los repositorios. De esta misma manera se agregaran carpetas o algún tipo de archivo que se desee.
8. Ya teniendo todo lo que se cargara al principio que es únicamente el Repositorio Principal Habrá que darle Click en la parte donde dice “Publish Repository” cuando se de click en esa parte se procederá a agregarle una descripción y a darle click en “Publish (Nombre de repositorio).”





9. En esta parte se dara click en La tuerca que aparece de lado derecho en la parte superior y luego se procede a dar click en “View on GitHub” esto para verificar que los archivos si fueron cargados correctamente.



10. Ya por ultimo aparecerá de esta manera en el explorador que tenga predeterminado y con esto se concluye de que si fue creado con éxito el Repositorio en GitHub.



Description

Short description of this repository

Website

Website for this repository (optional)

Save

or Cancel

<> Code

Issues

Pull request

Wiki

Pulse

Graphs

Settings

5 commits

1 branch

0 releases

1 contributor



branch: master

IPC2J2015 / +



Fase 1



Victor95 authored 2 days ago

latest commit 774d28fe5f



Fase 1

Fase 1

2 days ago



.gitattributes



Added .gitattributes & .gitignore files

2 days ago



.gitignore



Added .gitattributes & .gitignore files

2 days ago