|  |
| --- |
| REPOSITORIOS CON GITHUB |
| APLICACIONES AVANZADAS WEB |
| Eduardo Torres Ponguta |

**¿Qué es Git?**

Git es un software de control de versiones. El control de versiones, resumiéndolo mucho, es la gestión de los diversos cambios que se realizan sobre 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) Git usa cadenas de 40 caracteres llamadas SHA para identificar los cambios realizados. Ej: 74be3c2b93c8267de64e6dfde8658d31d7d2fe6c

**Objetos de Git.**

Hay 4 tipos de objetos en git (el más importante a entender es el commit).

Blob: se usa para almacenar datos de archivos, es generalmente un archivo.

**Tree:** es, básicamente, como un directorio, hace referencia a un conjunto de otros trees y/o blobs (por ej. archivos y subdirectorios).

**Commit:** apunta a un determinado tree, marcando como era en un momento determinado (quien no haya entendido lo que es un tree, sustituya la palabra tree por archivo). 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.

Tag: es una forma de marcar un commit como específico de alguna forma. Se usa normalmente para marcar algunos commits como releases específicos o algo destacable en esas líneas.

**Procedimiento**

En primer lugar ha de crearse una cuenta en Github.

El siguiente paso será instalar git.

Para Mac OS X -> Git

Para Windows -> Git

Antes de pasar a explicar el proceso de enlace del pc con Github quiero dejar algo claro. Git es un programa que se instala en nuestro pc (hasta ahí todo claro supongo :), pero hay que tener en cuenta que no todos los programas tienen una interfaz gráfica (gui). En el caso de git en mac no hay gui (aunque sí hay programas externos que permite manejar git a través de una gui) y en el caso de windows git sí viene incluido con una gui. Sin embargo, tanto en mac como en windows, no es recomendable trabajar con git mediante una interfaz gráfica. En vez de eso se trabajará mediante línea de comandos por terminal (en mac) / cmd (en windows).

**Set Up Clave SSH**

Las claves SSH se usan para establecer una conexión segura entre el pc y Github.

Para crear la clave se ha de seguir los siguientes pasos en terminal (en mac) o en cmd (en windows).

ssh-keygen -t rsa -C "your\_email@youremail.com" (sustituir "your\_email@youremail.com" por el email que se puso al crear la cuenta de github).

Esto devolverá lo siguiente:

Generating public/private rsa key pair. Enter file in which to save the key. C/Users/your\_user\_di

Hacemos click en enter porque queremos que guarde la clave en el directorio por defecto. Guardará la clave ssh en el directorio .ssh dentro de nuestra carpeta de usuario.

Ahora nos dirá: Enter passphrase (empty for no passphrase) No hay por qué indicar nada. Enter.

Y nos dirá que la volvamos a introducir o darle a enter si lo dejamos en blanco antes.

Esto devolverá algo así.

Your identification has been saved in /Users/your\_user\_directory/.ssh/id\_rsa. Your public key has been saved in /Users/your\_user\_directory/.ssh/id\_rsa.pub

Situar la clave SSH en Github.

Se deberá dirigir a Github y hacer click en "Add another pubic key". Control+v (Cmd+v en mac) en la zona de "Key" para situar ahí nuestra clave ssh. Se pone cualquier título en "Title". Por ej: algo para reconocer el pc del que es esa clave (Ej: Macbook Pro de Jesús).

**Crear un Repositorio**

En primer lugar se deberá crear el repositorio en Github, haciendo click en donde dice New Repository. Ahí se indicará el nombre del proyecto/repositorio (Project Name) y opcionalmente una descripción del mismo (Description) y un link a la web del proyecto (Homepage Url). También se puede indicar si el repositorio será público, siendo visible para todo el mundo, o bien privado, que solo será accesible para el usuario que lo crea y el o los contribuidores.

Una vez creado el Repo en Github se deberá crear en el PC siguiendo estos pasos:

Crear el directorio (carpeta) en donde se alojará el repo. Para ello recomiendo crear una carpeta en nuestra carpeta de usuario (mac) / disco local c (windows) y alojar ahí todos los repos.

En línea de comandos (terminal/cmd) navegar hasta el repo. Deberemos llegar hasta el repositorio mediante la línea de comandos, usando el comando cd. Ej:

<Macbook-Pro-de-Jesus:~ Jesus$ cd github>

<Macbook-Pro-de-Jesus:github Jesus$ cd Guia-sobre-Git-Github-y-Metodologia-de-Desarrollo-de-Software-usando-Git-y-Github>

<Macbook-Pro-de-Jesus:Guia-sobre-Git-Github-y-Metodologia-de-Desarrollo-de-Software-usando-Git-y-Github Jesus$>

Inicializar el repo de git. Una vez se llega al que será el directorio del repo se debe introducir lo siguiente para iniciar el repositorio en git.

<git init>

Añadir la dirección remota del repositorio de github. Se deberá indicar dónde estará alojado el repo.

<git remote add origin git@github.com:nombreusuario/nombrerepo.git>

Ej: [git@github.com:Hispano/test.git](mailto:git@github.com:Hispano/test.git)

**Clonar un Repositorio**

Esto es muy simple. Una vez se llegue al directorio donde se va a clonar el repo vía línea de comandos, se debe indicar lo siguiente.

<git clone linkhttp>

Tercera Parte: Git-> Realizar cambios, revertirlos y subirlos al repositorio remoto

Esta parte comprende el workflow que se realiza con normalidad cuando se está trabajando en un repositorio manejando Git. La dividiré en las siguientes partes:

Añadir uno o varios archivos a un commit.

Realizar un commit con su correspondiente mensaje.

Subir los cambios al repositorio remoto alojado en los servidores de GitHub.

Revertir los cambios realizados por dicho commit.

Para apreciar los cambios que se producen en el repositorio mientras se realizan estas acciones se utilizará el comando <git status> Si no hemos realizado ningún cambio en el repositorio, al usar ese comando nos debería mostrar algo así:

Guia-sobre-Git-Github-y-Metodologia-de-Desarrollo-de-Software-usando-Git-y-Github git:(master) git status> <On branch master> <nothing to commit (working directory clean)>