Primeros pasos con github: subir un proyecto al repositorio.

0. Índice de contenidos.

* [1. Introducción.](https://www.adictosaltrabajo.com/tutoriales/github-first-steps-upload-project/#01)
* [2. Entorno.](https://www.adictosaltrabajo.com/tutoriales/github-first-steps-upload-project/#02)
* [3. Preparar el entorno de git.](https://www.adictosaltrabajo.com/tutoriales/github-first-steps-upload-project/#03)
* [4. Preparar el entorno de github.](https://www.adictosaltrabajo.com/tutoriales/github-first-steps-upload-project/#03)
* [5. Subir el proyecto.](https://www.adictosaltrabajo.com/tutoriales/github-first-steps-upload-project/#05)
* [6. Referencias.](https://www.adictosaltrabajo.com/tutoriales/github-first-steps-upload-project/#06)
* [7. Conclusiones.](https://www.adictosaltrabajo.com/tutoriales/github-first-steps-upload-project/#07)

1. Introducción

[Github](https://github.com/) es un hosting para [git](https://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=git" \t "_blank), gratuito para proyectos opensource. Git es un sistema de control de versiones distribuido, para más info leed este [tutorial de Alejandro](https://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=git).

En este vamos a ver cómo configurarnos el entorno local para trabajar contra github y cómo realizar una primera subida de un proyecto al repositorio.

2. Entorno.

El tutorial está escrito usando el siguiente entorno:

* Hardware: Portátil MacBook Pro 17′ (2.93 GHz Intel Core 2 Duo, 4GB DDR3 SDRAM).
* Sistema Operativo: Mac OS X Snow Leopard 10.6.7
* Git 1.7.4.4.

3. Preparar el entorno de git

Lo primero, siguiendo el tutorial que os he comentado de Alejandro, es instalarnos en local git y configurarlo.

Para la instalación, en función del sistema operativo, tendremos que realizar más o menos operaciones, el objetivo no es compilar nosotros el código con lo que nos descargaremos una empaquetación del mismo lista para instalar, para Mac, de la siguiente ubicación: <http://git-scm.com/>

La instalación es simple, tras la misma comprobaremos la versión instalada ejecutando por línea de comandos:

|  |  |
| --- | --- |
| 1 | git --version |

Lo siguiente es configurar el usuario, email y los decoradores de la salida por consola, para ello seguiremos los pasos aquí indicados [configuración de git](https://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=git#4.%20Configuraci%C3%B3n%20de%20Git%7Coutline).

Añadiremos a la configuración anterior la creación de un fichero de exclusiones. Es típico que, por sistema operativo, por el IDE o por el sistema de gestión del ciclo de vida del proyecto, se generen en el directorio de nuestra aplicación ficheros que no queremos que formen parte del repositorio. Para ello podemos tener una configuración local por proyecto o global para todos los proyectos.

Nostoros vamos a configurar un fichero de exclusiones particular por proyecto, de modo que cuando alguien se incorpore al equipo no tenga que configurar nada al respecto y no haya posibilidad de subidas al repositorio de ficheros de configuración locales.

Para ello crearemos un fichero .gitignore en el directorio ráiz del proyecto con el siguiente contenido:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | # Ignores  \.DS\_Store  \.classpath  \.project  \.settings  \.metadata/  Servers/  target/ |

Son las exclusiones típicas si trabajamos en Mac, con Eclipse y Maven.

El resultado de todas estas configuraciones debería ser un fichero **.gitconfig** en nuestro directorio de usuario con el siguiente contenido:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | [user]  name = Jose Manuel Sánchez Suárez  email = jmsanchez@autentia.com  [color]  status = auto  branch = auto  diff = auto  interactive = auto |

4. Preparar el entorno de GitHub.

GitHub usa una clave ssh para establecer una conexión segura entre nosotros y el servidor de git. Si no disponemos de una debemos generarla y añadir la misma a nuestro perfil de GitHub.

Para generar una nueva clave, desde el terminal debemos ejecutar los siguientes comandos:

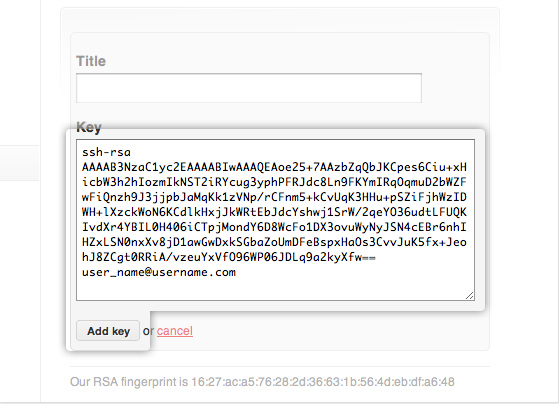
ssh-keygen -t rsa -C "your\_email@yourcompany.com"

Nos solicitará una ubicación para la clave, por defecto: /Users/your\_user\_directory/.ssh/id\_rsa y una “frase de paso” como contraseña.

Al finalizar deberíamos tener algo como esto:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | 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.  The key fingerprint is:  01:0f:f4:3b:ca:85:d6:17:a1:7d:f0:68:9d:f0:a2:db your\_email@yourcompany.com  The key's randomart image is:  +--[ RSA 2048]----+  |     .+   +      |  |       = o O .   |  |        = \* \*    |  |       o = +     |  |      o S .      |  |     o o =       |  |      o . E      |  |                 |  |                 |  +-----------------+ |

Ahora dentro del sitio de GitHub en “Account Settings” > “SSH Public Keys” > “Add another public key”, denemos añadir el contenido del fichero id\_rsa.pub generado en /Users/your\_user\_directory/.ssh/



Para comprobar la configuración debemos ejecutar lo siguiente:

|  |  |
| --- | --- |
| 1 | ssh git@github.com |

La salida debería ser algo como esto:

|  |  |
| --- | --- |
| 1  2  3 | The authenticity of host 'github.com (207.97.227.239)' can't be established.  RSA key fingerprint is 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48.  Are you sure you want to continue connecting (yes/no)? |

Escribimos yes y pulsando intro tendremos la siguiente salida:

|  |  |
| --- | --- |
| 1  2  3 | PTY allocation request failed on channel 0  Hi username! You've successfully authenticated, but GitHub does not provide shell access.  Connection to github.com closed. |

5. Subir el proyecto.

Una vez tenemos el entorno montado ya podemos realizar la subida del pryecto para compartilo al resto del equipo.

Al tratarse de un proyecto nuevo lo primero es posicionarnos a nivel de la carpeta padre del proyecto (allí donde se encuentra, por ejemplo, el directorio .metadata) y ejecutar:

|  |  |
| --- | --- |
| 1 | git init |

Con ello crearemos el repositorio de git, que no es más que el directorio .git en la misma ubicación (si tuvieramos que volver a crearlo basta con borrarlo).

A continuación, añadimos todos los fuentes al repositorio, aquí entrarán en juego las exclusiones, con el siguiente comando:

|  |  |
| --- | --- |
| 1 | git add \* |

Para revisar lo añadido al repositorio podemos ejecutar el siguiente comando:

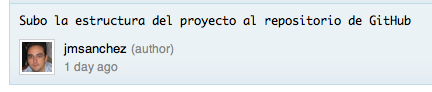
|  |  |
| --- | --- |
| 1 | git status |

Todo lo que devuelva el listado será lo que se suba con el proyecto.

Solo queda hacer un commit, asignar el repositorio remoto y hacer un push

|  |  |
| --- | --- |
| 1  2  3 | git commit -m 'Subo la estructura del proyecto al repositorio de GitHub'  git remote add origin git@github.com:my\_company/my\_project.git  git push -u origin master |

Ahora en el site del proyecto en GitHub tendremos el push:



Con ello ya tenemos el proyecto compartido.

6. Referencias.

* [Git y cómo trabajar con un repositorio de código distribuido](https://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=git)
* [http://help.GitHub.com/git-ignore/](http://help.github.com/git-ignore/)
* <http://help.github.com/mac-set-up-git/>