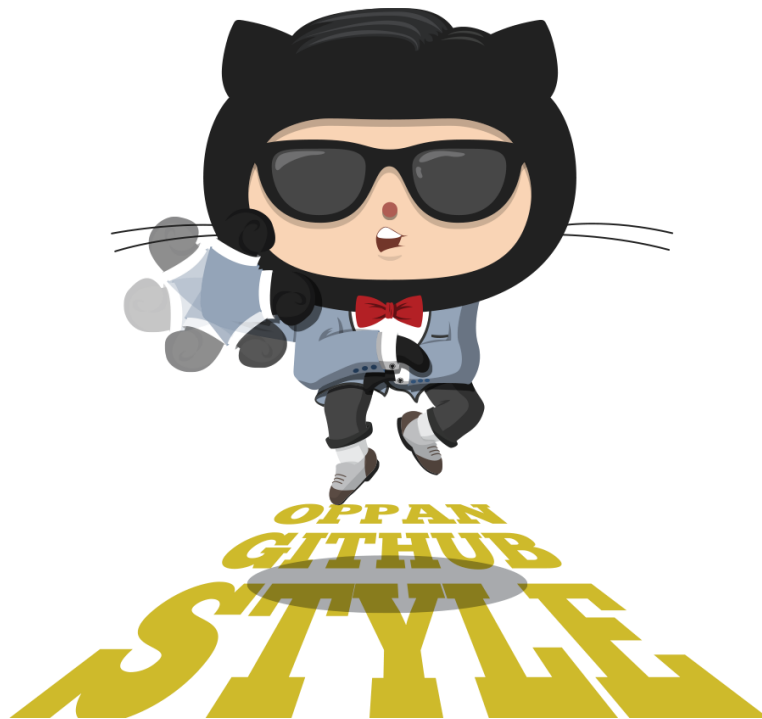
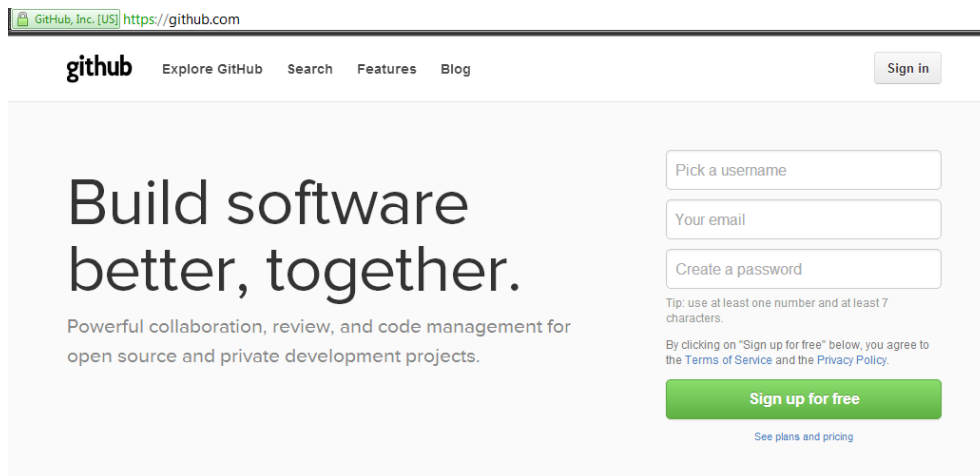


Universidad Gerardo Barrios
Facultad de Ciencia Y Tecnología
Ingeniería de Software I
San Miguel, 2013

First Git & Github



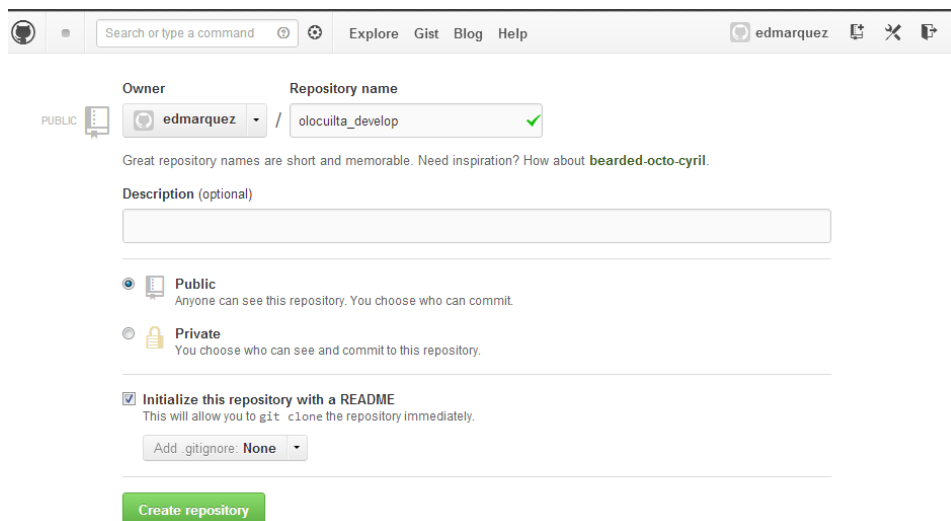
Antes de comenzar tienes que crear una cuenta en Github



The screenshot shows the GitHub homepage with the sign-up form. The form includes fields for 'Pick a username', 'Your email', and 'Create a password'. Below the password field is a tip: 'Tip: use at least one number and at least 7 characters.' There is also a note about agreeing to the Terms of Service and Privacy Policy by clicking 'Sign up for free'. A green 'Sign up for free' button is at the bottom right of the form area.

Tienes que descargar software de Git, puedes hacerlo desde la página <http://git-scm.com/>, lo instalas y comenzamos.

- 1- Se crea un nuevo repositorio en Github, se dejará como público porque la opción de dejarlo privado es pagada.



The screenshot shows the 'Create repository' form in GitHub. The 'Owner' is 'edmarquez' and the 'Repository name' is 'olocuila_develop'. The 'Description (optional)' field is empty. The 'Public' option is selected, with a note: 'Anyone can see this repository. You choose who can commit.' The 'Private' option is also visible. There is a checkbox for 'Initialize this repository with a README' which is checked. Below it, a dropdown menu for 'Add .gitignore' is set to 'None'. A green 'Create repository' button is at the bottom.

- 2- Recordar que el proyecto (carpeta en nuestro Pc) que queremos publicar debe tener el mismo nombre que el repositorio creado en Github

- 3- Con el gitBash que se instaló, buscamos la carpeta del proyecto con `cd Desktop/develop/` digitamos `git init` para iniciar el proyecto.

- 4- Para agregar todos los archivos del proyecto

```
git add .
```

5- Para ver el estatus de lo que tenemos en el proyecto.

```
git status
```

6- Configuramos el correo electrónico y el nombre del usuario, igual que en la cuenta de Github.

```
git config --global user.email "usuario@dominio.com"
```

```
git config --global user.name "Nombre de Usuario"
```

```
git commit -m 'Mensaje de configuración'
```

7- Debemos crear una llave que nos permitirá crear un “túnel” entre nuestro computador y nuestra cuenta de Github permitiéndonos subir los archivos que necesitamos.

```
ssh-keygen -t rsa -C "usuario@dominio.com"
```

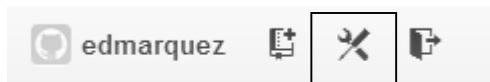
8- Debemos registrar la key en Github, ésta se encuentra en la carpeta

```
C:\Users\usuario\.ssh
```

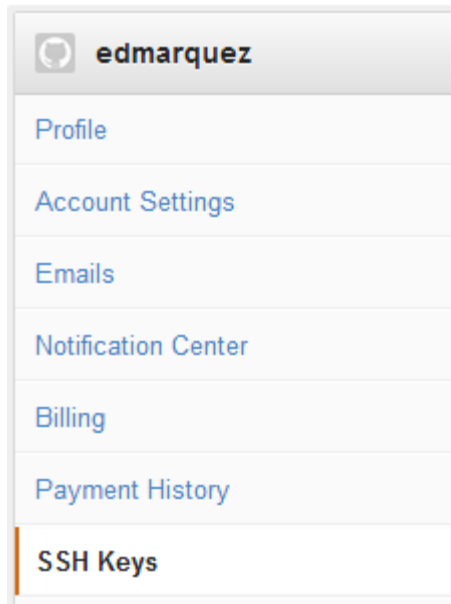
El archivo se llama

```
Id_rsa.pub
```

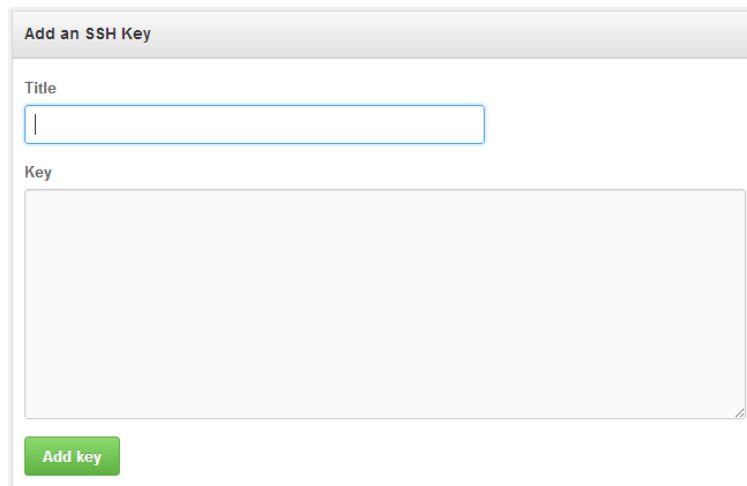
Lo abrimos, copiamos todo lo que está dentro del archivo y nos vamos a nuestra cuenta de Github. Nos dirigimos a la sección de Account settings



Luego a la sección de SSH Keys



Damos clic en “Add SSH key” y las letras que copiamos del archivo `Id_rsa.pub` las pegamos en el “key” y le colocan un título que haga referencia a su llave, cuando la agreguemos nos pedirá autenticación, y colocamos la contraseña de nuestra cuenta Github.



9- Teniendo la key que creamos en nuestro Pc registrada en Github, debemos registrar la dirección de nuestro repositorio de Github que se encuentra en la sección SSH del repositorio que estamos trabajando, como muestra la imagen.



En nuestra consola de Git y lo hacemos con la siguiente línea.

```
git remote add origin git@github.com:edmarquez/develop.git
```

Nos pedirá autenticación y colocaremos la contraseña que colocamos cuando creamos nuestra key

10- La siguiente línea nos permitirá actualizar el repositorio local al más nuevo

```
git pull origin master
```

11-

```
git config --global push.default current
```

12- Con la siguiente línea enviamos los cambios a nuestro repositorio remoto.

```
git push origin master
```