Arquitectura Java

Clase “2”

Agenda:

- Introducción a Git.
- Instalación.
- Creación de public keys
- Git en un IDE
- Comandos útiles
- Ejercicio

GIT

Introducción

“**Git** (pronunciado "guit") es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente” -- Wikkipedia

Git es uno de los sistemas de control de versiones más populares de hoy en día. Gracias a su potencia y versatilidad muchos grandes proyectos de software libre están migrando sus repositorios a Git.

Cada vez más es más importante saber usar Git, tanto a nivel personal como laboral. Parte del éxito de este sistema de repositorios son los repositorios online como GitHub, GitLab o Bitbucket.

Instalación de GIT en windows

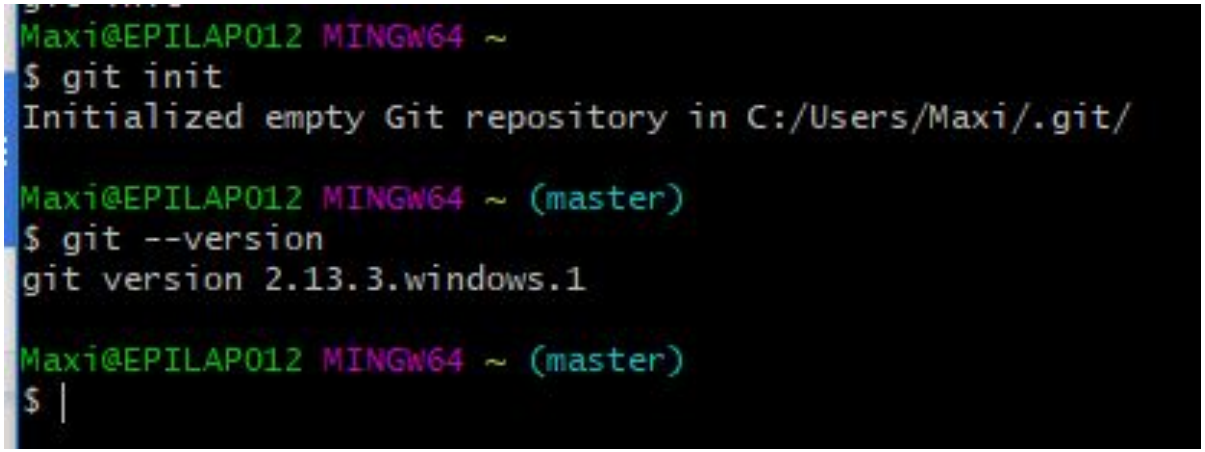
Para trabajar con GIT en windows, tendremos que bajarnos la última versión en https://git-for-windows.github.io/

Una vez instalado, lo primero que tenemos que hacer es configurar nuestro nombre y nuestro email para que Git pueda “firmar” nuestros commits.

$ git config --global user.name "Pepe Sanga"

$ git config --global user.email "pepe@educacionit.com"

Luego creamos un repositorio GIT

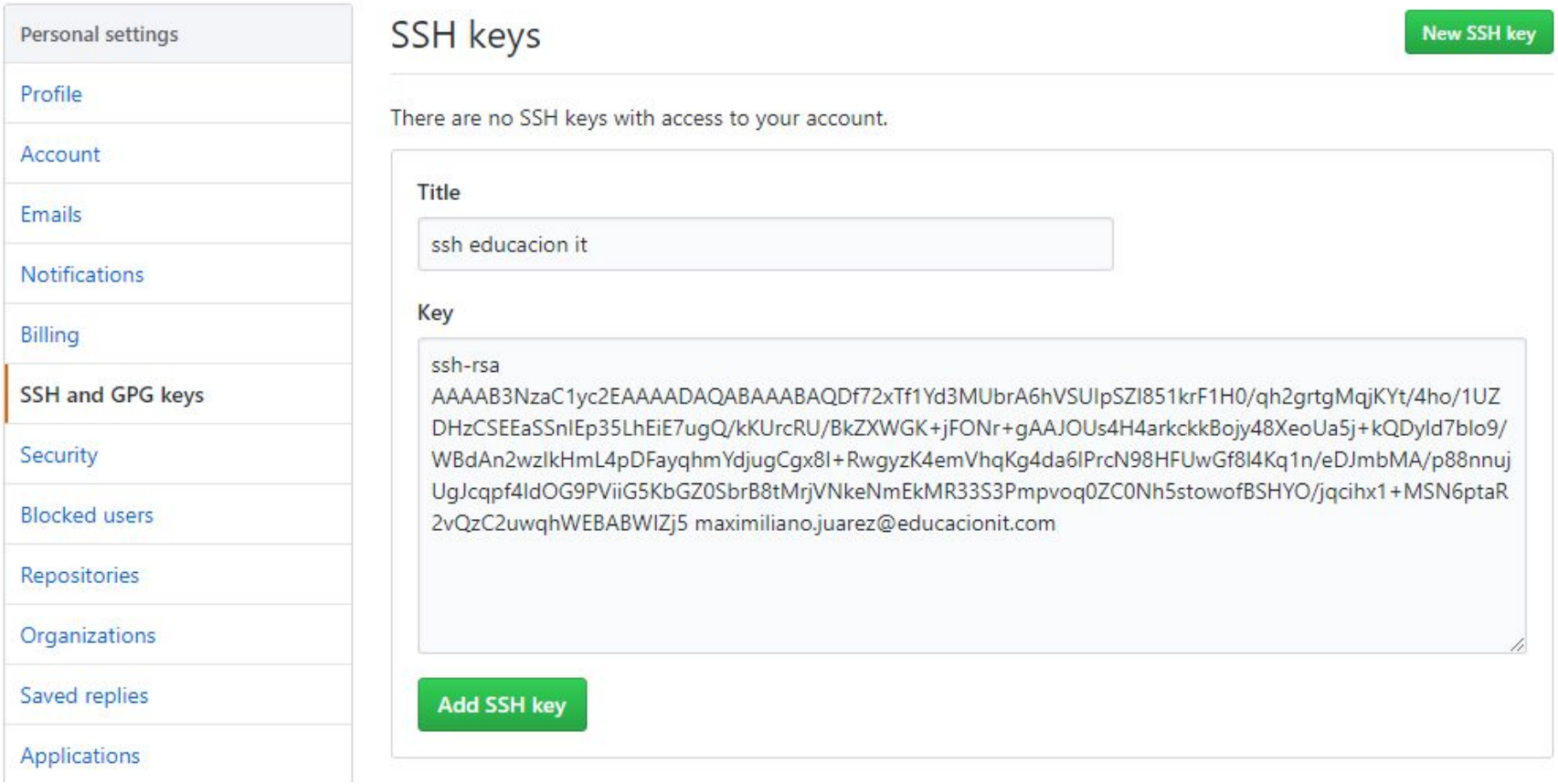


Esto quiere decir que se ha creado el directorio `.git/` donde se guardará toda la información de control.

Si vamos a trabajar con Github, nos va a solicitar una clave para mantener una conexión segura con el servidor:

```
Maxi@EPILAPO12 MINGW64 ~ (master)
$ git config --global user.name "Maxi Juarez"

Maxi@EPILAPO12 MINGW64 ~ (master)
$ git config --global user.email "maximiliano.juarez@educacionit.com"

Maxi@EPILAPO12 MINGW64 ~ (master)
$ 1
ssh-keygen -t rsa -C "your_email@yourcompany.com"

Maxi@EPILAPO12 MINGW64 ~ (master)
$ 1
bash: 1: command not found

Maxi@EPILAPO12 MINGW64 ~ (master)
$ ssh-keygen -t rsa -C "maximiliano.juarez@educacionit.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Maxi/.ssh/id_rsa): log
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in log.
Your public key has been saved in log.pub.
The key fingerprint is:
SHA256:yaSDNaipo4Jw7gk5KricvMLAqIcUhZdlxp++Qtqlrbo maximiliano.juarez@educacioni
t.com
The key's randomart image is:
+---[RSA 2048]----+
|  . ++           |
| . +oo           |
|  o . + o        |
| . o o B .       |
|o + . + S        |
|+=.  . +         |
|#+  + + .        |
|&=+o + o         |
|BO+Eo.o          |
+----[SHA256]-----+
```

En el sitio GitHub ---> “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/



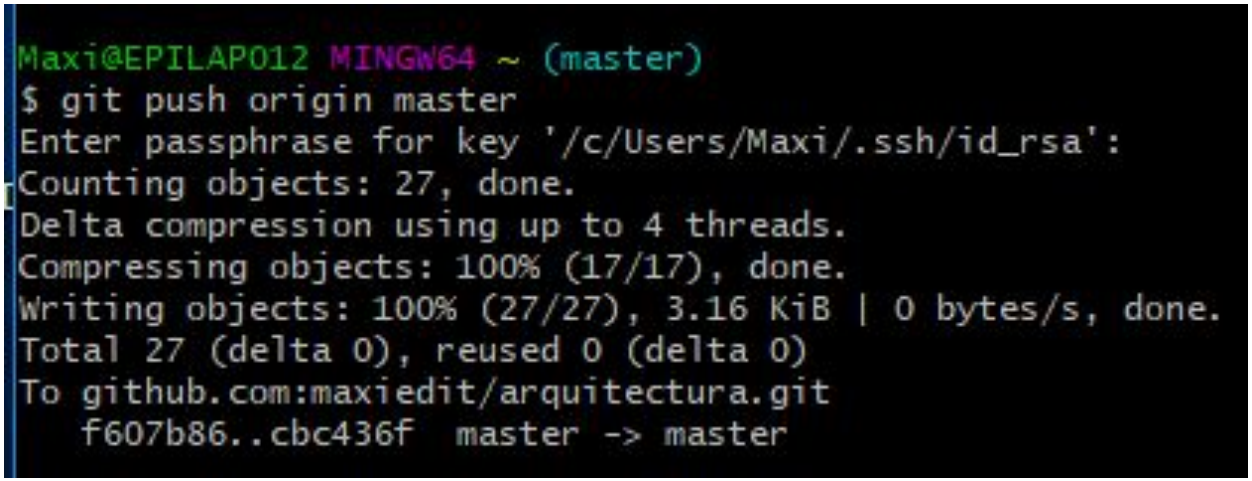
En windows muchas veces no se crea la carpeta .ssh. Habrá que verificar si la carpeta ssh fue creada y que el archivo id_rsa.pub esta dentro.

Subiendo un proyecto a GitHub

- Crear un proyecto Java.
- Ejecutar git add *, para agregar todos los archivos del proyecto
- Ejecutar git status para verificar lo que se enviará al repositorio
- Deberíamos ver algo así:

```
Maxi@EPILAP012 MINGW64 ~/workspace2 (master)
$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   arquitectura/.classpath
        new file:   arquitectura/.project
        new file:   arquitectura/.settings/org.eclipse.jdt.core.prefs
        new file:   arquitectura/.settings/org.eclipse.m2e.core.prefs
        new file:   arquitectura/pom.xml
        new file:   arquitectura/src/main/java/arquitectura/Main.java
        new file:   arquitectura/target/classes/META-INF/MANIFEST.MF
        new file:   arquitectura/target/classes/META-INF/maven/arquitectura/arquitectura/pom.properties
        new file:   arquitectura/target/classes/META-INF/maven/arquitectura/arquitectura/pom.xml
        new file:   arquitectura/target/classes/arquitectura/Main.class
```

- Ejecutar git commit -m 'Subo el proyecto a GitHub'.
- git remote add origin git@github.com:maxiedit/arquitectura.git
- git pull origin master
- git push -u origin master

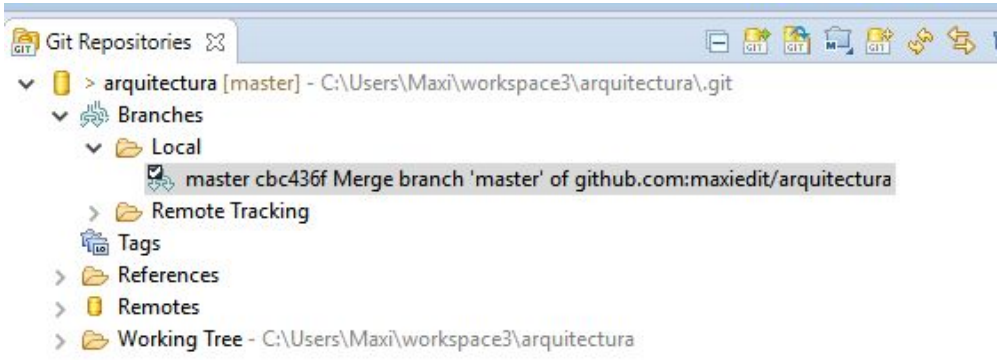


Clonar un proyecto de GitHub

Para clonar un repositorio online se utiliza el comando git clone:

```
Maxi@EPILAPO12 MINGW64 ~/workspace3 (master)
$ git clone https://github.com/maxiedit/arquitectura.git
Cloning into 'arquitectura'...
remote: Counting objects: 30, done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 30 (delta 0), reused 27 (delta 0), pack-reused 0
Unpacking objects: 100% (30/30), done.
```

Luego se lo importa como proyecto maven.



Comandos útiles de GIT

- Push a la rama actual branch: git push origin $mi_branc
- Volver al commit anterior sin guardar los cambios: git reset --HARD $SHA1
- Ver las ramas remotas: git remote show origin
- Traer una rama nueva sin mergearla: git fetch origin
- Traer una rama nueva mergeandola: git checkout -t origin/$branch_name
- Ver todas las ramas locales: git branch -a
- Crear una rama desde una remota: git checkout -b $branch remotes/origin/$branch
- Crear una nueva rama desde la HEAD: git branch $branch
- Crear una nueva rama desde la actual: git checkout -b $new_branch $other
- Borrar la rama local: git branch -d $branch
- Borrar la rama remota: git push origin :$branch
- Cambiar nombre de la rama: git branch -m $lastname $newname

Más comandos

- Dehacer el ultimo commit sin pushearlo:git reset --soft HEAD~1
- Deshacer el ultimo commit despues de haber hecho push:
- git revert HEAD
- Ver los commit no pusheados todavia: git log origin/master..master
- Deshacer el ultimo commit: git reset --soft HEAD^
- Mezcla los ultimos 10 commits en uno solo: git rebase -i HEAD~10