

**Autor:** David Rodríguez-Pastrana Parareda

**Profesor:** Dagoberto Castellanos Nieves

**Asignatura:** Laboratorio de desarrollo y herramientas

**Fecha:** 15 de Noviembre de 2013

# Análisis GitHub

## Proyecto Jenes

### a. Introducción

GitHub es un hosting para Git, gratuito para proyectos open-source. Git es un sistema de control de versiones distribuido, de código abierto y gratuito que permite controlar todo proyecto sin necesidad de estar conectado a la Red.

En este informe vamos a ver cómo configurar el entorno en local y en el servidor GitHub para comprobar la utilidad y facilidad que nos permite la herramienta de trabajar, distribuir y compartir los ficheros de nuestro proyecto. En nuestro caso, vamos a realizar la prueba sobre el proyecto open-source 'Jenes' que se analizó en la práctica virtual I y que vamos a compartir remotamente desde el servidor de Github de forma pública.

GitHub permite:

- Sincronizar automáticamente todos los ficheros del proyecto y notificar si se ha producido algún cambio.
- Clonar repositorios alojados en GitHub para modificar ficheros del proyecto desde nuestro equipo para luego publicarlos en GitHub. Sincronizar automáticamente todos los ficheros del proyecto y notificar si se ha producido algún cambio.
- Gestionar cada repositorio con sus respectivos ficheros desde la propia aplicación: permite consultar el historial de todas las modificaciones realizadas en el código, de los Commits realizados, revertir los cambios realizados gracias al servidor, comparar modificaciones nuevas con los ficheros anteriores del servidor, etc.

## b. Herramienta utilizada

Git nos permite desde la consola crear nuestro repositorio en local de nuestro proyecto.

En nuestro caso, siendo el proyecto Jenes, realizamos las siguientes operaciones desde la consola de Max OSX:

Nos vamos al directorio de nuestro proyecto:

```
MacBook-Pro-de-David:Desktop David$ cd jenes/
```

```
MacBook-Pro-de-David:jenes David$ ls
```

```
AlgorithmEventListener.java      performance
AlgorithmException.java          population
GenerationEventListener.java     stage
GeneticAlgorithm.java            statistics
JenesException.java              tutorials
algorithms                       utils
chromosome
```

Añadimos el link para establecer la conexión remota con GitHub:

```
MacBook-Pro-de-David:jenes David$ git remote add origin https://
github.com/alu0100508031/jenes.git
```

```
MacBook-Pro-de-David:jenes David$ git init
```

```
Reinitialized existing Git repository in /Users/David/Desktop/
jenes/.git/
```

```
MacBook-Pro-de-David:jenes David$ git add *
```

```
MacBook-Pro-de-David:jenes David$ git commit -m "primer commit"
```

```
[master (root-commit) 35aa520] primer commit
```

```
131 files changed, 24472 insertions(+)
```

Actualizamos los ficheros que pudieran ser modificados en GitHub:

```
MacBook-Pro-de-David:jenes David$ git push -u origin master  
  
Branch master set up to track remote branch master from origin.  
  
Everything up-to-date
```

Subimos los fichero de nuestro proyecto al servidor GitHub:

```
MacBook-Pro-de-David:git remote add origin https://github.com/  
alu0100508031/jenes.git  
  
MacBook-Pro-de-David:jenes cd .git  
  
MacBook-Pro-de-David:.git push -u origin master
```

Podemos ver el historial de las modificaciones realizadas (Commits) en el proyecto desde el fichero 'COMMIT\_EDITMSG':

```
MacBook-Pro-de-David:.git David$ cat COMMIT_EDITMSG  
  
primer commit
```

En caso de no querer subir ciertos directorios tendremos que crear el fichero '.gitignore' para que no suba ficheros innecesarios como clases compiladas de directorios '/bin, /target, /test'.

En nuestro caso añadimos los directorios '/bin' y '/tutorials' a ignorar:

```
MacBook-Pro-de-David:jenes David$ vi .gitignore  
  
MacBook-Pro-de-David:jenes David$ cat .gitignore  
  
/bin  
  
/tutorials
```

Fichero de configuración de nuestro proyecto Jenes, fichero 'config':

```
MacBook-Pro-de-David:.git David$ cat config

[core]

    repositoryformatversion = 0

    filemode = true

    bare = false

    logallrefupdates = true

    ignorecase = true

    precomposeunicode = false

[remote "origin"]

    url = https://github.com/alu0100508031/jenes.git

    fetch = +refs/heads/*:refs/remotes/origin/*

[branch "master"]

    remote = origin

    merge = refs/heads/master
```

### c. Protocolos utilizados en GitHub

GitHub permite usar cuatro tipos de protocolo que son:

**1. Local:** permite copiar otros proyectos en disco.

ej.: `git clone /Desktop/David/jenes.git`

**2. HTTP/HTTPS:** permite ser usado por el navegador web, simplicidad.

ej.: `git clone https://github.com/alu0100508031/jenes.git`

**3. SSH:** permite clonar, subir, descargar desde git URL.

`git clone ssh://user@server/project.git`

**4. GIT:** permite clonar y descargar desde puerto dedicado (9418).

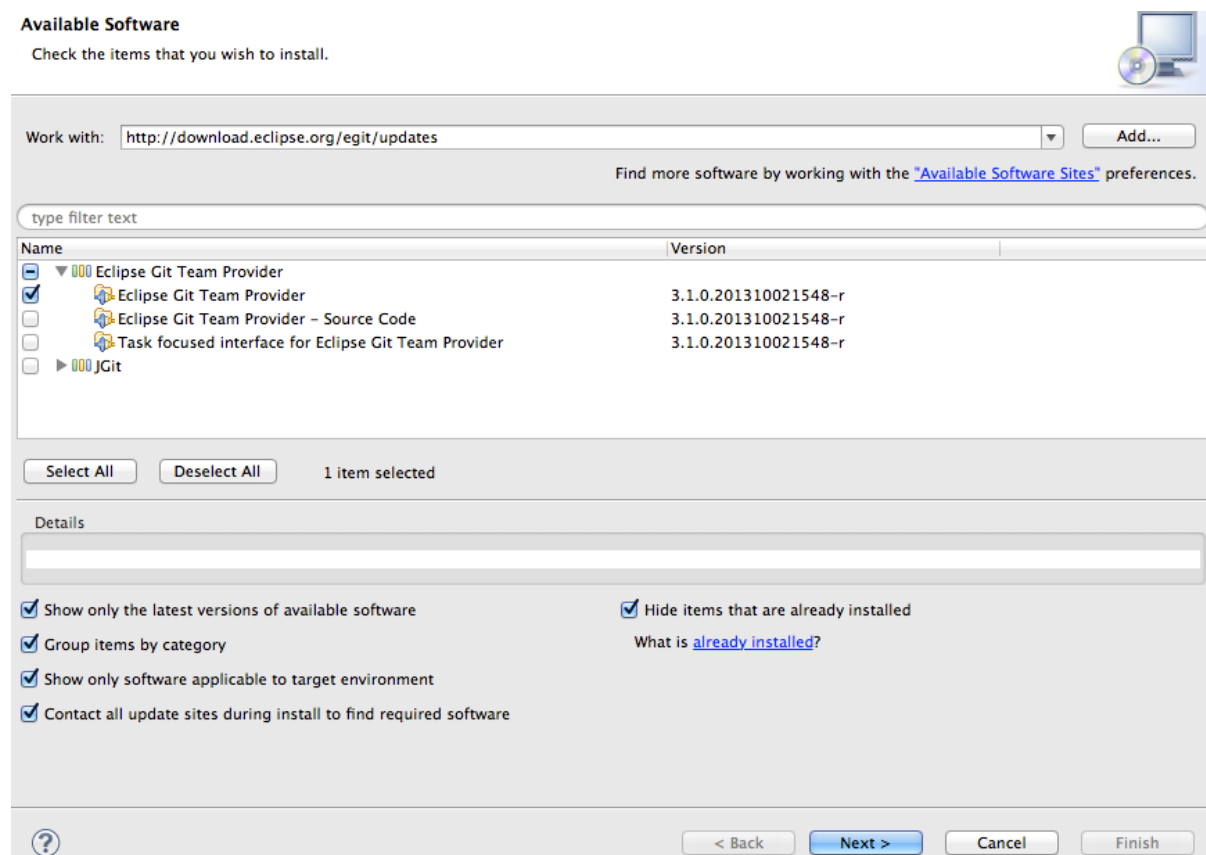
En nuestro caso hemos usado el protocolo HTTPS con el enlace URL que nos proporciona GitHub para nuestro proyecto.

Enlace: <https://github.com/alu0100508031/jenes.git>

## d. Revisión Plugin EGit de Eclipse en proyecto Jenes

Como herramienta cliente se ha utilizado el Plugin 'EGit' para Eclipse que se ha instalado desde el menu 'Help > Install New Software' y en 'Work with' click en 'Add' añadimos el enlace:

<http://download.eclipse.org/egit/updates>



Seleccionamos sólo el pack 'Eclipse Git Team Provider' y damos click en 'Next' hasta finalizar.

Creamos una cuenta GitHub en 'http://www.github.com', en nuestro caso para el proyecto 'Jenes' ha sido con usuario 'alu0100508031', nos dirigimos al menu de nuestra cuenta github 'Create a new repo' que va a ser 'jenes' y seleccionamos la opción predefinida 'Public' para que

cualquier persona pueda ver nuestro proyecto (podremos seleccionar quién puede realizar Commits para la modificación de los ficheros). La opción 'Private' servirá en el caso de que queramos seleccionar quién puede ver y realizar Commits sobre el proyecto.

The screenshot shows the GitHub 'Create a new repository' form. At the top, there's a search bar and navigation links (Explore, Gist, Blog, Help). The user 'alu0100508031' is logged in. The form has a 'PUBLIC' toggle, an 'Owner' dropdown set to 'alu0100508031', and a 'Repository name' field containing 'jenes' with a green checkmark. Below this is a 'Description (optional)' text area. There are two radio buttons for visibility: 'Public' (selected) and 'Private'. Under 'Public', it says 'Anyone can see this repository. You choose who can commit.' Under 'Private', it says 'You choose who can see and commit to this repository.' There's a checkbox for 'Initialize this repository with a README' with the text 'This will allow you to git clone the repository immediately.' Below this are two dropdowns: 'Add .gitignore: None' and 'Add a license: None'. At the bottom is a green 'Create repository' button.

Tras hacer click en 'Create repository', se nos ofrece el enlace Git junto con las instrucciones para instalar Git por consola el enlace de nuestro proyecto con el servidor Github como en la siguiente imagen:

The screenshot shows the GitHub repository page for 'jenes' by user 'alu0100508031'. The repository is 'PUBLIC'. At the top right, there are buttons for 'Unwatch', 'Star' (1), and 'Fork' (0). Below the repository name, there's a 'Quick setup' section with a yellow background. It has a green button 'Set up in Desktop' and a text input field for the repository URL 'https://github.com/alu0100508031/jenes.git'. Below this, it says 'We recommend every repository include a README, LICENSE, and .gitignore.' The main section is titled 'Create a new repository on the command line' and contains a code block with the following commands:

```
touch README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/alu0100508031/jenes.git
git push -u origin master
```

Below this is another section titled 'Push an existing repository from the command line' with a code block containing the following commands:


```
git remote add origin https://github.com/alu0100508031/jenes.git
git push -u origin master
```

Click derecho sobre el proyecto y seleccionamos Git desde

‘Team > Share Project’, Next y finalizar.

Click derecho de nuevo sobre el proyecto y seleccionamos ‘Team > Remote > Fetch from’ e introducimos el enlace en ‘URI’, usuario y contraseña que creamos para nuestra cuenta de Github:

<https://github.com/alu0100508031/jenes.git>

**Source Git Repository** 

Enter the location of the source repository.

---

**Location**

URI:

Host:

Repository path:

---

**Connection**

Protocol:

Port:

---

**Authentication**

User:

Password:

Store in Secure Store ☐

Click en ‘Next’ y luego a ‘Force Update All Specs’.

Desde el menu 'Team' del proyecto Jenes podremos realizar Commits de los ficheros que modifiquemos mediante 'Team > Push', con 'Team > Update to Head' vamos a permitir actualizar los ficheros desde nuestro servidor de GitHub.

En caso de haber algún conflicto porque dos personas han modificado el mismo fichero, Eclipse nos lo indicará a través del símbolo 'U' encima del fichero pudiendo reeditar ambos códigos. También nos mantiene informado del historial de los ficheros modificados pudiendo retroceder en cada fichero del proyecto mediante 'Team > Revert'.

A continuación se muestra el directorio GitHub del proyecto Jenes:

Enlace público: <https://github.com/alu0100508031/jenes>

The screenshot shows the GitHub repository page for 'jenes' by user 'alu0100508031'. The repository has 1 commit, 1 branch, 0 releases, and 1 contributor. The main content area shows a list of files and folders, all of which were committed '3 minutes ago' by 'primer commit'. The files listed are:

- algorithms
- chromosome
- performance
- population
- stage
- statistics
- tutorials
- utils
- AlgorithmEventListener.java
- AlgorithmException.java
- GenerationEventListener.java
- GeneticAlgorithm.java

The right sidebar contains links to 'Code', 'Issues', 'Pull Requests', 'Wiki', 'Pulse', 'Graphs', 'Network', and 'Settings'. At the bottom of the sidebar, there is a section for cloning the repository with the HTTPS clone URL: <https://github.com/alu0100508031/jenes>. Below this, there are buttons for 'Clone in Desktop' and 'Download ZIP'.



## e. Recursos

Directorio Git proyecto Jenes: <https://github.com/alu0100508031/jenes>

Plugin EGit Eclipse: <http://www.vogella.com/articles/EclipseGit/article.html>

Definición Git: [http://en.wikipedia.org/wiki/Git\\_\(software\)](http://en.wikipedia.org/wiki/Git_(software))

Comandos Git: <http://gitref.org>

Web Git: <http://git-scm.com>

Web GitHub: <https://github.com>