Cacique
Installation Guide

Indice

1. Introduction

The main objective of this guide is bringing into detail the necessary steps Cacique's tool correct installation. Explain the different existing processes and which the roles within the tool are.

2. ¿What is Cacique?

Cacique is a Web tool for automating tasks developed on Ruby on Rails. For its operation it is necessary that a web server is online constantly.
We recommend and confirm its correct behavior on Firefox (v. 3.6.X).

As an initial requirement a computer is needed, to which we will refer later as Cacique server, with Ubuntu or Debian Operative system (Ubuntu v. 10.10).

3. Installation

Cacique's Installation steps are found below, which must be executed always from Cacique's root directory.
(To know them in detail, read the paragraph Installation Steps)

```
./installer.sh
```

Edit the file: cacique/config/cacique.yml with the corresponding information, and then:

```
ruby config.rb
rake db:create  RAILS_ENV=production
rake db:migrate RAILS_ENV=production
sudo /etc/init.d/apache2 start
```

Ready!

We can navigate Cacique at the following url: http://<server_ip>/

4. Execution

In order to execute a script in Cacique it is necessary for some processes to run on the server (to know in detail each one of them, read the paragraph called

Processes).

4.1 Local Execution

For executing in the user's local computer, run the following processes on the server:

```
/script/starling.rb

/script/workling_client start
```

4.2 Remote Execution

Optional for remote execution:

<path_to_project>/extras/selenium-grid-1.0.8 and execute in command line:

```
ant launch-hub
```

Then, run one or more RC agents in the computers you wish, pointing to the previously running Hub.

Cacique allows you to program executions; in order to do that run the following process on the server:

```
rake jobs:work RAILS_ENV=production
```

4.3 CaciqueOnMyComputer Execution

For the execution of a script on the same computer as the user, it is necessary to run an agent in that computer. For more information, read chapter 6.3.

5. Installation Steps

5.1 Installer

./installer.sh

It is necessary to install a series of packages that are specified in the installer.sh file, located on the root of the project Cacique.

In order to execute the installer, open a console, go to the root directory of the tool and execute the command **./installer.sh**

The different gems and packages that are being installed will start to appear on the console.

MySql Database engine is on the list of packages to install. When this installation starts, a password will be asked for the engine's root user. Remember this password, because you will need it later.

If a package can't be installed, it will be logged in a file called installer.log on the same directory where the installer is found. In that case we must re-execute the installer, or installing the pending package ourselves.

Once the installation is complete, we will have all the software packages needed to run Cacique in the server.
In order to continue, the user profile must be reloaded, executing the command:
source /etc/profile

5.2 Configuration

Configuration File (YML).

Cacique has a configuration file called "cacique.yml" that can be found inside the directory "cacique/config/". This file must be used to configurate everything related to the application's operation:

- Server IP
- Database connection
- Worker's IP
- Selenium Grid IP
- Memcached

etc.

*cacique.yml

```
:server:
  :ip: localhost

:db:
  :development:
    :adapter: mysql
    :encoding: utf8
    :database: cacique_prod
    :pool: 10
    :username: root
    :password: password
    :host: localhost
  :test:
    :adapter: mysql
    :encoding: utf8
    :database: cacique_test
    :pool: 10
    :username: user
    :password: password
    :host: localhost
  :production:
    :adapter: mysql
    :encoding: utf8
    :database: cacique_prod
    :pool: 10
    :username: user
    :password: password
    :host: localhost

```

We must remember that the numeric values (IP type or port) must be written between quotation marks. This will guarantees the correct configuration of the application.

In this step, amongst other configurations, we will generate Cacique's Database where all the information will be stored. MySql Database was installed in the installer executed at step 1, but in case you'd want any other Database, the steps are the same.

Once the configuration file is completed with the necessary data, it will be enough to execute the following command on our Cacique directory:

ruby config.rb

Take into account that every time the file cacique/config/cacique.yml is changed, r**uby config.rb** command must be executed for changes to be updated.

It is also important to configure correctly the time zone in which the Cacique server is found, for this, go into config/environment.rb and modify the default zone, Buenos Aires.

<path_to_project>


```
# Make Time.zone default to the specified zone, and make Active Record store time values
# in the database in UTC, and return them converted to the specified local zone.
# Run "rake -D time" for a list of tasks for finding time zone names. Comment line to use default local
config.time_zone = 'Buenos Aires'

```

5.3 DataBase

Once the configuration is done, open the console, go to Cacique's root directory and execute the following command **rake** db:create **RAILS_ENV=production**

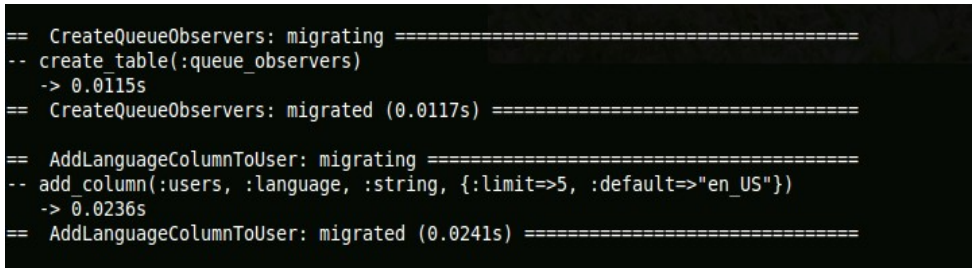
This command is typical of a Rails application and creates the database according to the configuration specified in the file config/database.yml

Once the database is created you should run the migration to generate the tables, to do this you must run the command **rake db:migrate RAILS_ENV=production**

This statement is typical of a Rails application and creates the database according to the configuration specified in the config / database.yml

If **rake** command is not installed, do the following:
$ Sudo apt-get install rake
$ Gem install rake

On the console, executed migrations will be shown as follows:



Once these migrations are completed, we will have a database ready for being used.

5.4 Navigating Cacique

As a last installation step we only have to set up our web server, Mongrel for example, but can be any other, such as Apache with Passenger.

With the implementation of *installer.sh* have the installation of Mongrel, therefore we are able to launch the Cacique:
script / server-e production-p-d <port>
(Remember to use port 80, you must first log in as root with the **su** command)

Mongrel is used for development environments and for those pages that do not have a high level of concurrent users, for companies that have many concurrent recommend the use of passenger, for more information on this module see **Appendix A**.

Now you can browse the url http://<ip_server>:<port>

- You can make in Cacique the following:
- Create a Project
- Create Users
- Assign users to a project
- Create the directory structure to store the scripts
- Scripting
- Create data sets
- Create suites

At this point, is **not yet** possible to execute scripts, for that see the following section of Execution.

6. Execution

6.1 Local Execution

This section gives a brief explanation of the role of each process within the architecture of Cacique.

Apache/Passenger: Cacique's web page server.

MySql: Cacique's database engine for storing.

Memcached: Rails's application Cache.
To run this service, use the command line:
/usr/bin/memcached -u cacique -d

Starling: Process that administrates queues and workers. It receives different executions that are generated on Cacique, queues them up and sends them to different connected workers for their execution.
To run this service, use the command line:
<path_to_project>/script/starling.rb

Workling: A process that runs in the background that is used for executing in Cacique.
To run this service, use the command line:
<path_to_project>/script/workling_client start
Execute this command as many times as workers you'd want to run.

Selenium-Grid: Process used to execute Selenium scripts in different browser configurations.
To run this service, go to **<path_to_project>/extras/selenium-grid-1.0.8** and use the command line:
ant launch-hub
For more information on configuring Selenium-Grid see Appendix B.

Jobs: Rails process that is responsible for implementing programmed suites.
To run this service, go to **<path_to_project>/extras/selenium-grid-1.0.8** and use the command line:
rake jobs:work RAILS_ENV=production

Each process in Cacique is completely detachable from the other, they can all be found on the same server, or distributed in different.

6.2 Remote Execution

Cacique can run its tests on remote computers or on the user's computer.
For remote execution, the following is needed:

1- Having a computer installed with the OS and the browser in which we want to run the test.

2- Run a Selenium agent. For this, it is required to copy the directory that is can be found in /extras/selenium-grid-1.0.8 to the remote machine, open the console, go to the copied directory and execute the following command:

ant -Dport=<port> -Denvironment="<tag>" -DseleniumArgs="-singleWindow -trustAllSSLCertificates " -Dhost=<remote_ip> -DhubURL=http://<server_ip>:4444 launch-remote-control

Replace the values between <> in the command, where:

<port>: Any free port where the agent will stay connected.

<tag>: Configuration name that represents the agent. E.g.: Firefox3 on Windows.

<remote_ip>: ip of the remote computer, it will be used along with the label to identify an agent.

<server_ip>: Ip of the computer where Cacique's server is located.

After following these two steps, Cacique is in conditions of executing a script on a remote computer.
In order to see the active agents, you can go to: http://<server_ip>:4444/console

Note that the "labels" that I use to lift the agents have to be valid labels that Selenium Hub understands. To configure the Hub labels modify the file <path_to_project>/extras/selenium-grid-1.0.8/grid_configuration.yml

For more information on configuring Selenium-Grid see Appendix B.

6.3 CaciqueOnMyComputer Execution

For the execution of a script on the same computer as the user, it is necessary to run an agent in that computer. For that, follow these steps:

1- Download and extract in Cacique's home the .zip file that can be found here Cacique on My Computer.

2- Execute the runFirefox.bat or runIE.bat, depending on the browser you wish to use. A console will open and will show the running agent. Both agents can be running simultaneously.

After executing these two steps, Cacique is in conditions of executing a script on the user's computer.

7. Contacts

For more information, please visit:
http://cacique.mercadolibre.com

Appendix A

Configure Apache

Steps for installation of **Passenger**, which enables the framework for running Ruby on Rails:

1. $ **gem install passenger**

2. $ **passenger-install-apache2-module**

3. Add the end of file **/etc/apache2/apache2.conf** the lines:

```
LoadModule passenger_module     usr/lib/ruby/gems/1.8/gems/passenger
3.0.8/ext/apache2/mod_passenger.so
PassengerRoot /usr/lib/ruby/gems/1.8/gems/passenger-3.0.8
 PassengerRuby /usr/bin/ruby1.8
```

4. Also add to **/etc/apache2/apache2.conf**

```
PassengerMaxPoolSize  6
PassengerPoolIdleTime 150
<VirtualHost *:80>
     ServerName  http://192.xxx.xx.xx o Nombre del server
     DocumentRoot  /home/cacique/public  (ruta recomendada )
     <Directory   /home/cacique/public> (ruta recomendada)
       AllowOverride none
            PassengerMinInstances 1
     </Directory>
  </VirtualHost>
```

5. Then run **$ sudo /etc/init.d/apache2 restart**, to restart

6. Perform a test from a browser by entering the url: http://192.xxx.xx.xx. You should see the Cacique login screen

Apéndice B

Selenium RC

To use Selenium GRID (server), you need:

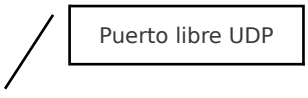
1. Download and install the JDK (Java Development Kit, version 1.5 or higher)
2. Descargar and install the ANT (version 1.7)
3. In the server, a position in the directory and run Selenium Grid:
   ant launch-hub
4. Luego, it is possible to lift the Rcs.

To run the Selenium RC (Remote Agent) on each computer, you need:

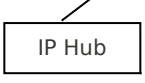
1. Download and install **JDK** (Java Development Kit, version above 1.5) for Windows 7 (in our configuration). Then add the JDK PATH \ bin to the Windows Environment Variables 7. If JDK is not installed, do not run the ANT
2. Download and install the **ANT** (version 1.7) for Windows 7 (if not, you can not lift the RC). Then add the PATH Ant \ bin Environment Variables for Windows 7
3. Create a basic **build.xml** file and create the / src directory that is within the Ant (example: Ant \ src).
4. Create the folder build \ src directory in the Ant (so you can run this version of build.xml). Example Ant \ build \ src
5. Copy the folder Selenium-grid-1.0.8 to the machine that will serve as a remote agent.
   Windows open a DOS console, positioned inside the folder selenium-grid-1.0.8 and run the command:

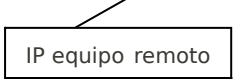
   ant-dport = <port>-Denvironment = "<tag>"-DseleniumArgs = "-singleWindow-trustAllSSLCertificates" - <ip_remota>-DhubURL Dhost = = http:// <ip_server>: 4444 launch-remote-control

   Example:



ant -Dport=5357 -Denvironment="IE8 on Windows" -DseleniumArgs="-singleWindow -trustAllSSLCertificates " -Dhost=192.xxx.xx.xx -DhubURL=http://192.xxx.xx.xx:4444 launch-remote-control





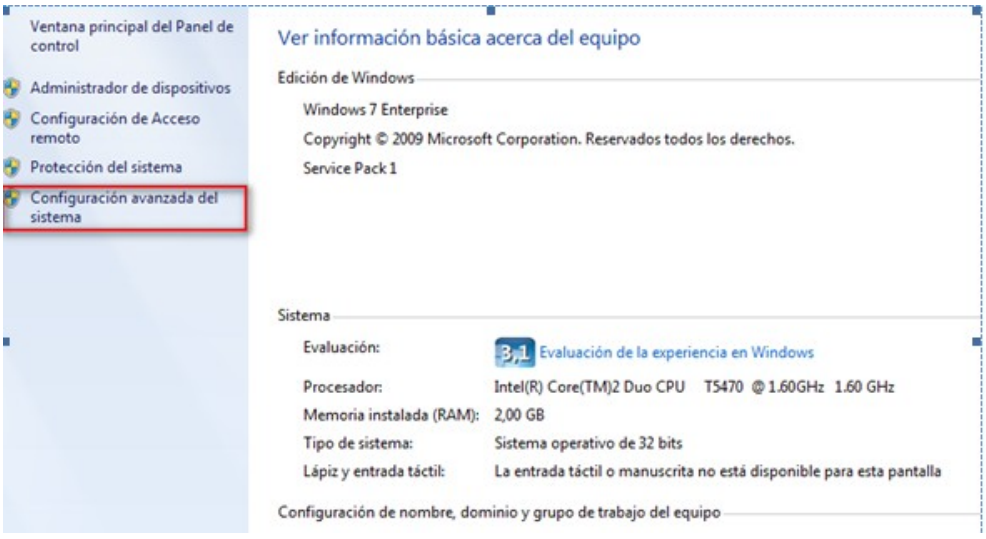
C:\Documents\Docs\selenium-grid-1.0.8>ant -Dport=**3702** -Denvironment="**Chrome on Windows**" -DseleniumArgs="-singleWindow -trustAllSSLCertificates " -Dhost=192.xxx.xx.xx -DhubURL=http://192.xxx.xx.xx:4444 launch-remote-control

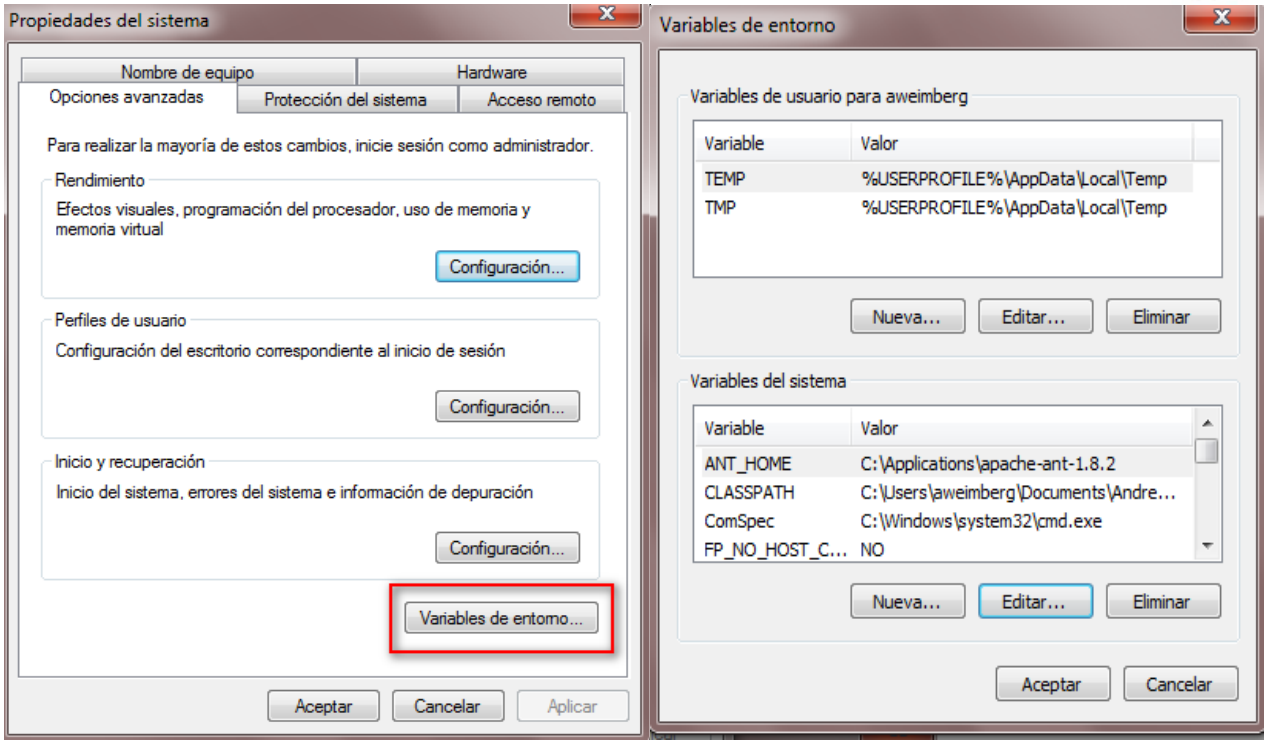
C:\Documents\Docs\selenium-grid-1.0.8>ant -Dport=**4500** -Denvironment="**Firefox3.6 on Windows**" -DseleniumArgs="-singleWindow -trustAllSSLCertificates " -Dhost=192.xxx.xx.xx -DhubURL=http://192.xxx.xx.xx:4444 launch-remote-control

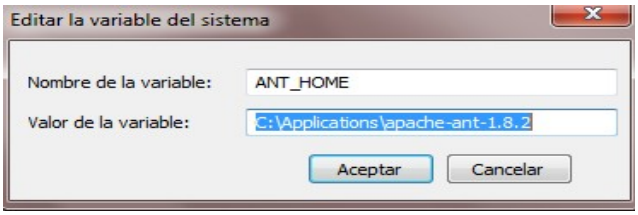
C:\ Documents\ Docs\selenium-grid-1.0.8>ant -Dport=**5355** -Denvironment="**IE8 on Windows**" -DseleniumArgs="-singleWindow -trustAllSSLCertificates " -Dhost=192.xxx.xx.xx -DhubURL=http://192.xxx.xx.xx:4444 launch-remote-control

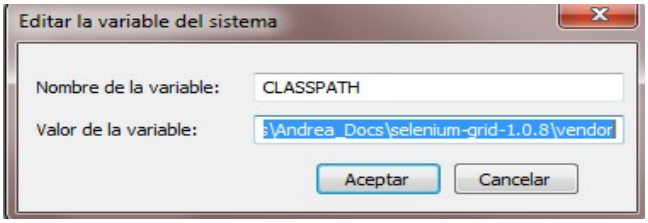
Variables de Entorno:

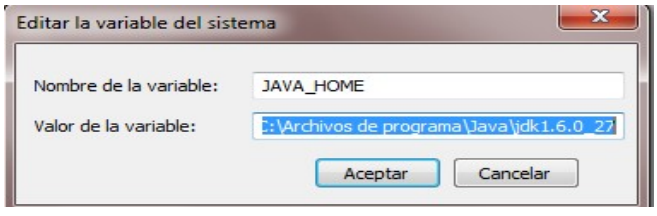
Panel de Control \ Sistema

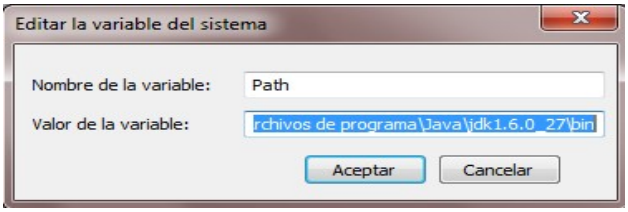



Propiedades del sistema
Nombre de equipo
Hardware
Opciones avanzadas
Protección del sistema
Acceso remoto
Para realizar la mayoría de estos cambios, inicie sesión como administrador.
Rendimiento
Efectos visuales, programación del procesador, uso de memoria y memoria virtual
Configuración...
Perfiles de usuario
Configuración del escritorio correspondiente al inicio de sesión
Configuración...
Inicio y recuperación
Inicio del sistema, errores del sistema e información de depuración
Configuración...
Variables de entorno...
Aceptar
Cancelar
Aplicar
Variables de entorno
Variables de usuario para aweimberg
Variable
Valor
TEMP
%USERPROFILE%\AppData\Local\Temp
TMP
%USERPROFILE%\AppData\Local\Temp
Nueva...
Editar...
Eliminar
Variables del sistema
Variable
Valor
ANT_HOME
C:\Applications\apache-ant-1.8.2
CLASSPATH
C:\Users\aweimberg\Documents\Andre...
ComSpec
C:\Windows\system32\cmd.exe
FP_NO_HOST_C...
NO
Nueva...
Editar...
Eliminar
Aceptar
Cancelar


Editar la variable del sistema
Nombre de la variable:
ANT_HOME
Valor de la variable:
C:\Applications\apache-ant-1.8.2
Aceptar
Cancelar


Editar la variable del sistema
Nombre de la variable:
CLASSPATH
Valor de la variable:
:\Andrea_Docs\selenium-grid-1.0.8\vendor
Aceptar
Cancelar


Editar la variable del sistema
Nombre de la variable:
JAVA_HOME
Valor de la variable:
:\Archivos de programa\Java\jdk1.6.0_27
Aceptar
Cancelar



Archivo Build.xml (crearlo en el directorio del Ant)

```
<?xml version="1.0"?>
<!-- Build file for our first application -->
<project name="Ant test project"  default="build" basedir=".">
<target name="build" >
<javac srcdir="src" destdir="build/src" debug="true"
includes="**/*.java"
/>
</target>
</project>
```

For more Selenium-Grid documentation, you can visit:
http://selenium-grid.seleniumhq.org/