



Cacique **Manual de Instalación**

Cacique - do it once!

Manual de Instalación
<http://cacique.mercadolibre.com>



Indice



1. Introducción

El objetivo de este manual es detallar los pasos necesarios para la correcta instalación de la herramienta Cacique. Explicar los diferentes procesos que existen y cuales son sus roles dentro de la arquitectura de la herramienta.

2. ¿Qué es Cacique?

Cacique es una herramienta Web de automatización de tareas desarrollada sobre el lenguaje Ruby on Rails. Para su funcionamiento es necesario que un servidor web este levantado constantemente.

Recomendamos y aseguramos su correcto funcionamiento en Firefox (v. 3.6.X) .

Como requerimiento inicial se necesita un maquina, a la cual haremos referencia a ella como servidor de Cacique, con el sistema operativo Ubuntu o Debian (Ubuntu v. 10.10).

3. Instalación

Los pasos para realizar la instalación de Cacique se encuentran a continuación, los cuales deben ser ejecutados, siempre desde el directorio raíz de Cacique.

(Para conocer en detalle cada uno de ellos, leer el apartado [Pasos de Instalación](#))

```
./installer.sh
```

Editar el archivo cacique/config/cacique.yml con la información correspondiente, y a continuación:

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

Listo!

Ya podemos navegar Cacique en la url `http://<ip_server>/`



4. Ejecución

Para poder ejecutar un script en Cacique es necesario que en el servidor corran determinados procesos. (Para conocer en detalle cada uno de ellos, leer el apartado Procesos)

4.1 Ejecución local

Para la ejecución en la maquina local de un usuario, levantar en el servidor los siguientes procesos:

```
/script/starling.rb  
  
/script/workling_client start
```

4.2 Ejecución remota

Opcional para ejecución remota:

<ruta_del_proyecto>/extras/selenium-grid-1.0.8 y ejecutar la línea de comando:

```
ant launch-hub
```

Luego, levantar uno o varios agentes RC en las maquina que se desee, apuntando al Hub levantado previamente.

Cacique permite realizar ejecuciones programadas, para ello es necesario levantar el siguiente proceso en el servidor:

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

4.3 Ejecución de Cacique en mi computadora

Para la ejecución de un script en la misma maquina del usuario es necesario levantar un agente en dicha maquina. Se detalla en el punto 6.3.

Cacique - do it once!

Manual de Instalación
<http://cacique.mercadolibre.com>



5. Pasos de instalación

5.1 Instalador

./installer.sh

En el servidor es necesario instalar una serie de paquetes que se encuentran especificados en el archivo `installer.sh` en la raíz del directorio de Cacique.

Para ejecutar el instalador abrir una consola, posicionarse sobre la carpeta raíz de la herramienta y ejecutar el comando **./installer.sh**

En la consola se mostraran las diferentes gemas y paquetes que se estén instalando.

```
~/cacique$ ./installer.sh
[sudo] password for :
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

En la lista de paquetes a instalar se encuentra el motor de base de datos Mysql. Cuando se comience con su instalación se pedirá la password para el usuario root del motor. Recordar esta password, porque se utilizará en un futuro.

Si algún paquete no puede ser instalado se logueara en un archivo llamado `installer.log` en la misma carpeta en que se encuentra el instalador. En ese caso debemos reintentar ejecutando nuevamente el instalador, o instalando nosotros mismo el paquete que haya quedado pendiente.

Una vez completo el instalador tendremos en el servidor todo los paquetes de software necesarios para levantar Cacique.

Para continuar, se debe recargar el perfil de usuario ejecutando el comando **source /etc/profile**



5.2 Configuración

Archivo de Configuración (YML).

Cacique posee un archivo de configuración llamado “cacique.yml” que puede ser encontrado dentro del directorio “cacique/config”.

Este archivo debe ser utilizado para configurar todo lo relativo al funcionamiento de la aplicación:

- IP del Servidor
- Conexión a la Base de Datos
- IP de los Workers
- IP del Selenium Grid
- Memcached
- etc.

```
*cacique.yml ✕
1 :server:
2 ..:ip: localhost
3
4 :db:
5 ..:development:
6 ....:adapter: mysql
7 ....:encoding: utf8
8 ....:database: cacique_prod
9 ....:pool: 10
10 ....:username: root
11 ....:password: password
12 ....:host: localhost
13 ..:test:
14 ....:adapter: mysql
15 ....:encoding: utf8
16 ....:database: cacique_test
17 ....:pool: 10
18 ....:username: user
19 ....:password: password
20 ....:host: localhost
21 ..:production:
22 ....:adapter: mysql
23 ....:encoding: utf8
24 ....:database: cacique_prod
25 ....:pool: 10
26 ....:username: user
27 ....:password: password
28 ....:host: localhost
29
```



Debemos recordar que los valores numéricos (tipo IP o Puerto) deberán ser escritos entre comillas.

Esto nos garantizara la correcta configuración de la aplicación.

En este paso, entre otras configuraciones, vamos a generar la base de datos de Cacique en donde se almacenará toda la información. En el instalador ejecutado en el paso 1 se instalo la base de datos Mysql, pero en caso de querer utilizar cualquier otra base de datos los pasos a seguir serian los mismos.

Una vez completo el archivo de configuración con los datos necesarios, bastara con ejecutar el siguiente comando en nuestra carpeta Cacique:

ruby config.rb

Tener en cuenta que siempre que el archivo `cacique/config/cacique.yml`, se deberá ejecutar `Ruby config.rb` para que impacten los cambios.

Es importante también configurar correctamente la zona horaria en la cual se encuentra el servidor de Cacique, para eso entrar a `config/environment.rb` y modificar la zona por defecto, Buenos Aires.

```
82 | # Make Time.zone default to the specified zone, and make Active Record store time values
83 | # in the database in UTC, and return them converted to the specified local zone.
84 | # Run "rake -D time" for a list of tasks for finding time zone names. Comment line to use default local
85 | config.time_zone = 'Buenos Aires'
86 |
```

<ruta_del_proyecto>

5.3 Base de datos

Una vez realizada la configuración, abrir una consola, posicionarse sobre la carpeta raíz de Cacique y ejecutar el comando **rake [db:create](#)**

RAILS_ENV=production

Este comando es propio de una aplicación de Rails y crea la base de datos según la configuración que se haya especificado en el archivo `config/database.yml`

Si el comando `rake` no se encuentra instalado, realice lo siguiente:

\$ sudo apt-get install rake

\$ gem install rake



Una vez creada la base de datos se deben ejecutar las migraciones que generar las tablas, para hacer esto se debe ejecutar el comando **rake db:migrate RAILS_ENV=production**

En la consola se irán mostrando las migraciones ejecutadas de la siguiente manera:

```
== CreateQueueObservers: migrating =====
-- create_table(:queue_observers)
-> 0.0115s
== CreateQueueObservers: migrated (0.0117s) =====

== AddLanguageColumnToUser: migrating =====
-- add_column(:users, :language, :string, {:limit=>5, :default=>"en_US"})
-> 0.0236s
== AddLanguageColumnToUser: migrated (0.0241s) =====
```

Una vez completadas las migraciones tendremos la base de datos lista para ser utilizada.



5.4 Navegar Cacique

Como último paso para la instalación solo nos queda configurar nuestro web server, en este caso Mongrel, pero puede ser cualquier otro, como por ejemplo Apache con Passenger.

Con la ejecución del instalador `installer.sh` ya contamos con la instalación de Mongrel, por lo tanto nos encontramos en condiciones de levantar el servidor de Cacique de la siguiente manera:

`script/server -e production -p <puerto> -d`

(Recuerde que para utilizar el puerto 80, primero debe loguearse con el usuario root con el comando **su**)

Mongrel se utiliza para entornos de desarrollo y para aquellas páginas que no tengan un alto nivel de concurrencia de usuarios, para empresas que a tengan mucha concurrencia recomendamos el uso de passenger que es el modulo de Ruby para Apache y para Nginx. Para mas información sobre este módulo vea el **Apéndice A**.

Ahora podemos navegar Cacique en la url `http://<ip_server>:<port>`

Hasta aquí se pueden realizar en Cacique las siguientes acciones:

- Crear un proyecto
- Crear usuarios
- Asignar usuarios a un proyecto
- Crear la estructura de directorios de Cacique para almacenar los scripts
- Crear scripts
- Crear sets de datos
- Crear suites

En este punto, **aún no** es posible la ejecución de los scripts, para ello ver el apartado siguiente de Ejecución.



6. Ejecución

6.1 Ejecución local

En esta sección se dará una breve explicación del rol que cumple cada proceso dentro de la arquitectura de Cacique.

Apache/Passenger: Servidor de la página Web de Cacique.

MySQL: Motor de base de datos que utiliza Cacique para almacenar.

Memcached: Cache que utiliza la aplicación de Rails.
Para levantar este servicio utilizar la línea de comando:
/usr/bin/memcached -u cacique -d

Starling: Proceso que administra las colas y los workers. Recibe las diferentes ejecuciones que se generan en Cacique, las encola y las envía a los diferentes workers conectados para su ejecución.
Para levantar este servicio utilizar la línea de comando:
<ruta_del_proyecto>/script/starling.rb

Workling: Proceso que corre en background que se utiliza para la ejecución en Cacique.
Para levantar este servicio utilizar la línea de comando:
<ruta_del_proyecto>/script/workling_client start
Ejecutar este comando tantas veces como cantidad de workers que se quiera levantar.

Selenium-Grid: Proceso utilizado para ejecutar los scripts de Selenium en las diferentes configuraciones de browser.
Para levantar este servicio posicionarse sobre la carpeta
<ruta_del_proyecto>/extras/selenium-grid-1.0.8
y ejecutar la línea de comando:
ant launch-hub
(Para más información sobre esta configuración ver Apéndice B)

jobs: Proceso de Rails que se encarga de ejecutar las suites programadas.
Para levantar este servicio posicionarse sobre la ruta_del_proyecto y ejecutar la línea de comando:
rake jobs:work RAILS_ENV=production

Cada proceso de Cacique es completamente desacoplable, pueden encontrarse todos en el mismo servidor, o distribuidos en distintos.

Cacique - do it once!

Manual de Instalación
<http://cacique.mercadolibre.com>



6.2 Ejecución remota

Cacique ejecuta sus pruebas en maquinas remotas o en la maquina del usuario. Para la ejecución remota es necesario:

- 1- Tener una maquina instalada con el SO y el browser en el cual queremos correr la prueba.
- 2- Levantar un agente de Selenium. Para esto es necesario copiar a la maquina remota la carpeta que se encuentra en /extras/selenium-grid-1.0.8, abrir un consola, posicionarse sobre la carpeta copiada y ejecutar el siguiente comando:

```
ant -Dport=<puerto> -Denvironment="<etiqueta>" -DseleniumArgs="-singleWindow -trustAllSSLCertificates " -Dhost=<ip_remota> -DhubURL=http://<ip_server>:4444 launch-remote-control
```

Reemplazar en el comando los valores entre <>, siendo:

<puerto>: Cualquier puerto libre en donde el agente quedara conectado.

<etiqueta>: Nombre de la configuración que representara el agente. Por ej: Firefox3 on Windows.

<ip_remota>: Ip de la maquina remota, se utilizara junto con la etiqueta para poder identificar un agente.

<ip_server>: Ip de la maquina donde se encuentra el servidor de Cacique.

Luego de realizar estos dos pasos Cacique esta en condiciones de ejecutar un script en la maquina remota.

Para poder ver los agentes activos puedo hacerlo en http://<ip_server>:4444/console

Tener en cuenta que las "etiquetas" que yo utilice para levantar los agentes tienen que ser etiquetas validas que el Selenium Hub entienda. Para configurar las etiquetas del Hub modificar el archivo <ruta_del_proyecto>/extras/selenium-grid-1.0.8/grid_configuration.yml

Para mas información sobre esta configuración ver Apéndice B.

6.3 Ejecución CaciqueOnMyComputer

Para la ejecución de un script en la misma maquina del usuario es necesario levantar un agente en dicha maquina.

Para esto es necesario:

- 1- Descargarse el .zip que se encuentra en el link [Cacique on My Computer](#) en la home de Cacique y extraer el contenido.

- 2- Ejecutar el script runFirefox.bat o runIE.bat, dependiendo el browser a

Cacique - do it once!

Manual de Instalación
<http://cacique.mercadolibre.com>



utilizar. Se abrirá una consola y se mostrara el agente levantado. Se puede tener ambos agentes, IE y FF simultáneamente.

Luego de realizar estos dos pasos Cacique esta en condiciones de ejecutar un script en la maquina de un usuario.



7. Contactos

Para mas información visitar:

<http://cacique.mercadolibre.com>



Apéndice A

Configurar Apache

Pasos a seguir para la instalación de **Passenger**, el cual habilita el framework para ejecutar aplicaciones de Ruby on Rails:

1. \$ **gem install passenger**
2. \$ **passenger-install-apache2-module**
3. Agregar al final de archivo **/etc/apache2/apache2.conf** las líneas:

```
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. Además agregar al **/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. Luego ejecutar **\$ sudo /etc/init.d/apache2 restart**, para reiniciar Apache para que se impacten los cambios.
6. Realizar una prueba desde un navegador, ingresando por la url: <http://192.xxx.xx.xx>. Debe aparecer la pantalla inicial de Cacique login.



Apéndice B

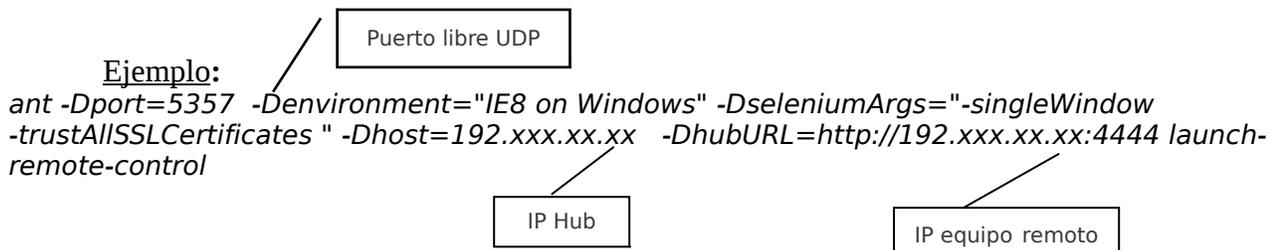
Selenium RC

Para utilizar Selenium GRID (en el servidor), se necesita:

1. Descargar e instalar el **JDK** (Java Development Kit, versión 1.5 o superior)
2. Descargar e instalar el **ANT** (versión 1.7)
3. En el servidor, posicionarse en el directorio de Selenium Grid y ejecutar:
ant launch-hub
4. Luego, ya es posible levantar los Rcs.

Para correr el Selenium RC (Agente Remoto) en cada equipo, se necesita:

1. Descargar e instalar el **JDK** (Java Development Kit, versión arriba de 1.5) para Windows 7 (en nuestra configuración). Luego agregar el PATH del JDK\bin a las Variables de Entorno de Windows 7. Si no se instala el JDK, no corre el ANT
2. Descargar e instalar el **ANT** (versión 1.7) para Windows 7 (si no, no se puede levantar el RC). Luego agregar el PATH del Ant\bin a las Variables de Entorno de Windows 7
3. Crear un archivo **build.xml** básico, y crear la carpeta /src que esté dentro del directorio de Ant (ejemplo: Ant\src).
4. Crear la carpeta build\src dentro del directorio Ant (para que se pueda ejecutar esta versión de build.xml). Ejemplo Ant\build\src
5. Copiar la carpeta Selenium-grid-1.0.8 a la máquina que servirá de agente remoto.
6. Abrir en Windows una consola de DOS, posicionarse dentro de la carpeta selenium-grid-1.0.8 y ejecutar el comando: `ant -Dport=<puerto> -Denvironment="<etiqueta>" -DseleniumArgs="-singleWindow -trustAllSSLCertificates" -Dhost=<ip_remota> -DhubURL=http://<ip_server>:4444 launch-remote-control`



Cacique - do it once!

Manual de Instalación
<http://cacique.mercadolibre.com>



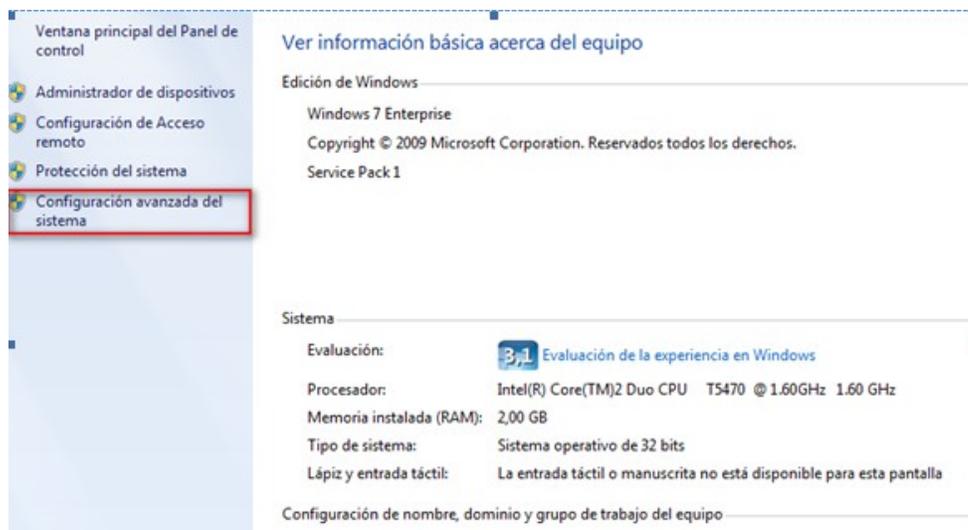
```
C:\Documents\Docs\selenium-grid-1.0.8>ant -Dport=3702 -Denvironment="Chrome on Windows" -DseleniumArgs="-singleWindow -trustAllSSLCertificates " -Dhost=192.xxx.xx.xx -DhubURL=http://192.xxx.xx.xx:4444 launch-remote-control
```

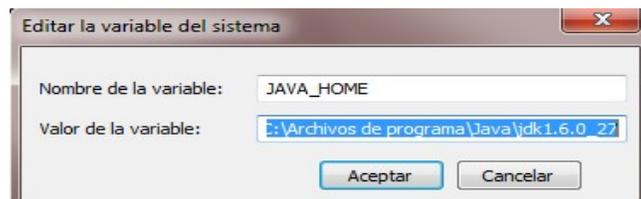
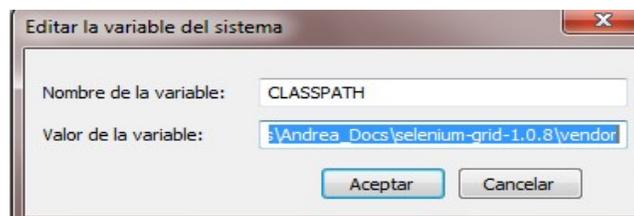
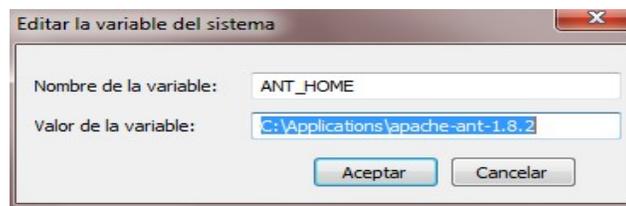
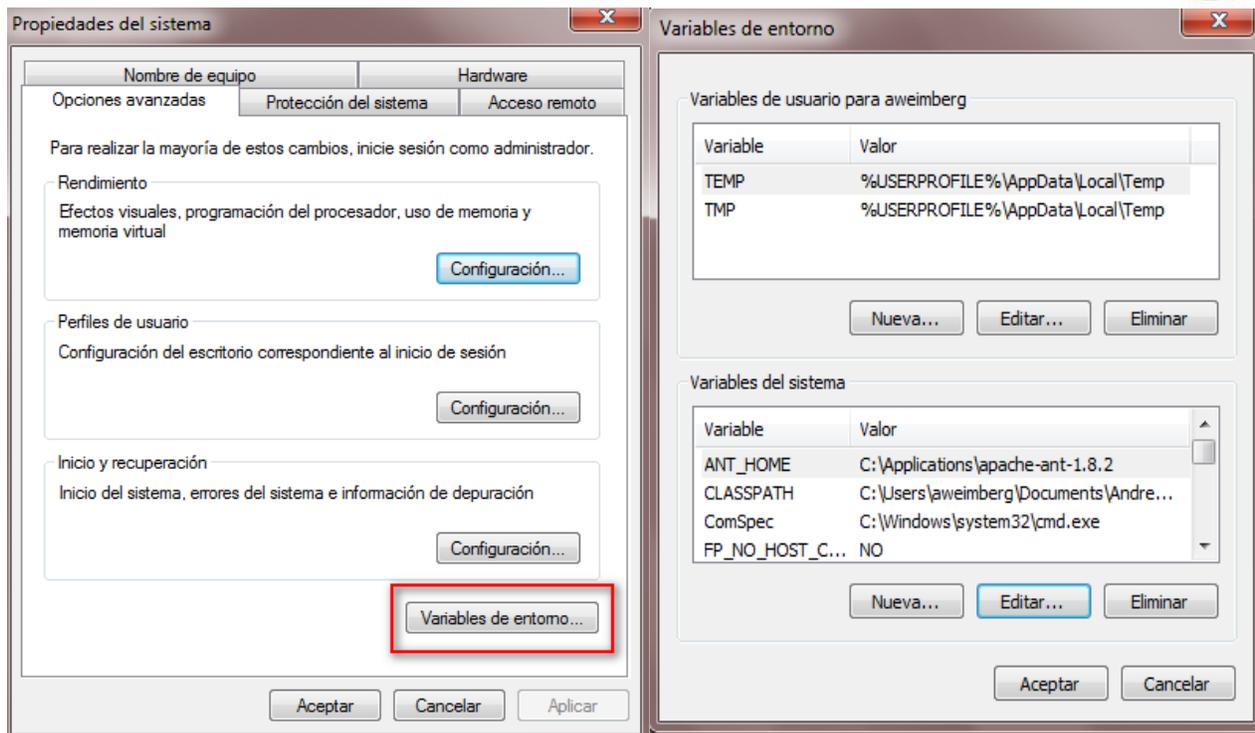
```
C:\Documents\Docs\selenium-grid-1.0.8>ant -Dport=4500 -Denvironment="Firefox3.6 on Windows" -DseleniumArgs="-singleWindow -trustAllSSLCertificates " -Dhost=192.xxx.xx.xx -DhubURL=http://192.xxx.xx.xx:4444 launch-remote-control
```

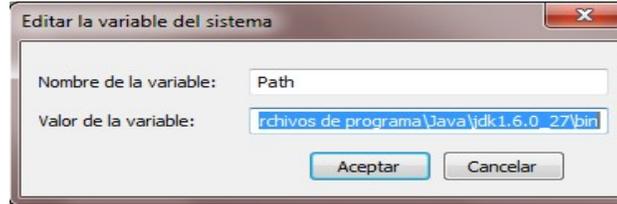
```
C:\Documents\Docs\selenium-grid-1.0.8>ant -Dport=5355 -Denvironment="IE8 on Windows" -DseleniumArgs="-singleWindow -trustAllSSLCertificates " -Dhost=192.xxx.xx.xx -DhubURL=http://192.xxx.xx.xx:4444 launch-remote-control
```

Variables de Entorno:

Panel de Control \ Sistema







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>
```

Para más documentación de Selenium-Grid visitar:
<http://selenium-grid.seleniumhq.org/>