

INSTALACIÓN EN EL SERVIDOR DEL SISTEMA DE COTIZACIONES CORONA

Presentado Por:

GRUPO LUANDOS

ANGIE DANIELA CHISCO CADAVID

JUAN FELIPE GUEVARA GARCÍA

LUKAS MESA BURITICÁ



UNIVERSIDAD DE CALDAS

MANIZALES

DICIEMBRE DE 2017

INSTALACIÓN EN EL SERVIDOR

Índice

1. Instalación de Ruby
2. Instalación Ruby on Rails
3. Configuración apache



FACULTAD DE
INGENIERÍAS

INSTALACIÓN EN SERVIDOR

Para este caso nuestro servidor será apache en Ubuntu 16.04 LTS. Este documento presupone que ya se tiene el servidor apache funcionando por lo que se omite su instalación y configuración.

INSTALACIÓN RUBY:

Instalaremos Ruby manualmente desde la fuente.

Antes de hacer cualquier otra cosa, debemos ejecutar una actualización para asegurarnos de que todos los paquetes que deseamos instalar estén actualizados:

```
sudo apt-get update
```

A continuación, instale algunas bibliotecas y otras dependencias. Esto hará que la instalación sea lo más sencilla posible:

```
sudo apt-get install build-essential libssl-dev libyaml-dev libreadline-dev openssl curl git-core  
zlib1g-dev bison libxml2-dev libxslt1-dev libcurl4-openssl-dev libsqlite3-dev sqlite3
```

Cree una carpeta temporal para los archivos fuente de Ruby:

```
mkdir ~/ruby
```

Muévase a la carpeta recién creada:

```
cd ~/ruby
```



FACULTAD DE
INGENIERÍAS

Descargue el último código fuente estable de Ruby. En el momento de escribir esto, esta es la versión 2.4.2. Puede obtener la última versión del sitio web de Ruby. Si hay una versión más nueva disponible, deberá reemplazarla en el enlace del siguiente comando:

```
wget https://cache.ruby-lang.org/pub/ruby/2.4/ruby-2.4.2.tar.gz
```

Descomprima el archivo descargado:

```
tar -xzf ruby-2.4.2.tar.gz
```

Diríjase la carpeta extraída:

```
cd ruby-2.4.2
```

Ejecute el script de configuración. Esto llevará algo de tiempo, ya que verifica las dependencias y crea un nuevo archivo Makefile, que contendrá los pasos que deben tomarse para compilar el código:

```
./configure
```

Ejecute la utilidad make, que usará el archivo Makefile que se acaba de crear, para compilar el programa ejecutable. Este paso puede demorar un poco más:

```
make
```



**FACULTAD DE
INGENIERÍAS**

Ahora, ejecute el mismo comando con el parámetro de instalación. Esto copiará los binarios compilados a la carpeta /usr/local/bin. Este paso requiere acceso de administrador para escribir en este directorio:

```
sudo make install
```

Ruby ahora debería estar instalado en el sistema. Podemos verificarlo con el siguiente comando, que debe imprimir la versión de Ruby:

```
ruby -v
```

Si su instalación de Ruby fue exitosa, debería ver el siguiente resultado:

```
ruby 2.4.2p198 (2017-09-14 revision 59899) [x86_64-linux]
```

Se recuerda que, si se descargó una versión más actualizada, el resultado varía en número de la versión instalada.

INSTALACIÓN RAILS:

Dado que Rails viene con tantas dependencias, vamos a necesitar instalar un runtime de Javascript como NodeJS. Esto le permite usar Coffeescript y Asset Pipeline en Rails, que combina y minimiza el código javascript para proporcionar un entorno de producción más rápido.

Para instalar NodeJS, vamos a agregarlo usando el repositorio oficial:

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -  
sudo apt-get install -y nodejs
```



FACULTAD DE
INGENIERÍAS

Luego instalamos Rails:

```
gem install rails -v 5.1.4
```

Si está utilizando rbenv, deberá ejecutar el siguiente comando para que los rieles sean ejecutables:

```
rbenv rehash
```

Ahora que ha instalado Rails, puede ejecutar el comando rails -v para asegurarse de tener todo instalado correctamente:

```
rails -v  
# Rails 5.1.4
```

CONFIGURACIÓN APACHE:

Ahora, necesitamos crear un archivo de host virtual para nuestro proyecto. Haremos esto copiando el host virtual Apache predeterminado:

```
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/sistemaCotizacion.conf
```

Abrimos el archivo de configuración:

```
sudo nano /etc/apache2/sites-available/testapp.conf
```

Edítelo o reemplace los contenidos existentes para que su resultado final coincida con el que se muestra a continuación. Los cambios que debe realizar se resaltan en rojo. Recuerde usar su propio nombre de dominio y la ruta correcta de la aplicación Rails:



FACULTAD DE
INGENIERÍAS

```
<VirtualHost *:80>
    ServerName prueba.com
    ServerAlias www.prueba.com
    ServerAdmin webmaster@localhost
    DocumentRoot /home/rails/sistemaCotizacion/public
    RailsEnv development
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    <Directory "/home/rails/sistemaCotizacion/public">
        Options FollowSymLinks
        Require all granted
    </Directory>
</VirtualHost>
```

Si no desea asignar su dominio a esta aplicación, puede omitir las líneas `ServerName` y `ServerAlias`, o usar su dirección IP.

Guarde el archivo tecleando la siguiente combinación (CTRL+X, Y, ENTER).

Por último, deshabilite el sitio predeterminado, habilite el nuevo sitio y reinicie Apache:

```
sudo a2dissite 000-default
sudo a2ensite sistemaCotizacion
sudo service apache2 restart
```

Ahora el sitio web de su aplicación debe ser accesible.



**FACULTAD DE
INGENIERÍAS**