

Instalación de ambiente (componentes obligatorios)

Preparación del sistema operativo

Se asume que para los efectos de esta instalación se posee un servidor con las características requeridas para el funcionamiento de la plataforma (esencialmente hablando, un sistema operativo CentOS 7 limpio, y firewall sin configurar).

Actualizar paquetería del sistema operativo.

```
sudo yum update
```

Instalar paqueterías EPEL y REMI, necesarias para obtener numerosas dependencias del proyecto.

```
sudo yum install epel-release  
sudo rpm -Uvh http://rpms.famillecollet.com/enterprise/remi-release-7.rpm
```

Instalar herramientas de uso general.

```
sudo yum install zip unzip gcc-c++ make screen
```

Instalar paquetes necesarios para la generación de reportes en PDF por parte de la API.

```
sudo yum install libXrender wkhtmltopdf
```

Preparar la carpeta **/var/www** para que pueda ser usada por el usuario actual, y por el servidor HTTP de Apache. Reemplazar USUARIO por el usuario actual de sistema.

```
sudo chown -R USUARIO:apache /var/www/  
sudo chmod -R 755 /var/www/
```

Instalación de bases de datos

Instalar MariaDB desde el repositorio EPEL.

```
sudo yum --enablerepo=remi,epel install mariadb-server
```

Iniciar servicio, y marcarlo para inicio automático en caso de reinicio del servidor.

```
sudo systemctl enable mariadb  
sudo service mariadb start
```

Efectuar configuración inicial de MySQL. Contestar las preguntas acordes a como desee su configuración.

```
/usr/bin/mysql_secure_installation
```

Crear una base de datos y un usuario para la plataforma de Congreso Virtual. Guarde el usuario, contraseña y bases de datos creadas para su uso mas adelante.

A modo de ayuda, se muestra un ejemplo de cómo debe efectuarse la configuración, aunque puede efectuarse de cualquier forma como el instalador desee.

```
sudo mysql -u root -p

mysql> CREATE DATABASE mibd;
mysql> CREATE USER 'miusuario' IDENTIFIED BY 'mipassword';
mysql> GRANT USAGE ON *.* TO 'congreso'@localhost IDENTIFIED BY 'congreso';
mysql> GRANT USAGE ON *.* TO 'congreso'@'%' IDENTIFIED BY 'congreso';
mysql> GRANT ALL privileges ON `congreso`.* TO 'congreso'@localhost;
mysql> FLUSH PRIVILEGES;
```

Instalación del servidor web

Instalar servidor Apache HTTPD Server.

```
sudo yum --enablerepo=remi,epel install httpd -y
```

Iniciar servicio y marcarlo para inicio automático en caso de reinicio de servidor.

```
sudo systemctl enable httpd
sudo service httpd start
```

Configurar firewall para permitir comunicación por puerto 80. Puede repetir el paso si desea habilitar HTTPS (puerto 443) si gusta.

```
sudo firewall-cmd --zone=public --add-port=80/tcp --permanent
sudo firewall-cmd --reload
```

Configurar servidor web apache para servir una página web (backend/api) en **/var/www/congreso-swagger/public**. Si desea puede cambiar la ruta a otra, bajo su responsabilidad. Del mismo modo, puede modificar esta configuración para permitir HTTPS si lo desea.

Agregar el siguiente contenido al fichero **/etc/httpd/conf/httpd.conf** (puede usar editores como Vim o Nano, o bien editores integrados de su cliente FTP para tal fin). Reemplace **dominiocongresoapi.com** por el dominio que utilizará.

```
<VirtualHost *:80>
    ServerName dominiocongresoapi.com
    DocumentRoot /var/www/congreso-swagger/public

    <Directory /var/www/congreso-swagger>
        AllowOverride All
    </Directory>
</VirtualHost>
```

Configurar servidor web de Apache para servir el frontend del sitio en **/var/www/congresovirtual-web-beta/dist**. Si desea puede cambiar la ruta a otra, bajo su responsabilidad. Del mismo modo, puede modificar esta configuración para permitir HTTPS si lo desea.

Agregar el siguiente contenido al fichero **/etc/httpd/conf/httpd.conf** (puede usar editores como Vim o Nano, o bien editores integrados de su cliente FTP para tal fin). Reemplace **dominiocongreso.com** por el dominio que utilizará.

```
<VirtualHost *:80>
    ServerName dominiocongreso.com
    DocumentRoot /var/www/congresovirtual-web-beta/dist
    <Directory /var/www/congresovirtual-web-beta/dist>
        AllowOverride All
    </Directory>
</VirtualHost>
```

Reinicie el servidor web de Apache.

```
sudo service httpd restart
```

Instalación de PHP y Composer

Activar la paquetería REMI para PHP 7.3, previamente instalada.

```
sudo yum install yum-utils
sudo yum-config-manager --enable remi-php73
```

Instalar PHP, junto con las extensiones necesarias para el funcionamiento de congreso.

```
sudo yum --enablerepo=remi,epel install php php-zip php-mysql php-mcrypt php-xml php-mbstring
```

Descargar e instalar Composer desde el sitio oficial (las versiones de paqueterías usualmente presentan incompatibilidades).

```
curl -sS https://getcomposer.org/installer | php
sudo mv composer.phar /usr/bin/composer
sudo chmod +x /usr/bin/composer
```

Configurar PHP para que acepte la subida de ficheros de un mayor tamaño al que viene configurado por defecto. Para ello se debe modificar el fichero **/etc/php.ini** y establecer las siguientes directivas con los siguientes valores recomendados o cómo se estime conveniente.

```
post_max_size = 60M
file_uploads = On
upload_max_filesize = 20M
max_file_uploads = 20
```

Una vez configurado PHP debe reiniciar el servidor web de Apache.

```
sudo service httpd restart
```

Instalación de GIT

Instalar GIT desde el gestor de paquetes.

```
sudo yum install git
```

Instalación de Elasticsearch

Agregar paquetería propia de Elasticsearch al sistema operativo. Se empezará agregando la llave GPG.

```
sudo rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch
```

Luego hay que editar el archivo `/etc/yum.repos.d/elasticsearch.repo` (crearlo si no existe) y colocar el siguiente contenido en él.

```
[elasticsearch]
name=Elasticsearch repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=0
autorefresh=1
type=rpm-md
```

Instalar Elasticsearch desde la paquetería recién añadida.

```
sudo yum install --enablerepo=elasticsearch elasticsearch
```

Iniciar servicio y marcarlo para inicio automático en caso de reinicio de servidor.

```
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable elasticsearch.service
sudo systemctl start elasticsearch.service
```

Instalación de NodeJS

Instalar NodeJS 12.x desde los repositorios oficiales de Nodesource.

```
curl -sL https://rpm.nodesource.com/setup_12.x | sudo -E bash -
sudo yum install nodejs
```

Instalación de Python

Instalar Python junto con las librerías de cabecera del anterior y MariaDB (usado para las dependencias a instalar).

```
sudo yum install python36 mariadb-devel python36-devel
```

Actualizar el gestor de paquetes Pip3.

```
sudo python3.6 -m ensurepip
```

Solución a problemas comunes

La API no conecta a base de datos, pese a colocar los datos de conexión correctamente.

Hay casos en que Laravel y MySQL discrimina entre los servidores 127.0.0.1 y localhost. Favor de especificar el server tal como se configuró en la fase de configuración de bases de datos.

La instalación de dependencias del frontend falla por una dependencia vue-i18n.

Pese a que el equipo de desarrollo ha reportado que el caso ha sido solucionado, en caso de poseer problemas, borrar carpeta **node_modules**, instalar a mano **vue-i18n** via npm install (**npm install vue-i18n**), y luego instalar el resto de las dependencias con (**npm install**).

El compilador de VueJS falla (crashea) al compilar con errores de tipo catastróficos.

Es conocido que el compilador de VueJS tiene problemas de uso de memoria, el cual puede fallar por colapso de memoria libre que NodeJS asigna. Se sugiere volver a intentar, o en su defecto, modificar el parámetro **--max-old-space-size** para optimizar la cantidad de memoria a asignar a costo de mayor tiempo de espera.

Puedo entrar al home del sitio. No obstante, al navegar a cualquier ruta obtengo un error del tipo 404, no importando cual ruta sea.

Verificar que **.htaccess** exista (como indica en los pasos de la instalación de frontend) y que también esté **mod_rewrite** funcionando. Una forma práctica de comprobar que estos funcionen, es editar el archivo **.htaccess** de forma deliberada para que de error de sintaxis y por ende un error 500. Si al editar el archivo no hay error, quiere decir que el servidor web de Apache no está bien configurado (específicamente la configuración **AllowOverride**).