

## 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 más 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 congresovirtual-bd;
mysql> CREATE USER 'congresovirtual-bd' IDENTIFIED BY 'secret';
mysql> GRANT USAGE ON *.* TO 'congresovirtual-bd'@localhost IDENTIFIED BY
'congresovirtual-bd';
mysql> GRANT USAGE ON *.* TO 'congresovirtual-bd'@'%' IDENTIFIED BY
'congresovirtual-bd';
mysql> GRANT ALL privileges ON `congresovirtual-bd`.* TO 'congresovirtual-
bd'@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/congresovirtual-backend/public

    <Directory /var/www/congresovirtual-backend>
        AllowOverride All
    </Directory>
</VirtualHost>
```

Configurar servidor web de Apache para servir el frontend del sitio en **/var/www/congresovirtual-frontend/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-frontend/dist
    <Directory /var/www/congresovirtual-frontend/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
```

## Instalación del proyecto

Descargar el repositorio en el servidor. Si es un entorno de desarrollo puede realizar esto clonando el repositorio GIT, tal como aparece a continuación.

```
git clone https://SERVIDOR_GIT/eii-pucv/congreso-virtual.git
```

**Importante:** Si va a clonar con repositorio GIT, verificar bien que se encuentre en la rama correcta. Consultar con el equipo de desarrollo respecto a cuál rama se debe utilizar.

### Backend/API (congresovirtual-backend)

Copiar la carpeta del repositorio llamada congresovirtual-backend a la ruta /var/www/congresovirtual-backend/.

Asignar permisos indicados a las carpetas que componen el backend recién copiado.

```
sudo chown -R $USER:apache /var/www/  
sudo chmod -R 755 /var/www/congresovirtual-backend  
sudo chmod -R 755 /var/www/congresovirtual-backend/storage  
sudo chcon -R -t httpd_sys_rw_content_t /var/www/congresovirtual-  
backend/storage
```

Acceder a la carpeta del backend y crear un archivo llamado .env.

```
cd /var/www/congresovirtual-backend
```

El contenido del archivo .env contendrá la configuración de la base de datos, entre otros atributos.

```
APP_NAME="Congreso Virtual"  
APP_ENV=staging  
APP_KEY=  
APP_DEBUG=false  
APP_URL=https://url_backend/api  
APP_CLIENT_URL=https://url_frontend/  
APP_ANALYTIC_URL=http://127.0.0.1:8080  
ELASTICSEARCH_ENABLED=true  
ELASTICSEARCH_HOSTS="localhost:9200"  
LOG_CHANNEL=stack  
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=congresovirtual-bd  
DB_USERNAME=congresovirtual-bd  
DB_PASSWORD=secret  
BROADCAST_DRIVER=log  
CACHE_DRIVER=file  
QUEUE_CONNECTION=sync  
SESSION_DRIVER=file  
SESSION_LIFETIME=120
```

```

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379
MAIL_DRIVER=smt
MAIL_HOST=smt.dominio.com
MAIL_PORT=587
MAIL_ENCRYPTION=null
MAIL_USERNAME=direccion@dominio.com
MAIL_PASSWORD=secret
MAIL_FROM_ADDRESS=direccion@dominio.com
MAIL_FROM_NAME="Congreso Virtual"
AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_APP_CLUSTER=mt1
MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
TWITTER_ID=
TWITTER_SECRET=
TWITTER_URL=http://url_backend/api/auth/twitter/callback
FACEBOOK_ID=
FACEBOOK_SECRET=
FACEBOOK_URL=http://url_backend/api/auth/facebook/callback
GOOGLE_ID=
GOOGLE_SECRET=
GOOGLE_URL=
GITHUB_ID=
GITHUB_SECRET=
GITHUB_URL=http://url_backend/api/auth/github/callback
CLAVEUNICA_CLIENT_ID=
CLAVEUNICA_SECRET_ID=
CLAVEUNICA_REDIRECT=http://url_backend/api/auth/claveunica/callback

```

Cambiar los valores en rojo por los siguientes:

APP_URL	URL al backend/api (lo que estamos montando ahora)
APP_CLIENT_URL	URL al frontend
DB_HOST, DB_PORT, DB_DATABASE, DB_USERNAME, DB_PASSWORD	Configuración de la base de datos previamente hecha (como referencia, el puerto MySQL predeterminado es el 3306, y el host de bases de datos es localhost, al menos que se quiera especificar algo distinto).
ELASTICSEARCH_ENABLED, ELASTICSEARCH_HOSTS	Configuración de Elasticsearch, por defecto Elasticsearch elige el puerto 9200 para escuchar las conexiones, pero si se cambia e incluso si se ejecutará en otro servidor, se debe especificar en este archivo de configuración.
MAIL_DRIVER, MAIL_HOST, MAIL_PORT, MAIL_ENCRYPTION, MAIL_USERNAME,	Configuración del correo electrónico desde el cual se enviarán las notificaciones a los usuarios de Congreso Virtual.

MAIL_PASSWORD, MAIL_FROM_ADDRESS, MAIL_FROM_NAME	
TWITTER_ID, TWITTER_SECRET, TWITTER_URL, FACEBOOK_ID, FACEBOOK_SECRET, FACEBOOK_URL, GOOGLE_ID, GOOGLE_SECRET, GOOGLE_URL, GITHUB_ID, GITHUB_SECRET, GITHUB_URL, CLAVEUNICA_CLIENT_ID, CLAVEUNICA_SECRET_ID, CLAVEUNICA_REDIRECT	Configuración de claves identificadoras para el registro de usuarios mediante servicios externos tales como Twitter, Facebook, Google, GitHub y Clave Única (Registro Civil de Chile).

Instalar las dependencias del backend con Composer.

```
composer install
```

Una vez configurado el archivo de variables de entorno .env e instaladas las dependencias, es recomendable ejecutar el siguiente comando para que Laravel actualice la cache con dichas configuraciones.

```
php artisan config:clear
```

Continuar la instalación creando una API key para el backend y creando la estructura de la base de datos que utilizará mediante la ejecución de las migraciones.

```
php artisan key:generate
php artisan migrate:refresh
```

Instalar scripts SQL en la base de datos utilizada en caso de que estos existan.

Una vez llenada la base de datos con los datos iniciales y cada vez que se editen datos directamente desde la base de datos, es necesario ejecutar el siguiente comando que tiene por objetivo actualizar los índices de datos generados en Elasticsearch.

```
php artisan search:reindex
```

Finalmente, es necesario ejecutar el siguiente comando para generar las claves de encriptación que utiliza Passport para crear los tokens de acceso a usuarios.

```
php artisan passport:install
```

Instalar paquetes opcionales de assets si existiesen, siguiendo los pasos correspondientes.

[Frontend \(congresovirtual-frontend\)](#)

Copiar la carpeta del repositorio llamada congresovirtual-frontend a la ruta /var/www/congresovirtual-frontend/.

Instalar dependencias del frontend, entrando a la carpeta de éste.

```
cd /var/www/congresovirtual-frontend
npm install
```

Configurar el acceso al backend/API, editando para ello al archivo `/src/backend/data_server.js`. Allí, buscar `API_URL` y reemplazarla por la URL configurada en la instalación del backend. Por ejemplo:

```
const API_URL = 'https://url_api';
export {
  API_URL
};
```

Importante: Cada vez que se haga un cambio, habrá que correr los siguientes pasos en adelante en cuanto al frontend de Congreso Virtual.

Correr el compilador de VueJS ejecutando el siguiente comando en la carpeta raíz del proyecto.

```
cd /var/www/congresovirtual-frontend
npm run build --max-old-space-size=4096
```

Reemplace 4096 por la cantidad de RAM que piensa asignarle para la compilación. Usualmente aquel valor funciona bien en la mayoría de los casos. Espere un momento, ya que el proceso puede tardar unos minutos sin responder.

Al terminar la compilación, se habrá generado una carpeta **dist**. **Dentro de esta carpeta** cree un fichero llamado **.htaccess** El contenido del archivo debe ser el siguiente:

```
<IfModule mod_rewrite.c>
  RewriteEngine On
  RewriteBase /
  RewriteRule ^index\.html$ - [L]
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteRule . /index.html [L]
</IfModule>
```

**Nota:** Cuando se regenere el proyecto, este archivo puede ser sobrescrito o eliminado, por lo que deberá crear el archivo nuevamente. De forma alternativa, puede dejar una copia de seguridad lista para restaurar luego de correr el compilador de VueJS.

### Instalación de los módulos de analítica (congresovirtual-data-analytics)

Copiar la carpeta del repositorio llamada `congresovirtual-data-analytics` a la ruta `/var/www/congresovirtual-data-analytics/`.

Instalar las dependencias de los módulos de analítica.

```
cd congresovirtual-data-analytics
sudo pip3 install -r requirements.txt
```



Editar el archivo **dbconfig.py**. En ella, ingresar los datos de conexión a la base de datos de Congreso Virtual (la misma a la cual se conecta el backend/API).

```
CONGRESO_MYSQL_HOST="localhost"
CONGRESO_MYSQL_USER="congresovirtual-bd"
CONGRESO_MYSQL_PASSWD="secret"
CONGRESO_MYSQL_DATABASE="congresovirtual-bd"
```

Configurar el sistema operativo para que inicie el servicio de analítica ante un reinicio del servidor con crontab. Para tal fin, iniciar Crontab.

```
crontab -e
```

**Nota:** Crontab utiliza el editor de texto Vim. Si desea utilizar otro editor (como Nano), luego de iniciar sesión por SSH escribir el siguiente comando.

```
export EDITOR=nano
```

En el editor escribir la siguiente línea.

```
@reboot screen -d -m bash -c "cd /var/www/congresovirtual-data-analytics/; python3 -m flask
run"
```

Guardar (importante) y salir.

Indicar al sistema operativo que debe correr Cron al inicio del sistema.

```
sudo chkconfig crond on
sudo systemctl enable crond.service
```

Iniciar el servicio de analítica ejecutando lo siguiente:

```
screen -d -m bash -c "cd /var/www/congresovirtual-data-analytics/; python3 -m flask
run"
```

**Nota:** Para ver el estado del servidor de analítica, se deberá usar una terminal paralela creada para tal fin. Para acceder a la terminal paralela, escribir **screen -r** y para salir de ella hay que presionar las teclas **Ctrl-a-d**.

## 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`).