Instrucciones para desplegar el SIA2 en un entorno de producción

Nombres: Francisco Ignacio Muñoz Valdebenito

Jorge Rodrigo Andrés Valdivia Peña

Ramo: Taller de Desarrollo

Profesor: Juan Paulo Soto Cornejo

Generación: 2019

Universidad del Bío-Bío

[**Consideraciones previas 3**](#_tjp238gh55h2)

[**Instrucciones 3**](#_dagbf84364nk)

[Transparentar puertos 3](#_wpr0iku7jj12)

[Instalación de dependencias 4](#_sgdnnulvgybj)

[Instalación de librerías 4](#_gpvba5g8fp9j)

[Instalación de Apache2 6](#_2rufz2j1lu6)

[Comando de instalación Apache2 6](#_prque68wzhla)

[Configuración de Apache2 6](#_18uux02zd6k9)

[Instalación de Composer 8](#_ey0bczwtt5f7)

[Comando de instalación de Composer 8](#_85emo8o1ciud)

[Descarga del proyecto 8](#_z0c0pqrp1nd5)

[Descarga del repositorio 8](#_6fgpggsqsapo)

[Instalación de dependencias 9](#_x7smwrw6d1p9)

[Configuración de permisos y acceso de carpetas 9](#_imw3ujciu4vh)

[Comando de cambio de permisos y propiedad 9](#_5jru32fsoq3g)

[Creación del VirtualHost para el proyecto 10](#_nv4qof3demdt)

[Instalación de MariaDB (MySQL) 11](#_kyqpqub9c253)

[Comando de instalación MariaDB 11](#_u6xkoa6xwh4u)

[Configuración de MariaDB 12](#_ajn70yeoqaz9)

[Respaldo de base de datos o migrar las tablas 13](#_3531s61334hw)

[Configuración del entorno de producción 13](#_qyerdynob0d4)

[Uso y generación de la llave de encriptación en Laravel (‘APP\_KEY’). 14](#_39m8cxzdwx03)

# **Instrucciones**

En este capítulo se detallarán todas las instrucciones con sus respectivos comandos para poder llevar a cabo una instalación exitosa.

## **Instalación de dependencias**

Se recomienda trabajar con un sistema operativo Debian 12 para garantizar la compatibilidad del proyecto.

### **Instalación de librerías**

Se debe ingresar como root/superusuario para poder llevar a cabo las instalaciones de librerías y dependencias generales con el siguiente comando:

**$ su –**

Luego instale todas las librerías básicas necesarias para la instalación y clonación del proyecto a través del comando:

**$ sudo apt-get update && sudo apt-get install -y curl git unzip libzip-dev libicu-dev libonig-dev libpng-dev libjpeg-dev libfreetype6-dev libxml2-dev**

**Instalando PHP y extensiones**

Para instalar PHP con las respectivas extensiones necesarias se debe introducir el siguiente comando:

**$ sudo apt-get install -y php libapache2-mod-php php-dom php-xml php-curl php-mysql**

## **Instalación del servidor web**

Se recomienda la instalación del servidor web de Apache2 para garantizar la compatibilidad introduciendo el siguiente comando:

**$ sudo apt-get install apache2 -y**

Sin embargo, se puede utilizar otro servicio web para lograr el mismo objetivo.

### **Configuración del servidor web**

Si usted ha transparentado un puerto, asegúrese de cambiarlo en el archivo de configuración de su servidor web usando un editor de texto:

**‘EDITOR\_TEXTO /etc/SERVICIO\_WEB/ARCHIVO\_CONFIGURACION\_PUERTO.**

En el caso de utilizar apache2 (recomendado):

**‘nano /etc/apache2/ports.conf’**

En el caso de ocupar otra IP, el procedimiento es similar. Diríjase al archivo de configuración de su servicio web:

**‘EDITOR\_TEXTO /etc/SERVICIO\_WEB/ARCHIVO\_CONFIGURACION\_SERVICIO’**

En el caso de apache2 (recomendado):

**‘nano /etc/apache2/apache2.conf’**

Esto hará que el servicio web utilice la IP designada en su máquina. Luego guarde el archivo y recargue el servicio.

En el caso de usar apache2 es importante habilitar el módulo **“rewrite”** de apache con el comando **“a2enmod rewrite”** y reiniciar el servicio con **“service apache2 restart”**.

**En caso de usar otro servicio web, busque el comando equivalente.**

Si ha seguido todos los pasos correctamente, asegúrese que pueda visualizar la página por defecto en **‘**[**http://IP\_MAQUINA:PUERTO\_TRANSPARENTADO/**](about:blank)**’**.

## **Instalación de Composer**

Para instalar dicho gestor, ingrese el siguiente comando como usuario root:  
 **$ curl -sSL https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer**

Una vez instalado, podemos corroborar la instalación ejecutando el comando:

**‘$ composer –version’**

## **Descarga del proyecto**

El siguiente paso es descargar el proyecto y configurarlo. Para ello se dividirá esta sección en 2 partes, una para descargarlo en un directorio recomendado y otra para configurar todas sus propiedades.

### **Descarga del repositorio**

Para descargar el repositorio desde GIT, elija un directorio donde clonara el repositorio. Una vez dentro introduzca siguiente comando:

**$ git clone** [**https://github.com/l4s10/SIA2\_SII.git**](https://github.com/l4s10/SIA2_SII.git)

### **Instalación de dependencias**

Una vez descargado el proyecto, entre al directorio con el comando:

**$ cd /DIRECTORIO\_RAIZ\_PROYECTO**

Una vez allí, ejecute el comando:

**‘$ composer install’**

## **Configuración de permisos y acceso de carpetas**

Generalmente en sistemas GNU/Linux, Laravel puede presentar conflictos con algunos permisos de sus directorios. Para subsanar este problema, ejecute el siguiente comando:

**$chown -R www-data:www-data /DIRECTORIO\_PROYECTO/storage DIRECTORIO\_PRYECTO/bootstrap/cache**

## **Creación del VirtualHost para el proyecto**

Ahora nos ubicamos en la carpeta de configuración de VirtualHost con el comando:

**$ cd /etc/SERVICIO\_WEB/DIRECTORIO\_HOSTING\_COMPARTIDO**

Una vez dentro, deberemos crear un archivo de configuración para el proyecto respetando las configuraciones de su servidor web y el directorio raíz, en el caso de apache2:

**<VirtualHost \*:PUERTO\_TRANSPARENTADO>**

**ServerAdmin webmaster@IP\_MAQUINA**

**DocumentRoot /DIRECTORIO\_RAIZ\_PROYECTO/public**

**<Directory /DIRECTORIO\_RAIZ\_PROYECTO/public>**

**AllowOverride All**

**Require all granted**

**</Directory>**

**ErrorLog ${APACHE\_LOG\_DIR}/error.log**

**CustomLog ${APACHE\_LOG\_DIR}/access.log combined**

**</VirtualHost>**

Ahora debemos habilitar el archivo de configuración creado, en el caso de apache2:

**“$ a2ensite NOMBRE\_ARCHIVO.conf”**

Posteriormente, deshabilitamos el sitio por defecto de su servidor web, en el caso de apache2 con:

**“a2dissite 000-default.conf”**

Y finalmente reiniciar el servicio web, en el caso de apache2:

**‘service apache2 restart’**

No olvide verificar que la sintaxis de su configuración esta correcta, en el caso de apache2:

**‘$ apache2ctl configtest’**

Si entramos ahora al enlace http://IP\_MAQUINA:PUERTO\_CONFIGURADO si aparece una página mostrando un “error 500”, significa que el archivo de configuración funciona correctamente, pero dicho error se muestra debido a que aún falta configurar la base de datos y el entorno del proyecto.

## **Instalación de MariaDB (MySQL)**

**(En caso de que usted ya posea un servidor de base de datos, ignore esta sección).**

Ahora debemos instalar y configurar la base de datos. Para ello debemos hacer lo siguiente.

### **Comando de instalación MariaDB**

Como usuario ROOT debemos ejecutar el siguiente comando para instalar MariaDB:  
 **$ apt-get install mariadb-server -y**

Una vez instalado debemos ejecutar la instalación segura:

**$ mysql\_secure\_installation**

Se le preguntará por la contraseña de su usuario root, por lo que deberá ingresarla.

En **“Switch to unix socket authentication”** ingrese **“n”**.

En **“Change root password”** si ya tiene una ingrese **“n”**.

En **“Remove anonymous users”** ingrese **“y”**.

En **“Disallow root login remotely”** ingrese **“y”**.

En **“Remove test databases and access to it”** ingrese **“y”**.

Finalmente, recargue la tabla de privilegios cuando la terminal diga **“Reload privilege tables now?”**, ingrese **“y”**.

Seguido de esto recibirá un mensaje de MariaDB diciendo que ya ha instalado correctamente MariaDB, en caso de cualquier problema se puede volver a ejecutar el comando anterior y cambiar los ajustes.

### **Configuración de MariaDB**

Para configurar la base de datos primeramente necesitamos entrar como root a MariaDB por lo que con el comando:

**$ mariadb -u root -p**

Una vez dentro, crearemos una nueva base de datos con el comando:

**“CREATE DATABASE NOMBRE\_BDD;”**.

Si todo ha funcionado correctamente, usted será capaz de visualizar la nueva base de datos con el comando:

**“SHOW DATABASES;”**.

Ahora debemos configurar un usuario para la base de datos. Esto se hace con el siguiente comando dentro del motor:

**“CREATE USER ‘nombre\_usuario’@’localhost’ IDENTIFIED BY ‘contraseña’;”**.

Una vez realizado este paso, lo próximo será otorgarle los permisos al usuario que acabamos de crear.

**“GRANT ALL PRIVILEGES ON NOMBRE\_BDD.\* TO ‘usuario\_creado’@’IP’;”**.

Una vez otorgados los permisos debemos refrescar la tabla de privilegios con el comando **“FLUSH PRIVILEGES;”**.

Una vez hecho esto, escribimos

**“use NOMBRE\_BDD;”**

Para marcarla como “en uso”. Y salimos con el comando:

**exit**.

### **Respaldo de base de datos o migrar las tablas**

Si este instructivo contiene dentro de sus archivos uno denominado “respaldo.sql”, usted puede hacer el siguiente paso.

Con un comando, usted puede importar este archivo.sql el cual contendrá los datos necesarios para la correcta ejecución del proyecto. Este es:

**“mariadb -u root -p NOMBRE\_BDD < DIRECTORIO\_ARCHIVO.SQL”**

Por ejemplo: “mariadb -u root -p sia2 < /home/taller/DockerLaravel/respaldo.sql”

si todo funciona correctamente, puede entrar a revisar la base de datos y ejecutar el comando:

**“SHOW TABLES;”**

Podrá ver que la base de datos ya se encuentra poblada.

## **Configuración del entorno de producción**

Ahora debemos conectar el servidor de base de datos con el proyecto en cuestión. Por lo que ahora debemos seguir esta serie de pasos:

1. Diríjase al directorio en donde se encuentra el proyecto, con el comando **“$cd directorio\_proyecto”**. (Por ejemplo: cd /var/www/SIA2\_SII/).
2. Con el siguiente comando debe copiar un archivo llamado **“.env.example”** y llamarlo **“.env”**. **$cp .env.example .env**.
3. Con el editor de texto de su preferencia, usted debe abrir el archivo recién copiado.

Dentro del archivo **“.env”** usted debe tener algo similar a esto:

APP\_NAME**=**Laravel

APP\_ENV**=**local

APP\_KEY**=**

APP\_DEBUG**=**true

APP\_URL**=**http://localhost

DB\_CONNECTION**=**mysql

DB\_HOST**=**db

DB\_PORT**=**3306

DB\_DATABASE**=**laravel

DB\_USERNAME**=**laraveluser

DB\_PASSWORD**=**yourpassword

Y cambiarlo de la siguiente forma:

APP\_NAME**=**Laravel

APP\_ENV**=**production

APP\_KEY**=**

APP\_DEBUG**=**true

APP\_URL**=**http://localhost

DB\_CONNECTION**=**MOTOR\_BDD

DB\_HOST**=**IP\_SERVIDOR\_BDD

DB\_PORT**=**PUERTO\_BDD

DB\_DATABASE**=**NOMBRE\_BDD

DB\_USERNAME**=**USUARIO\_BDD

DB\_PASSWORD**=**CREDENCIAL\_USUARIO\_BDD

## **Uso y generación de la llave de encriptación en Laravel (‘APP\_KEY’).**

Para finalizar, lo único que queda por hacer es generar la llave de encriptación de la aplicación de Laravel.

Para generar una nueva llave de aplicación en un proyecto Laravel, siga estos pasos:

1. Navegar al Directorio del Proyecto:
   1. **‘cd /DIRECTORIO\_RAIZ\_PROYECTO’**
2. Ejecutar el Comando de Generación de llave:
   1. **$ php artisan key:generate**

Una vez hecho esto visite en un navegador su sitio configurado previamente:

**“http:/IP\_SERVIDOR:PUERTO\_TRANSPARENTADO/”**

Y utilice las credenciales de prueba que se encuentran en el **README.md**