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)

# **Consideraciones previas**

Antes de comenzar, es importante repasar una serie de condiciones que se deben de tener a consideración antes de seguir las instrucciones.

● Se debe tomar en cuenta la IP de la máquina que actuará de “servidor” para alojar el proyecto a instalar.

● Al igual que la IP, se debe también considerar los puertos a utilizar. Siendo estos principalmente el puerto destinado para apache, base de datos, y el puerto del proyecto persé.

● Con base en lo anterior, si usted se encuentra trabajando desde una máquina virtual dependiendo de qué puertos tenga disponibles o desee transparentar, es importante configurarlos en su máquina virtual.

● Se recomienda llevar a cabo la instalación en un sistema operativo “Debian 12” para garantizar una instalación exitosa debido a que el sistema fue concebido en esta distribución de GNU/Linux.

Sin más que añadir a este capítulo, el lector ya poseería las herramientas necesarias para poder comenzar a implementar el proyecto en un entorno de producción.

# **Instrucciones**

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

## **Transparentar puertos**

Si usted está ocupando una máquina virtual sea o no con una IP otorgada dentro de una red interna y/o va a utilizar otros puertos para levantar el proyecto a instalar, debe realizar una configuración previa.

1. Abra su configuración de red.

2. Conéctelo a “NAT”

3. Haga clic en “Avanzado” y en “Reenvío de puertos”.

4. Agregue tanto los puertos de la máquina anfitrión como los transparentados de la máquina virtual y recuerdelas y/o anótelas para configurar el entorno más adelante.

## **Instalación de dependencias**

Independiente si posee una máquina virtual o no (preferiblemente un sistema operativo Debian 12), se debe ingresar como ROOT/superusuario para poder llevar a cabo las instalaciones de librerías y dependencias generales.

### **Instalación de librerías**

Inicie sesión como usuario root con el comando

**$ su –**

Con el siguiente comando usted instalará todas las librerías básicas necesarias para la instalación y clonación del proyecto:

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

En resumen, lo que hace este comando es lo siguiente:

● Se actualiza la lista de paquetes disponibles y sus versiones, pero no instala ni actualiza ninguna de ellas.

● Instala las dependencias con **‘apt-get install -y’** aceptando dicha instalación

● **‘curl’**: Una herramienta para transferir datos desde o hacia un servidor.

● **‘git’**: Un sistema de control de versiones distribuido (es donde está alojado el proyecto).

● **‘unzip’**: Una utilidad para descomprimir archivos.

● **‘libzip-dev’**: Biblioteca para manejar archivos ZIP.

● **‘libicu-dev’**: Biblioteca para soporte internacional de caracteres “Unicode”.

● **‘libonig-dev’**: Biblioteca para expresiones regulares.

● **‘libpng-dev’**: Para la manipulación de imágenes PNG.

● **‘libjpeg-dev’**: Para la manipulación de imágenes JPEG.

● **‘libfreetype6-dev’**: Para la renderización de fuentes.

● **‘libxml2-dev’**: Una biblioteca para procesar archivos XML.

Una vez instaladas todas estas librerías, el siguiente paso es la instalación del lenguaje de programación PHP.

**Instalando PHP y extensiones**

El lenguaje de programación PHP es una dependencia muy importante para la correcta ejecución del proyecto, principalmente por ser el lenguaje base que utiliza Laravel, el cual es el framework con el que se ha desarrollado el proyecto. Por lo que, sin PHP, el proyecto no podría funcionar correctamente.

Además, también es necesario para la instalación de Apache2 el cual será el servidor web HTTP que alojará el proyecto y lo levantará.

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

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

Esto lo que hará será instalar el intérprete de PHP junto con una serie de instrucciones:

● **‘libapache2-mod-php’**: Instala el módulo de PHP para el servidor web Apache. Permite a Apache interpretar y ejecutar scripts PHP. Este módulo es necesario para poder servir aplicaciones PHP a través de Apache como es nuestro caso.

● **‘php-dom’ y ‘php-xml’**: Estas extensiones se utilizan para trabajar con documentos XML. Son esenciales para tareas que involucran la manipulación o análisis de XML, como lectura, escritura y modificación de archivos XML.

● **‘php-curl’**: Permite a PHP utilizar bibliotecas CURL para realizar solicitudes HTTP. Es muy utilizada para acceder o recuperar información de la base de datos de otras webs o servicios web.

● **‘php-mysql’**: Instala la extensión de PHP para interactuar con las bases de datos MySQL. Es esencial para aplicaciones web que necesiten conectarse y realizar operaciones en bases de datos MySQL como es este caso.

## **Instalación de Apache2**

Como se mencionó anteriormente, Apache2 será el servicio que levantará nuestra aplicación y para ello se debe instalar siguiendo una serie de pasos.

### **Comando de instalación Apache2**

Para instalar Apache2 se debe introducir el siguiente comando sin salir del usuario ROOT:

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

Consideración: a veces puede pasar que la máquina virtual indique que ya está instalado apache2 y eso se debe a que al momento de instalar la extensión de PHP que necesita apache2, la máquina detectará que aún no se ha instalado y procederá a hacerlo de forma automática, igual se recomienda ejecutar el comando para verificar que esté instalado.

### **Configuración de Apache2**

Generalmente, el puerto 80 es el que se habilita por defecto una vez instalado apache, pero si usted configuró otro puerto u otra IP en la máquina virtual, para este procedimiento se recomienda tomar las siguientes precauciones:

● Debe modificar el archivo de configuración ubicado en **‘/etc/apache2/ports.conf’** y en la línea que dice **‘Listen 80’** usted debe cambiarla por **‘Listen PUERTO\_CONFIGURADO’**. Una vez guardado, recargue el servicio de apache2 con **‘service apache2 restart’**. Esto lo que hará es cambiar el puerto donde apache2 va a trabajar.

● Recuerde que ese será el puerto que usted utilizará para configurar los diversos virtualhost.

● Adicionalmente, debe configurar el VirtualHost por defecto de apache editando el archivo **‘/etc/apache2/sites-available/000-default.conf’** y cambiar la línea que dice **‘<VirtualHost \*:80>’** por **‘<VirtualHost \*:PUERTO\_CONFIGURADO>’** para que la página por defecto de apache se levante en el puerto que haya configurado. Una vez hecho esto, guarde el archivo y ciérrelo. Para poder aplicar los cambios debe deshabilitar el sitio con el comando **‘a2dissite 000-default.conf’**, recargar apache con el comando **‘service apache2 restart’**, habilitarlo de nuevo con **‘a2ensite 000-default.conf’** y recargar con el mismo comando service.

● En el caso de ocupar otra IP, el procedimiento es similar. Diríjase al archivo **‘/ect/apache2/apache2.conf’** y al final del archivo escriba lo siguiente **‘ServerName IP\_MAQUINA’**. Esto hará que el apache tome como el nombre del servidor la IP que usted esté utilizando en su máquina. Luego guarde el archivo y recargue apache2 con **‘service apache2 restart’**.

Independientemente, si usted configuro los puertos o no, se debe de habilitar el módulo **“rewrite”** de apache con el comando **“a2enmod rewrite”** y reiniciar el servicio con **“service apache2 restart”**. Es muy importante habilitar este módulo, debido a que sin este el enrutamiento de la aplicación no funcionará correctamente.

Si ha seguido todos los pasos correctamente, asegúrese que pueda visualizar la página por defecto en **‘**[**http://IP\_MAQUINA:PUERTO\_TRANSPARENTADO/**](about:blank)**’** (Ejemplo: http://localhost:8080), si logra visualizar la página significa que ha terminado de configurar correctamente apache2.

## **Instalación de Composer**

Composer en Laravel es un gestor de dependencias para PHP, utilizado para manejar las librerías y paquetes que una aplicación Laravel requiere, facilitando su instalación y actualización.

### **Comando de instalación de Composer**

Para instalar dicho gestor, es importante que en las librerías instaladas se encuentre ‘curl’ entre ellas debido a que el comando que utilizaremos para instalarlo será con curl:  
 **$ curl -sSL https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer**

Lo que hará este comando será instalar composer y ejecutar el comando de PHP que lo instale en el directorio **‘/usr/local/bin’**. Para no tener que hacerlo manualmente más tarde.

Una vez instalado ejecutando el comando **‘composer –version’** la versión con 2 guiones podremos ver que el comando se instaló correctamente indicando la versión.

## **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, debemos ubicarnos en la carpeta **‘/var/www/’**, con el comando **‘$cd /var/www’**. Una vez dentro se debe clonar el proyecto con el 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, lo siguiente es entrar a la carpeta **‘SIA2\_SII’** con el comando

**$ cd /SIA2\_SII’**

Una vez dentro del directorio se debe ejecutar el comando

**‘$ composer install’**

El cual instala todas las dependencias del proyecto dentro de la carpeta raíz de este.

En este punto ya habremos instalado el proyecto e instalado sus dependencias, restando solamente la configuración del entorno del proyecto, pero eso deberemos hacerlo más tarde porque aún falta hacer 3 cosas:

● Cambiar permisos de algunos archivos.

● Crear el VirtualHost para el proyecto.

● Instalar MariaDB, configurar el usuario y la base de datos.

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

Es importante destacar que en sistemas como GNU/Linux, Laravel puede dar algunos problemas al momento de montar proyectos en Linux, y esto se debe principalmente a 2 carpetas a las que apache no tiene permiso por defecto, haciendo que los logs de la aplicación no se puedan ni leer y escribir. Lo cual es un aspecto crucial de toda aplicación hecha con Laravel. Para corregir esto se debe realizar la siguiente serie de pasos.

### **Comando de cambio de permisos y propiedad**

Para cambiar de dueño ciertos archivos que apache necesita para levantar el sistema es el siguiente:

**$chown -R www-data:www-data /var/www/SIA2\_SII/storage /var/www/SIA2\_SII/bootstrap/cache**

Adicionalmente, se puede ejecutar este comando por si no funciona el anterior o para asegurar los permisos necesarios.

**$ chmod -R 775 /var/www/SIA2\_SII/storage /var/www/SIA2\_SII/bootstrap/cache**

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

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

**$ cd /etc/apache2/sites-available**

Una vez dentro, deberemos crear un archivo de configuración con el comando

**$ nano NOMBRE\_ARCHIVO.conf**

Respetando que primero sea el nombre del archivo con extensión.conf. Una vez dentro del archivo con nano, deberemos escribir la siguiente configuración:

**<VirtualHost \*:PUERTO\_CONFIGURADO>**

**ServerAdmin webmaster@IP\_MAQUINA**

**DocumentRoot /var/www/SIA2\_SII/public**

**<Directory /var/www/SIA2\_SII/public>**

**AllowOverride All**

**Require all granted**

**</Directory>**

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

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

**</VirtualHost>**

Una vez escrito, guardamos con la combinación de teclas **CTRL + o** seguido de la tecla **ENTER** para confirmar el nombre del archivo y cerramos nano con **CTRL + x**.

Ahora debemos habilitar el archivo de configuración creado con el comando:

**“$ a2ensite NOMBRE\_ARCHIVO.conf”**

deshabilitar el sitio por defecto con:

**“a2dissite 000-default.conf”**

Y finalmente reiniciar el servicio con el siguiente comando:

**“service apache2 restart”**.

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)**

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**

· mariadb, seguido de -u, especificamos usuario y -p debido a que usa contraseña.

Nos pedirá la contraseña del usuario ROOT previamente ingresada en el paso anterior o la nueva si usted deseó cambiarla al momento de instalar el motor.

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.

Como Laravel solo funciona con una cuenta de base de datos, tendremos que darle todos los permisos, pero solo en la base de datos que acabamos de crear, por lo que primero con el comando:

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

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”**

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 recapitulando un poco, instalamos las dependencias del proyecto, el servicio de apache con su respectiva configuración, Composer, clonamos el proyecto principal y configuramos la base de datos. Pero ahora debemos conectar la 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 NANO usted debe abrir el archivo recien copiado con el comando **“nano .env”**.
4. En la segunda linea que dice “**APP\_ENV=local”**  cámbiela por “**APP\_ENV=production**”. (esto hará que Laravel nos pida verificar cada vez que ejecutemos un comando artisan).
5. En la línea que dice **“DB\_DATABASE”**, cambie el valor asignado por el nombre de la base de datos que configuro en la sección anterior.
6. De igual manera, con la línea que diga **“DB\_USERNAME”** y **“DB\_PASSWORD”** Indique las credenciales que configuró en su motor para el nuevo usuario creado.
7. Una vez realizado todo esto, guarde el archivo y ciérrelo.

## **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 qué sirve la llave de aplicación en Laravel?**

1. **Cifrado de Datos**: La llave **APP\_KEY** es esencial para el cifrado de datos dentro de Laravel. Se utiliza para asegurar datos sensibles, especialmente aquellos almacenados en cookies y en la base de datos. Un cifrado fuerte y efectivo depende de una llave única y segura.
2. **Sesiones y Tokens de Seguridad**: Esta llave también se emplea en la generación de tokens de sesión y otros tokens de seguridad. Esto ayuda a proteger la aplicación contra ataques como el CSRF (Cross-Site Request Forgery), asegurando que las sesiones y acciones en la aplicación sean legítimas.
3. **Protección de Datos del Usuario**: Utilizar la **APP\_KEY** para el cifrado garantiza la seguridad de los datos del usuario, permitiendo que solo se acceda a ellos de manera autorizada y previniendo modificaciones no legítimas

.

**Nota Importante:**

● La llave de aplicación debe ser única para cada proyecto y nunca debe compartirse públicamente.

● Cambiar esta llave en un proyecto existente puede causar la invalidación de datos cifrados previamente, como las cookies de sesión, lo que podría resultar en que los usuarios sean deslogueados.

**Generación de la Llave de Aplicación:**

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

1. **Abrir la Terminal o Línea de Comandos**: Accede a la terminal o línea de comandos en tu entorno de desarrollo.
2. **Navegar al Directorio del Proyecto**: Usa el comando cd para navegar al directorio raíz de tu proyecto Laravel.
3. **Ejecutar el Comando de Generación**: Escribe y ejecuta el siguiente comando:

**$ php artisan key:generate**

Este comando crea una nueva llave de aplicación aleatoria y segura, y la coloca automáticamente en el archivo .env, el cual es el archivo de configuración de tu proyecto Laravel.

Luego puede ir al virtual host creado en las secciones anteriores y ya tendrá el sistema operativo.

Por ejemplo, visite **“http:/localhost:8080/”** (puede cambiar según los parámetros que haya configurado).

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