



# **AUTOSIRIS**

## **Guía Despliegue App Web**

Versión: 001

Fecha: 08/11/2023

Segunda Versión

Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso y por escrito a AutOsiris



<b>AutOsiris</b> <b>Guía Configuración Despliegue App Web</b>	<b>Servicio Nacional de Aprendizaje</b> <b>(SENA)</b>
--	--

## HOJA DE CONTROL

<b>Organismo</b>	Servicio Nacional de Aprendizaje (SENA)		
<b>Proyecto</b>	AutOsiris		
<b>Entregable</b>	Guía Configuración Despliegue App Web		
<b>Cliente</b>	Miscelánea y Papelería Osiris		
<b>Versión/Edición</b>	001	<b>Fecha Versión</b>	08/11/2023
<b>Aprobado por</b>		<b>Fecha Aprobación</b>	DD/MM/AAAA
		<b>Nº Total de Páginas</b>	25

## REGISTRO DE CAMBIOS

<b>Versión Documento</b>	<b>Causa del Cambio</b>	<b>Responsable del Cambio</b>	<b>Fecha del Cambio</b>
001	Versión inicial	Elián Eduardo Ibarra Contreras	08/11/2023

## CONTROL DE DISTRIBUCIÓN

<b>Nombre y Apellidos</b>
Elián Eduardo Ibarra
Juan Sebastián Silva Garzón
Alan Darién Prada



<b>AutOsiris</b> <b>Guía Configuración Despliegue App Web</b>	<b>Servicio Nacional de Aprendizaje</b> <b>(SENA)</b>
--	--

## Tabla de Contenido

<b>1. Introducción</b>	4
<b>2. Objetivo</b>	4
<b>3. Plan de Instalación</b>	4
<b>3.1 Crear Aplicación</b>	5
3.1.1 Tipo de Aplicación:	5
3.1.2 Nombre de Aplicación	6
3.1.3 Nombre Servidor CloudWays.	6
3.1.4 Nombre de proyecto	6
3.1.5 Servidor de Aplicaciones.	6
3.1.5 Localización Servidor.	7
<b>3.2 Conexión a Servidor</b>	7
3.2.1 Luego de crear nuestro servidor.	7
3.2.2 Credenciales de conexión remota.	7
3.2.3 Conectar servidor con credenciales.	8
3.2.3 Instalar composer en el proyecto.	11
3.2.4 Editar variables de entorno.	12
3.2.4 Conectar proyecto con Git/GitHub.	14
3.2.5 Conectar proyecto con Git/GitHub: Crear tablas/Ejecutar migraciones.	18
<b>4. Anexos</b>	19



<b>AutOsiris</b> <b>Guía Configuración Despliegue App Web</b>	<b>Servicio Nacional de Aprendizaje</b> <b>(SENA)</b>
--	--

## 1. Introducción

Esta guía de configuración despliegue de App Web tiene como objetivo definir los pasos para desplegar un proyecto de este tipo, con su uso de Frameworks y tecnologías variadas. Esta guía precisa cómo podemos subir un proyecto utilizando Laravel como framework para realizar el back-end de nuestro proyecto PHP, junto con un material de apoyo en el cual estará el video por el cual nos basamos para realizar esta práctica y documentación propia de Laravel.

## 2. Objetivo

El objetivo de este documento es definir el conjunto de pasos que fueron ejecutados por el grupo del proyecto AutOsiris para desplegar su proyecto de forma gratuita y con el uso del framework Laravel para construir su back-end.

## 3. Plan de Instalación

Para desplegar nuestro proyecto de Laravel utilizamos una herramienta de App Service llamada CloudWays, para realizar esta práctica, utilizamos la prueba gratuita de 3 días premium solo con registrarnos. Luego de ello seguimos con cada uno de los siguientes pasos:



## AutOsiris Guía Configuración Despliegue App Web

## Servicio Nacional de Aprendizaje (SENA)




### 3.1 Crear Aplicación

Para desplegar nuestra aplicación, primero debemos crear un servidor en el cual podremos registrar todos los proyectos y/o aplicaciones. Pero luego de iniciar sesión nos saltará un panel en el cual podremos encontrar los servidores:

DEPLOY YOUR MANAGED APPLICATION

**APPLICATION & SERVER DETAILS**

WordPress Version 6.2.2	Name Your Managed App Your Application	Name Your Managed Server Your Server	Name your Project Your Project
----------------------------	---	---	-----------------------------------

**SERVER SIZE**  
Please select your server size. You can always scale your server size whenever required.

1GB 1GB 2GB 2GB 4GB 4GB 8GB 8GB 16GB 16GB 32GB 32GB 48GB 48GB 64GB 64GB 96GB 128GB 192GB

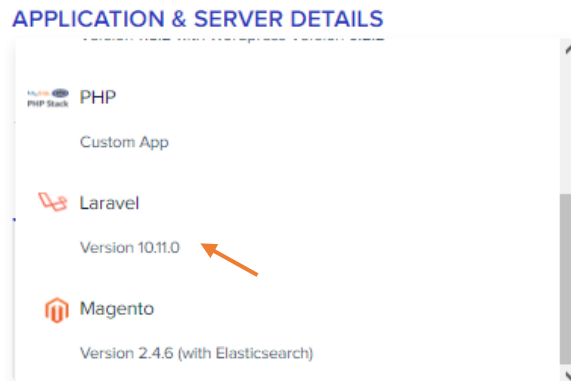
**LOCATION**  
Please select your server location.

New York

**3.1.1 Tipo de Aplicación:** Primero encontraremos que tipo de aplicación (**cuadro con borde rojo**) vamos a subir a la nube, podemos seleccionar la opción de Laravel:



<b>AutOsiris</b> <b>Guía Configuración Despliegue App Web</b>	<b>Servicio Nacional de Aprendizaje</b> <b>(SENA)</b>
--	--



**3.1.2 Nombre de Aplicación.** Luego le pondremos un nombre a nuestra aplicación **(cuadro borde azul)**

**3.1.3 Nombre Servidor CloudWays.** Para crear nuestra aplicación, debemos primero crear un servidor que CloudWays lo creará junto con nuestra aplicación, para esto debemos darle un nombre a nuestro servidor **(cuadro borde verde)**

**3.1.4 Nombre de proyecto.** Para crear nuestra aplicación, debemos al mismo tiempo crear un proyecto, en un proyecto podemos almacenar más de una aplicación, para esto le daremos un nombre **(cuadro borde naranja)**

**3.1.5 Servidor de Aplicaciones.** Para esta práctica hemos utilizado DigitalOcean, puesto que era el único que estaba disponible de forma gratuita, de igual forma esto nos va a permitir desplegar nuestra aplicación sin problemas.

Para seleccionar nuestro servidor DigitalOcean, debemos seleccionar un tamaño en gb (podremos cambiar este valor luego de crear el servidor). Para esta práctica de prueba vamos a utilizar 2gb **(cuadro borde negro)**

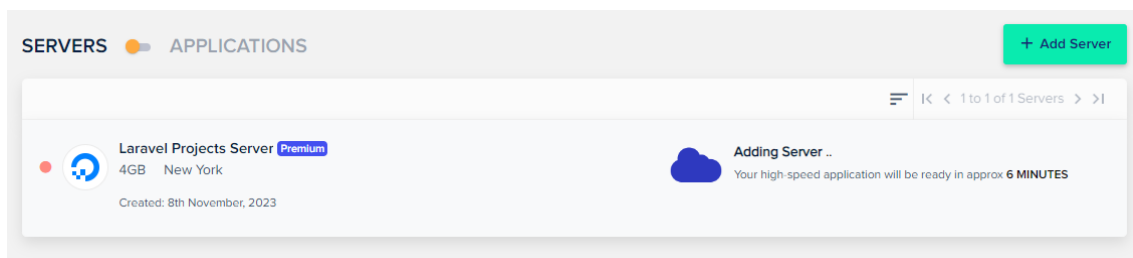


<b>AutOsiris</b> <b>Guía Configuración Despliegue App Web</b>	<b>Servicio Nacional de Aprendizaje</b> <b>(SENA)</b>
--	--

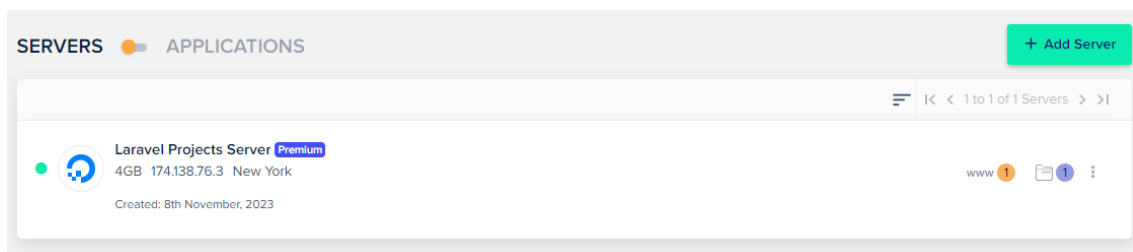
**3.1.5 Localización Servidor.** Luego de estos pasos, vamos a seleccionar una localización física para nuestro servidor, en este caso vamos a elegir New York (**cuadro borde violeta**)

## 3.2 Conexión a Servidor

**3.2.1 Luego de crear nuestro servidor.** Luego de crear nuestra aplicación junto con su proyecto y un servidor, nos encontraremos con esta vista:



Aquí está creando el servidor y luego de un tiempo lo tendremos disponible:



**3.2.2 Credenciales de conexión remota.** Luego de dar click en el servidor que hemos creado y que se ha registrado como “Laravel Project Server”, vamos a encontrar unas credenciales muy importantes que nos van a servir para conectarnos después desde nuestro computador al servidor y a subir la carpeta de nuestro proyecto de Laravel:



## AutOsiris Guía Configuración Despliegue App Web

## Servicio Nacional de Aprendizaje (SENA)

**MASTER CREDENTIALS**  
SFTP and SSH access details for all applications

Public IP:	174.138.76.3
Username:	master_gtnmxgqtrn
Password:	••••••••

On the left, you have Master credentials to gain the access for SFTP or SSH (e.g. using Putty). Alternatively, you can upload multiple SSH Public Keys to your Cloudways server to access it without password prompts.

[SSH PUBLIC KEYS](#)

Or, launch SSH terminal in your browser by clicking the button below.

[LAUNCH SSH TERMINAL](#)

**3.2.3 Conectar servidor con credenciales.** Para conectar nuestro servidor desde nuestro computador vamos a abrir la consola de nuestro sistema operativo (en mi caso Windows) y vamos a definir la ruta de la carpeta de nuestro proyecto de Laravel:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.3570]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\USUARIO>cd C:\Users\USUARIO\Desktop\AutOsiris - Despliegue_
```





## AutOsiris Guía Configuración Despliegue App Web

## Servicio Nacional de Aprendizaje (SENA)

Luego de posicionarnos en la carpeta de nuestro proyecto de Laravel, vamos a escribir la siguiente combinación junto con nuestras credenciales de servidor:

```
C:\Windows\system32\cmd.exe
C:\Users\USUARIO\Desktop\AutOsiris - Despliegue>ssh master_gtnmxgqtrn@174.138.76.3
```

Nombre de usuario de servidor      IP de servidor

Después de dar enter nos saldrá esto:

```
C:\Windows\system32\cmd.exe - ssh master_gtnmxgqtrn@174.138.76.3
C:\Users\USUARIO\Desktop\AutOsiris - Despliegue>ssh master_gtnmxgqtrn@174.138.76.3
master_gtnmxgqtrn@174.138.76.3's password: 
```

En este punto tendremos que escribir la contraseña del servidor, toda esta información la vamos a encontrar en el panel principal del servidor, tendremos que escribirla manualmente y esta no se verá por razones de seguridad. Luego de escribirla y que el sistema la valide ya estaremos dentro del servidor:

```
master_gtnmxgqtrn@174.138.76.3's password:
Linux 1161846.cloudwaysapps.com 5.10.0-26-amd64 #1 SMP Debian 5.10.197-1 (2023-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
(master_gtnmxgqtrn):~$
```



<b>AutOsiris</b> <b>Guía Configuración Despliegue App Web</b>	<b>Servicio Nacional de Aprendizaje</b> <b>(SENA)</b>
--	--

Ahora, vamos a tener que acceder a la aplicación que creamos dentro del servidor, para esto vamos a ejecutar el comando “ls”, esto para ver qué contiene y luego de ello accederemos a la carpeta “applications” con el comando “cd”:

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
[master_gtnmxgqtrn]:~$ ls
applications
[master_gtnmxgqtrn]:~$ cd applications
[master_gtnmxgqtrn]:applications$
```

Luego de esto, vamos a posicionarnos en la carpeta de nuestra aplicación, como tenemos sólo 1 aplicación dentro de nuestro servidor vamos a seleccionar la que nos aparece al ejecutar el comando “ls”, para luego elegir el directorio de esta aplicación con el comando “cd” seguido del nombre de tal carpeta:

```
[master_gtnmxgqtrn]:~$ cd applications
[master_gtnmxgqtrn]:applications$ ls
vmfrpfjkvx
[master_gtnmxgqtrn]:applications$ cd vmfrpfjkvx
[master_gtnmxgqtrn]:vmfrpfjkvx$
```

Así mismo vamos a posicionarnos dentro de la carpeta “public\_html” de la aplicación en donde vamos a ejecutar todos los comandos de composer y donde encontraremos todas las carpetas de nuestro proyecto de Laravel:



<b>AutOsiris</b> <b>Guía Configuración Despliegue App Web</b>	<b>Servicio Nacional de Aprendizaje</b> <b>(SENA)</b>
--	--

```
[master_gtnmxgqtrn]:applications$ cd vmfrpfjkvx
[master_gtnmxgqtrn]:vmfrpfjkvx$ ls
conf logs private_html public_html ssl tmp
[master_gtnmxgqtrn]:vmfrpfjkvx$ cd public_html
[master_gtnmxgqtrn]:public_html$ ls
README.md  artisan  composer.json  config  package.json  public  routes  tests  vite.config.js
app        bootstrap  composer.lock  database  phpunit.xml  resources  storage  vendor
```

3.2.3 Instalar composer en el proyecto. Aquí encontraremos todas las carpetas y archivos principales de un proyecto de Laravel, como lo es config/, public/, resources/, composer.json; etc. Pero vemos que no encontramos una carpeta muy importante, con la que podremos ejecutar acciones de nuestro proyecto como la creación de las tablas de la base de datos. Para esto vamos a instalar composer dentro de nuestra carpeta “public\_html”, con el comando “composer install”:

```
master_gtnmxgqtrn@1161846: ~/applications/vmfrpfjkvx/public_html

install [--prefer-source] [--prefer-dist] [--prefer-install PREFER-INSTALL] [--dry-run] [--download-only] [--dev] [--no-suggest] [--no-dev] [--no-autoloader] [--no-progress] [--no-install] [--audit] [--audit-format AUDIT-FORMAT] [-v|vv|vvv|--verbose] [-o|--optimize-autoloader] [-a|--classmap-authoritative] [--apcu-autoloader] [--apcu-autoloader-prefix APCU-AUTOLOADER-PREFIX] [--ignore-platform-req IGNORE-PLATFORM-REQ] [--ignore-platform-reqs] [--] [<packages>...]

[master_gtnmxgqtrn]:public_html$ composer install
Installing dependencies from lock file (including require-dev)
Verifying lock file contents can be installed on current platform.
Nothing to install, update or remove
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

81 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
[master_gtnmxgqtrn]:public_html$
```

Además, podemos verificar si composer se ha instalado correctamente con el comando: “composer”:



<b>AutOsiris</b> <b>Guía Configuración Despliegue App Web</b>	<b>Servicio Nacional de Aprendizaje</b> <b>(SENA)</b>
--	--

```
[master_gtnmxgqtrn]:public_html$ composer

Composer version 2.6.5 2023-10-06 10:11:52

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display help for the given command. When no command is given display help for the list
  -q, --quiet                Do not output any message
  -V, --version              Display this application version
  --ansi|--no-ansi          Force (or disable --no-ansi) ANSI output
  -n, --no-interaction       Do not ask any interactive question
  --profile                  Display timing and memory usage information
  --no-plugins               Whether to disable plugins.
  --no-scripts               Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache                 Prevent use of the cache
  -v|vv|vvv, --verbose       Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
  3 for debug
```

**3.2.4 Editar variables de entorno.** Ahora lo que vamos a hacer es editar nuestro archivo `.env`, en donde están ubicadas todas nuestras variables de entorno, en el cual podremos conectar nuestro proyecto con una base de datos y así crear nuestras tablas. Para esto, vamos a escribir el comando “`nano .env`”, esto ya que “`nano`” es un comando que nos permite editar cualquier archivo de la carpeta que tengamos en nuestra carpeta, y le añadiremos el nombre del archivo:

```
GNU nano 5.4 .env *
APP_NAME=AutOsirisDespliegue
APP_ENV=local
APP_KEY=base64:eQ34lMFcHCJ2kDuY1bFtF3KvzQlPGw8ispBss0+zoL4=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

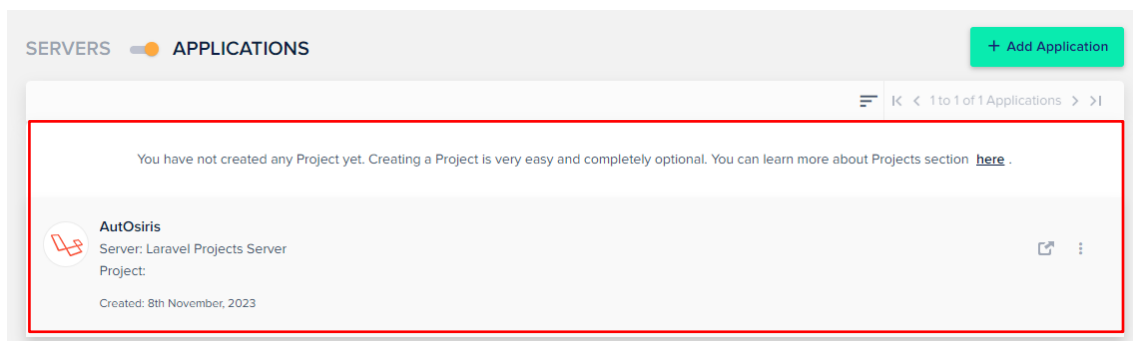
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File ^J Replace   ^U Paste    ^J Justify  ^_ Go To Line M-E Redo     M-G Copy
```



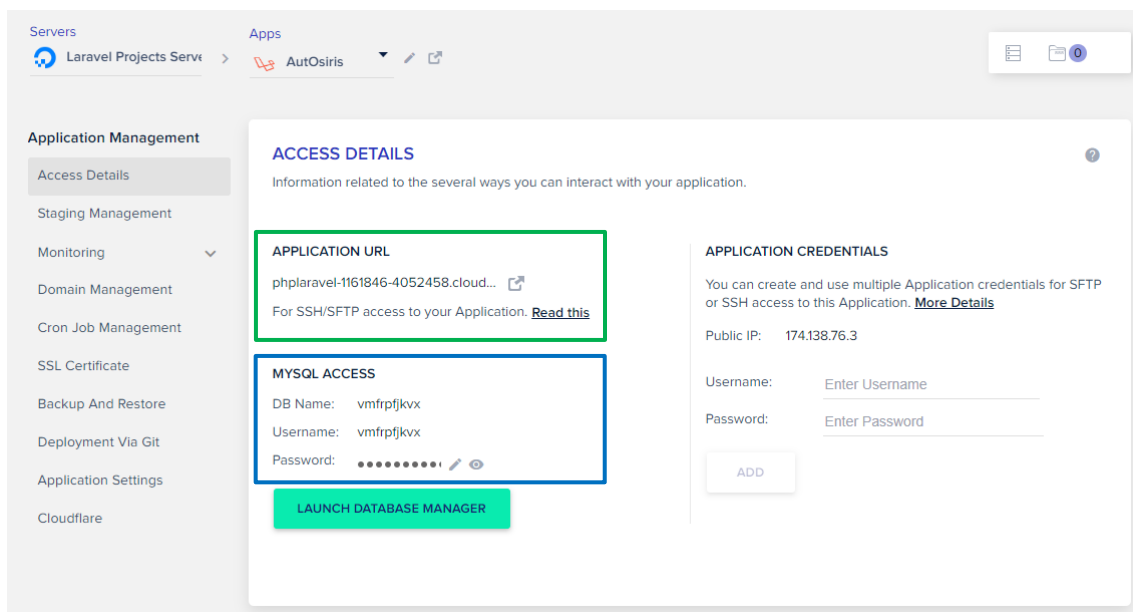
## AutOsiris Guía Configuración Despliegue App Web

## Servicio Nacional de Aprendizaje (SENA)

Ahora, vamos a devolvemos a nuestra página de CloudWays y vamos a seleccionar en el menú superior la opción de Aplicaciones, en la cual vamos a elegir la que hemos creado y conectado desde nuestro computador:



Luego de entrar a nuestra aplicación veremos este panel:



Primero encontramos la URL que generó CloudWays para acceder a la página principal o index de nuestro proyecto (**cuadro borde verde**)



## AutOsiris Guía Configuración Despliegue App Web

## Servicio Nacional de Aprendizaje (SENA)

Luego veremos información para acceder a nuestra base de datos MySQL, dentro de esto está el nombre de la base de datos, el usuario y la contraseña (**cuadro borde azul**)

Estos datos tendremos que añadirlos a nuestro archivo .env:

```
GNU nano 5.4 .env *
APP_NAME=AutOsirisDespliegue
APP_ENV=local
APP_KEY=base64:eQ34lMfCHCJ2kDuY1bFtF3KvzQlPGw8ispBss0+zoL4=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=vmfrpfjkvx
DB_USERNAME=vmfrpfjkvx
DB_PASSWORD=
BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo      M-G Copy
```

Luego de esto vamos a guardar los cambios con Ctrl + x, y tecleamos “Y” cuando nos pregunten si queremos guardar estos cambios:

```
Save modified buffer?
Y Yes
N No      ^G Cancel
```

**3.2.4 Conectar proyecto con Git/GitHub.** Después de esto vamos a generar una clave SSH para activar el proyecto, para esto vamos a ir a la sección de “Deployment via Git” y generaremos una clave SSH y la podremos ver dando click al botón (**con borde rojo**):



## AutOsiris Guía Configuración Despliegue App Web

## Servicio Nacional de Aprendizaje (SENA)

**DEPLOYMENT VIA GIT**  
Deploy code via Git

An SSH key has been generated. You can download it below. You will need to add the generated public key to your Git vendor (such as Bitbucket, Github, Assembla or others).

Once you have added the public key, insert your repository address and click Authenticate. This will pull the branch(es) name (master will be selected as default). To deploy the code click Start Deployment. You are done! From now onwards, you can simply pull your latest changes whenever you want.

**VIEW SSH KEY** **VIEW DEPLOYMENT LOGS**

Git Remote Address:  **Authenticate**

Branch:  **Fetch**

Deployment path:  public\_html/

**START DEPLOYMENT**

**Note:** Your application has been configured to use a custom webroot directory. You will be required to modify your Git deployment path accordingly.

Luego de esto veremos una ventana emergente en la que veremos la clave SSH que tendremos que copiar dando click y posteriormente añadir al repositorio de GitHub que tengamos del proyecto:

**SSH KEY**

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDgRzpBsNjbB1k61hrVjrGjHgRfzAsh8QNvSduXnBA78nmL9x5WVRfg9wWPUHwVLEg9dzwEniJyaArf  
pPdRZwmPTc6mP54LVPjOabsOUHbSFeMwvLSZMo7TaVDEJioASTD6kqJjrhqhpq3BKrEXoKOGkEmiV1SIYJ9VTLBKVAccMPxiFIAMJ/157x92caJooBxhL  
NWanQMypWH4RwLC9ef0XkCIEjdggam4fLRTQOEBSFSW9n0i5DFtjppSlyfVWGDNMvYQIvjuhgiR1acxe8kuWyYXirJouOwdmy4T72Weglxm0jfi7ea394B  
QS88+Jan6hgAqz0VNdGzJ@gmail.com
```

**DOWNLOAD SSH KEY** **CANCEL**

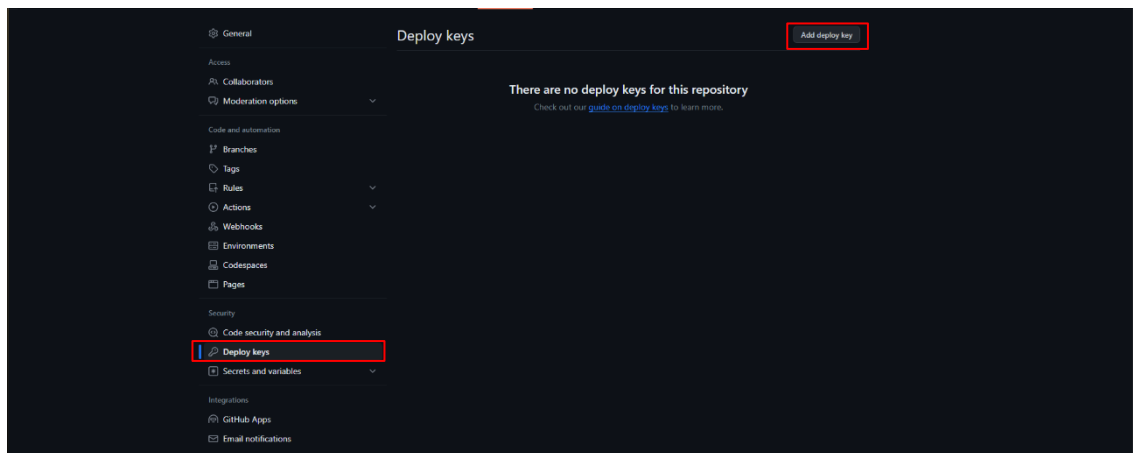
Para añadir esta clave a nuestro repositorio de GitHub tendremos que ir a configuraciones del repositorio en la sección SSH Keys y le daremos click al botón de



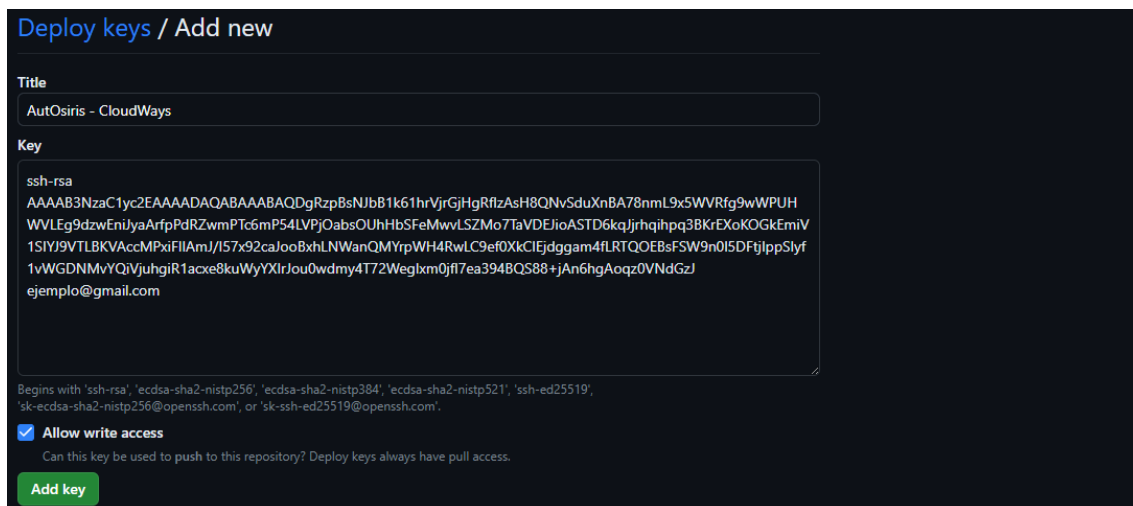
## AutOsiris Guía Configuración Despliegue App Web

## Servicio Nacional de Aprendizaje (SENA)

Añadir clave de despliegue:



Luego le añadiremos un título y pegaremos la clave que copiamos desde CloudWays:



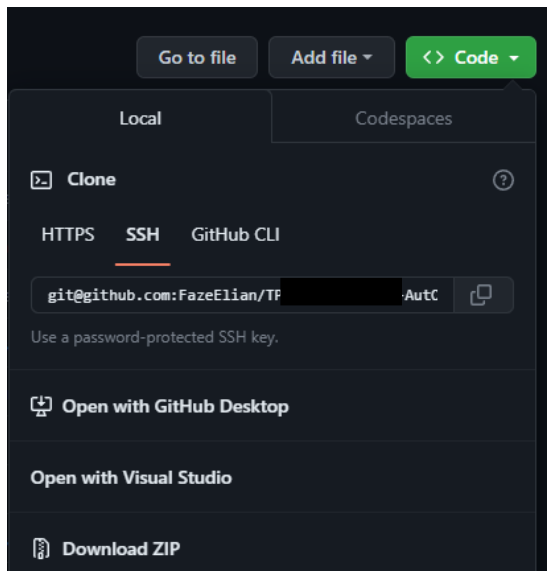
Luego de esto, vamos a ir a la página principal de nuestro repositorio y vamos a copiar la dirección SSH:



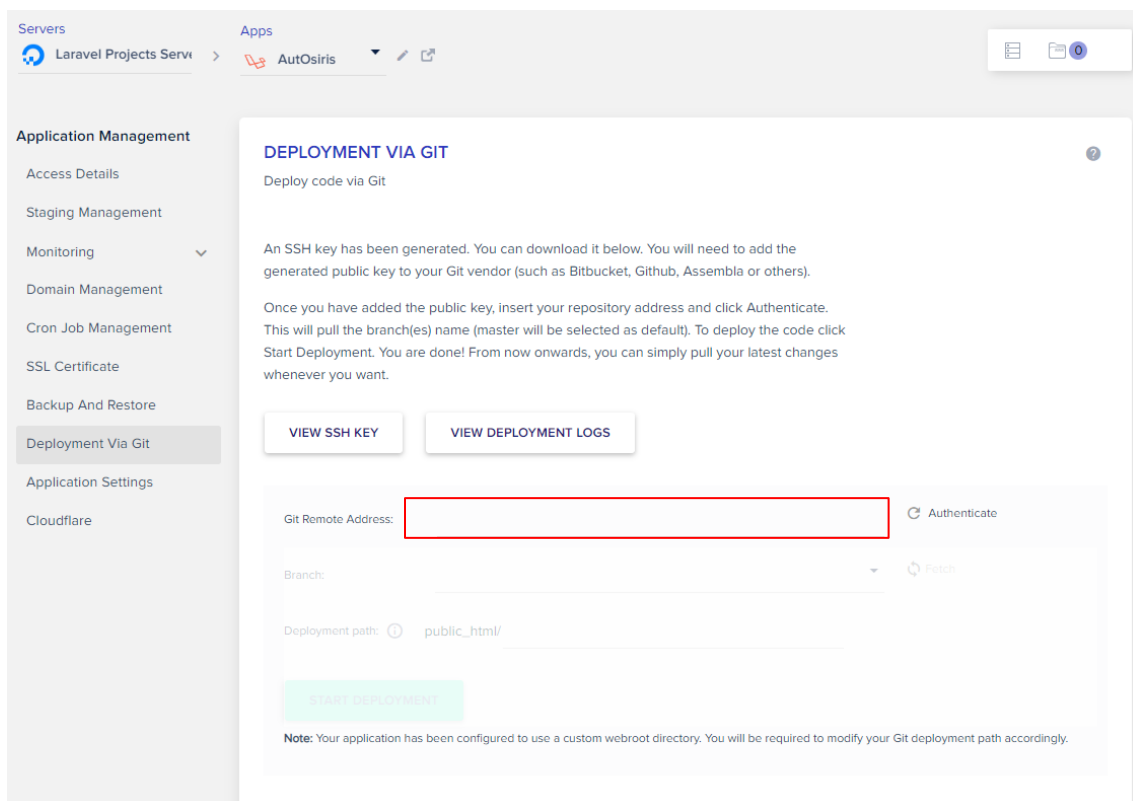


## AutOsiris Guía Configuración Despliegue App Web

## Servicio Nacional de Aprendizaje (SENA)



Después de lo anterior, vamos a pegar esa dirección SSH de nuestro repositorio en la vista donde copiamos la clave SSH de CloudWAYS





## AutOsiris Guía Configuración Despliegue App Web

## Servicio Nacional de Aprendizaje (SENA)

Luego le daremos click a el botón de autenticación y seleccionaremos la rama de nuestro repositorio e iniciaremos el despliegue:

### 3.2.5 Conectar proyecto con Git/GitHub: Crear tablas/Ejecutar migraciones.

Agregado a lo anterior vamos a ejecutar el comando de composer que nos permite ejecutar las migraciones (este es “php artisan migrate”), estas son las que nos van a permitir crear todas las tablas:

```
[master_gtnmxgqtrn]:public_html$ php artisan migrate
INFO: Preparing database.
Creating migration table ..... 6ms DONE
INFO: Running migrations.
2014_10_12_000000_create_users_table ..... 7ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 8ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 6ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 9ms DONE
[master_gtnmxgqtrn]:public_html$
```

Además de esto vamos a ejecutar todos los comandos de instalación de paquetes, dependencias y demás cosas que instalamos en nuestro proyecto desde local para que se aplique a producción.



<b>AutOsiris</b> <b>Guía Configuración Despliegue App Web</b>	<b>Servicio Nacional de Aprendizaje</b> <b>(SENA)</b>
--	--

## 4. Anexos

### Anexo 1: Video Guía

Este anexo incluye un video de YouTube en el cual nos basamos para crear esta Guía Configuración Despliegue App Web.

[Video desde YouTube](#)

### Anexo 2: Repositorio GitHub

En este anexo, se presenta el repositorio de GitHub en el cual se realizó esta práctica y en el cual está alojado todo el proyecto formativo de AutOsiris.

[Link Repositorio](#)