**SIMPLE 2.0**

**Requerimientos**

* NodeJS >= 8.11.3
* NPM >= 5.6.0
* MySQL 5.7 ó MariaDB 10.2
* PHP 7.1
* Librerías PHP necesarias:
	+ OpenSSL
	+ PDO
	+ PDO\_MYSQL
	+ Mbstring
	+ Tokenizer
	+ curl
	+ mcrypt
	+ Ctype
	+ XML
	+ JSON
	+ GD
	+ SOAP

**Instalación**

**Mysql >= 5.7**

Si estas usando una versión mayor o igual a MySQL 5.7, deberas desactivar el only\_full\_group\_by, para eso en el sql mode deberás tener las siguientes lineas (my.cnf) https://dev.mysql.com/doc/refman/5.7/en/sql-mode.html.

sql-mode = "STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION"

**Permisos de directorio**

Es posible que deba configurar algunos permisos. Los directorios dentro de storage y bootstrap/cache deben ser editables por su servidor web o Laravel no se ejecutará.

**Variables de entorno**

El siguiente paso es copiar el archivo .env.example a .env y editar las variables de configuración de acuerdo a tu servidor:

cp .env.example .env

Descripción de variables de entorno a utilizar

APP\_NAME: Nombre de la aplicación.

APP\_ENV: Entorno de ejecución.

APP\_KEY: llave de la aplicacion, se auto genera con php artisan key:generate.

APP\_DEBUG: true o false.

APP\_LOG\_LEVEL: Nivel de log (EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFO, DEBUG).

APP\_URL: URL de tu aplicación incluir http.

APP\_MAIN\_DOMAIN: Dominio de tu aplicación, incluir http.

DB\_CONNECTION: Tipo de conexión de tu Base de datos, para este proyecto por defecto se usa mysql.

DB\_HOST: host donde se aloja tu Base de Datos.

DB\_PORT: puerto por donde se esta disponiendo tu Base De Datos en el Host.

DB\_DATABASE: Nombre Base de datos (Debe estar previamente creada).

DB\_USERNAME: Usuario Base de datos.

DB\_PASSWORD: Contraseña Base de datos.

MAIL\_DRIVER: soporta ("smtp", "sendmail", "mailgun", "mandrill", "ses", "sparkpost", "log", "array").

MAIL\_HOST: Aquí puede proporcionar la dirección de host del servidor.

MAIL\_PORT: Este es el puerto utilizado por su aplicación para entregar correos electrónicos a los usuarios de la aplicación.

MAIL\_ENCRYPTION: Aquí puede especificar el protocolo de cifrado que se debe usar cuando la aplicación envía mensajes de correo electrónico.

MAIL\_USERNAME: Si su servidor requiere un nombre de usuario para la autenticación, debe configurarlo aquí.

MAIL\_PASSWORD: Si su servidor requiere una contraseña para la autenticación, debe configurarlo aquí.

ROLLBAR\_TOKEN: Token de acceso proporcionado por Rollbar.

RECAPTCHA\_SECRET\_KEY: reCaptcha secret key, proporcionado por Google.

RECAPTCHA\_SITE\_KEY: reCaptcha site key, proporcionado por Google.

BASE\_SERVICE: URL del microservicio de agendas.

CONTEXT\_SERVICE: Contexto de aplicación del servicio de agendas.

AGENDA\_APP\_KEY: Identificado de aplicación o cuenta para acceder al microservicio de agendas.

RECORDS: Cantidad de registros que se mostrarán por pagina.

TIEMPO\_CONFIRMACION\_CITA: Minutos para eliminar una cita si no ha sido confirmada.

JS\_DIAGRAM: Libreria que se va a utilizar para hacer los diagramas de flujo, default: jsplumb (Gratuita y libre uso).

MAP\_KEY: Key de acceso a Google Maps.

SCOUT\_DRIVER: driver para agregar búsquedas de texto completo a sus modelos Eloquent.

ELASTICSEARCH\_INDEX: Nombre lógico que interpretara elasticsearch como índice.

ELASTICSEARCH\_HOST: Aquí puede proporcionar la dirección de host de elasticsearch.

**Instalar las dependencias con composer**

Laravel utiliza Composer para administrar sus dependencias. Entonces, antes de usar este proyecto desarrollado en Laravel, asegúrese de tener Composer instalado en su máquina. Y ejecute el siguiente comando.

composer install

Luego, la instalación de las librerías JS necesarias:

npm install

Compilación de JS

npm run prod

Luego, Migración y Semillas de la base de datos:

php artisan migrate --seed

**Actualizaciones**

Cada vez que se realice un pull del proyecto, este deberá ser acompañado de la siguiente lista de ejecución de comandos.

npm install

npm run production

composer install

php artisan migrate --force

vendor/bin/phpunit

**Elasticsearch**

Para crear el índice:

php artisan elasticsearch:admin create

Para indexar todo (Realizar esto en instalación inicial):

php artisan elasticsearch:admin index

Para indexar solo páginas:

php artisan elasticsearch:admin index pages

**Creación de usuarios en Backend y Manager**

Para crear un usuario perteneciente al Backend, basta con ejecutar este comando especificando email y contraseña:

php artisan simple:backend {email} {password}

php artisan simple:backend mail@example.com 123456

Y para crear un usuario perteneciente al Manager,

php artisan simple:manager {user} {password}

php artisan simple:manager siturra qwerty

**Generar la llave de aplicación**

php artisan key:generate

**Tests con PHPUnit**

Listado de Tests:

* Verificar que las librerías de PHP requeridas por SIMPLE, estan habilitadas (VerifyLibrariesAvailableTest)
* Validación de Reglas Customizadas (CustomValidationRulesTest)
* Creación de Usuarios (Front, Backend, Manager) (CreateUsersTest)
* Motor de Reglas SIMPLE BPM (RulesTest)

Para ejecutar los Tests solo debes ejecutar el siguiente comando:

vendor/bin/phpunit

**Adicionales,**

Si desea poder utilizar una acción de tipo Soap, debe tener habilitada la librería Soap en su php.ini

**Configuración Clave Única**