**Guía para clonar la Aplicación**

**en Goormide**

Las diferentes herramientas que utilizamos son :

|  |  |
| --- | --- |
| Laravel 8 | Css |
| PHP 7.3 | Bootstrap |
| MySQL | Jquery |
| PhpMyadmin |  |

Antes de clonar el proyecto en un nuevo contenedor de goormide se debe crear el contenedor, para ello debemos:

* Iniciar sesion en goormide y darle a crear un contenedor.
* Lo siguiente será elegir el nombre del contenedor,

Seleccionar PHP y activamos los modulo adicionales:

* + Install MySQL
  + Enable mysql-ctl command
* Posteriormente preparamos el entorno para ello realizamos los comandos
  + sudo apt-get update
  + sudo apt-get upgrade

Estos comandos son para evitar errores por versiones que no estén actualizadas

* Descargamos el composer 2, realizando este comando:

curl -sS https://getcomposer.org/installer | sudo php -- --install-dir=/usr/local/bin --filename=composer

* Una vez obtenido esto, descargamos las extensiones que necesitemos :
  + php-xmL = sudo apt install php7.3-xml
  + php-mbstring = sudo apt-get install php7.3-mbstring
  + php-mysql = sudo apt-get install php7.3-mysqL
* Para finalizar, mediante el composer, descargamos la versión 8 de Laravel. Para realizar la descargar utilizaremos este comando:
  + composer create-project laravel/laravel="8.\*" nombre\_proyecto

Una vez preparado el contenedor, clonaremos el proyecto.

* En primer lugar ejecutaremos en una terminal del contenedor el comando :
  + git clone <dirección de github>

Probablemente nos pida nuestro usuario y contraseña de github

* Debido a que el proyecto está en desarrollo todas las configuraciones están en el fichero .env. Este fichero debe ser creado dentro del la carpeta del proyecto y como guía he creado el fichero “.env.example” que seria la configuración el fichero .env

Una vez todo esto hecho, no debería de haber ningún problema.

Para finalizar tenemos que generar los datos de la base de datos

* Generamos una clave = “ php artisan key:generate ”
* Ejecutamos las migraciones = “ php artisan migrate “
* Ejecutamos los seeders = “ php artisan db:seed ”
* Ejecutamos los factories =
  + En mi caso se utilizarían dos comandos y tendríamos que entrar dentro del tinker :
    - php artisan tinker
    - Una vez dentro ejecutamos:
      * App\Models\Medico::factory()->count(20)->create();
      * App\Models\Paciente::factory()->count(20)->create();
* En el caso de que se vaya a utilizar un entorno gráfico para la base de datos, por ejemplo phpmyadmin, adjuntare un .sql para la importación de la base de datos.

Si deseas descargar phpmyadmin simplemente tendremos que :

* + Instalar mysqi para mysql mediante este comando:

sudo apt-get install php7.3-mysql

* + A continuación instalamos phpmyadmin con este comando:

sudo apt-get install phpmyadmin

* + Posteriormente crearemos un enlace simbólico dentro del directorio public de laravel para poder acceder a phpmyadmin

ln -s /usr/share/phpmyadmin ./phpmyadmin

* + Por últimos, debemos de asegurarnos de que esté corriendo el servidor de mysql :

sudo service mysql restart

Para poner en funcionamiento finalmente el proyecto simplemente tendremos que a la hora de ejecutar el mismo, es decir cuando le damos a “ project-> run->laravel8” (así es como lo tengo yo) se nos inicie también la base de datos. Para configurar eso simplemente tendremos que darle a “ project-> run->add run command”, seleccionamos el nombre que nosotros queramos y en el apartado del script, lo editamos colocando para que coja la carpeta del proyecto y para que cada vez que le demos al run reinicie la base de datos :

php -S 0.0.0.0:${current.using.port} -t ${current.project.path}/laravel8/public & sudo service mysql restart

Para realizar pruebas, se genera mediante los Seeders algunos usuarios, por ejemplo

Usuario: [carlosfelipemartin@gmail.com](mailto:carlosfelipemartin@gmail.com)

Contraseña: 12345678