

ANEXO V: Documentación Técnica y Despliegue.

Software de gestión ganadera y análisis genético

Trabajo de Fin de Grado
INGENIERÍA INFORMÁTICA



**VNiVERSiDAD
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

Julio de 2017

Autor

Pablo Jesús Espinosa Bermejo.

Tutores

Fernando De la Prieta Pintado.

Juan Ramos González.

Elena Solera Segura.

Tabla de contenido

1.	Introducción.....	5
2.	Despliegue	5
2.1.	Requisitos	5
2.2.	Instalación de requisitos	5
2.2.1.	Apache Web Server.....	5
2.2.2.	PHP 7	5
2.2.3.	MySQL.....	6
2.2.4.	cURL.....	6
2.2.5.	Composer.....	6
2.2.6.	Git	7
2.3.	Instalación de la aplicación	7
3.	Documentación técnica del código.....	9
4.	Referencias	11

1.Introducción.

El presente anexo tiene como objetivo recoger tanto el despliegue del sistema en un servidor Debian como la documentación técnica del código.

2.Despliegue

2.1. Requisitos

Para realizar el despliegue de nuestro sistema necesitamos que el servidor donde se va a instalar cumpla una serie de requisitos:

- Servidor Web Apache v2.
- PHP en su versión 7.
- MySQL
- cURL
- Composer
- Git

2.2. Instalación de requisitos

2.2.1. Apache Web Server

Para la instalación de apache utilizamos los siguientes comandos:

```
sudo apt-get install apache2 libapache2-mod-php
```

Después de correr este comando tendremos instalado apache en nuestro servidor, y podremos acceder a los ficheros públicos de este en la ruta “/var/www/html”.

Necesitamos posibilitar que apache reescriba sobre los ficheros ya que la aplicación Laravel lo requiere, después reiniciamos el servidor apache:

```
sudo a2enmod rewrite  
sudo service apache2 restart
```

2.2.2. PHP 7

Utilizamos PHP 7 ya que las futuras versiones de Laravel lo van a requerir como mínimo. Dado que Debian aún no lo incluye en sus repositorios debemos agregarlo desde una fuente externa, para ello debemos editar el fichero “sources.list”:

```
sudo nano /etc/apt/sources.list
```

Añadimos las siguientes líneas:

```
deb http://packages.dotdeb.org jessie all
deb-src http://packages.dotdeb.org jessie all
```

Para el uso de estos repositorios debemos de obtener una clave, para ello ejecutamos los siguientes comandos:

```
wget --no-check-certificate https://www.dotdeb.org/dotdeb.gpg
sudo apt-key add dotdeb.gpg
rm dotdeb.gpg
```

Después de esto ya podremos instalar PHP 7 y los módulos necesarios gracias a los siguientes comandos:

```
sudo apt-get update
sudo apt-get install php php-xml php7.0-mbstring libapache2-
mod-php7.0 php7.0-mysql
```

2.2.3. MySQL

Para la instalación de MySQL necesitamos ejecutar el siguiente comando, durante la instalación, se nos pedirá establecer la contraseña para el usuario root:

```
sudo apt-get -y install mysql-server mysql-client
```

Una vez realizada la instalación es recomendado realizar la instalación segura e ir siguiendo los pasos mediante la ejecución de:

```
mysql_secure_installation
```

Además, después de esto debemos de crear una base de datos donde albergar nuestra aplicación:

```
mysql -u root -p
```

Nos identificamos con la contraseña establecida y creamos la nueva base de datos:

```
mysql> CREATE DATABASE TFG;
```

2.2.4. cURL

Para la instalación de Composer necesitamos tener cURL instalado, para ello debemos de utilizar el siguiente comando:

```
sudo apt-get install curl
```

2.2.5. Composer

Composer es un gestor de dependencias de PHP. Para su instalación utilizamos los siguientes comandos:

```
curl -sS https://getcomposer.org/installer | php
sudo mv composer.phar /usr/local/bin/
alias composer='/usr/local/bin/composer.phar'
```

2.2.6. Git

Para desplegar la aplicación la clonamos del repositorio público de GitHub (también se pueden copiar los ficheros manualmente por ftp u otro método), para ello necesitamos tener instalado el VCS Git, lo conseguimos de la siguiente manera:

```
sudo apt-get install git
```

2.3. Instalación de la aplicación

Para instalar la aplicación conjuntamente con Apache nos dirigimos al directorio `"/var/www/html/"` y clonamos allí el repositorio de GitHub:

```
git clone https://github.com/espipj/TFG.git
```

Copiamos los recursos de la carpeta `"TFG/Public"` a la carpeta `"/var/www/html/"`:

```
sudo cp -R -a TFG/public/. /var/www/html/
```

Debemos de editar el fichero `index.php` para que apunte a la carpeta de nuestro proyecto:

```
sudo nano /var/www/html/index.php
```

Donde reemplazaremos :

```
require __DIR__.'../bootstrap/autoload.php';
$app = require_once __DIR__.'../bootstrap/app.php';
```

Por:

```
require __DIR__.'TFG/bootstrap/autoload.php';
$app = require_once __DIR__.'TFG/bootstrap/app.php';
```

También debemos de cambiar tanto los permisos de la carpeta `"html"` y sus hijos como sus propietarios:

```
sudo chown -R usuario:www-data /var/www/html/
sudo chmod -R 775 /var/www/html
```

Debemos hacer unos pequeños ajustes en la configuración de Apache, para ello modificamos el fichero `"/etc/apache2/apache2.conf"` añadiendo las siguientes líneas:

```
DocumentRoot "/var/www/html/"
<Directory "/var/www/html/">
    AllowOverride All
</Directory>
```

Después de todo esto, podemos ejecutar la instalación de Composer en el proyecto que resolverá las dependencias, para ello:

```
cd /var/www/html/TFG
```

composer install

Tras ésto ya solo nos queda configurar el fichero de entorno de Laravel, y general la clave para la aplicación. Editamos hacemos una copia del fichero “/var/www/html/TFG/.env.example” en la misma ruta que se llame “.env” y lo editamos para obtener algo similar a lo siguiente dependiendo de nuestras preferencias:

```

1 APP_ENV=local
2 APP_KEY=3YLbCaoL6mkIKVBQZInf8cPmC2UIN8wv
3 APP_DEBUG=false
4 APP_LOG_LEVEL=debug
5 APP_URL=http://localhost
6
7 DB_CONNECTION=mysql
8 DB_HOST=localhost
9 DB_PORT=3306
10 DB_DATABASE=TFG
11 DB_USERNAME=root
12 DB_PASSWORD=contraseña
13
14 BROADCAST_DRIVER=log
15 CACHE_DRIVER=file
16 SESSION_DRIVER=file
17 QUEUE_DRIVER=sync
18
19 REDIS_HOST=127.0.0.1
20 REDIS_PASSWORD=null
21 REDIS_PORT=6379
22
23 MAIL_DRIVER=smtp
24 MAIL_HOST=smtp.gmail.com
25 MAIL_PORT=465
26 MAIL_USERNAME=correo@gmail.com
27 MAIL_PASSWORD=contrasena
28 MAIL_ENCRYPTION=ssl
29
30 PUSHER_APP_ID=
31 PUSHER_APP_KEY=3YLbCaoL6mkIKVBQZInf8cPmC2UIN8wv
32 PUSHER_APP_SECRET=

```

Los parámetros importantes a entender y configurar son:

- APP_KEY y PUSHER_APP_KEY: se generará a continuación corriendo el commando “php artisan key generate”. Debemos de comprobar que la key que

nos devuelve al ejecutarlo es la misma que está escrita en nuestro fichero. Esta key se utiliza por motivos de seguridad.

- APP_DEBUG: si queremos que muestre mensajes de error al programador o no cuando haya fallos. Se le pueden dar los valores “true/false” activado y desactivado.
- DB_CONNECTION: especificamos el tipo de base de datos que vamos a utilizar.
- DB_HOST: especificamos donde está alojada la base de datos.
- DB_PORT: puerto de escucha de la base de datos.
- DB_DATABASE: nombre de la base de datos.
- DB_USERNAME: usuario de la base de datos a utilizar.
- DB_PASSWORD: contraseña de la base de datos.
- MAIL_DRIVER: tipo de cliente de correo electrónico.
- MAIL_HOST/PORT: dominio/puerto donde está alojado el correo electrónico.
- MAIL_USERNAME: dirección de correo electrónico a usar.
- MAIL_PASSWORD: contraseña del correo electrónico.
- MAIL_ENCRYPTION: tipo de encriptación para el envío de correos electrónicos.

Después de esto nos podremos dirigir al dominio que tengamos asignado para utilizar nuestra aplicación Laravel.

3. Documentación técnica del código

Se ha realizado la documentación técnica del código del proyecto Laravel para que cualquier persona pueda consultar o seguir su desarrollo en un futuro. Es importante señalar que se ha realizado en inglés para que sea accesible al mayor público posible.

Para la generación de la documentación se ha utilizado la herramienta de generación de documentación automatizada **phpDocumentor**:

Es la herramienta de código abierto de documentación para el lenguaje PHP y que puede ser usado directamente desde la línea de comandos. Basado en el estándar PhpDoc (aun proceso de formalizarse), utiliza una sintaxis similar a JavaDoc. Puede ser usado para generar documentación de forma profesional directamente desde el código fuente de tu proyecto tanto para generar documentación en programación procedural como orientada a objetos ([1]).

Se ha utilizado este sistema de documentación sobre otras alternativas como ApiDoc por la facilidad de su instalación y uso mediante Composer.

La documentación generada puede consultarse en el directorio presente. Es importante señalar que para su mejor visualización se aconseja no usar Google Chrome ya

que hay funciones no soportadas, como por ejemplo ver el código de una determinada clase haciendo click en el botón “ver el código”. ya que Chrome no permite peticiones AJAX de forma local.

4. Referencias

- [1] phpDocumentor: About: <https://www.phpdoc.org/about>
- [2] Houston, P. (2017). The simple guide to deploy Laravel 5 application on shared hosting. [online] Medium. Available at: <https://medium.com/laravel-news/the-simple-guide-to-deploy-laravel-5-application-on-shared-hosting-1a8d0aee923e> [Accessed 22 Jun. 2017].