



José Javier Sáez González ---- 2ºDAW

**Índice.**

Título……………………………………………………………………………………………………………………….

Descripción………………………………………………………………………………………………………………

Autor……………………………………………………………………………………………………………………….

Destinatarios……………………………………………………………………………………………………………

Objetivos y Justificación------------------------------------------------------------------------------

**DOCUMENTACION DE LA APLICACIÓN**

Recogida de Información…………………………………………………………………………

Estudio de Procesos…………………………………………………………………………………

Entidad Relación………………………………………………………………………………………

Base de datos relacional…………………………………………………………………………..

Diccionario de datos………………………………………………………………………………..

Boceto del diseño…………………………………………………………………………………….

Tutorial de la herramienta………………………………………………………………………..

Manual de usuario por perfiles…………………………………………………………………

Manual del programador………………………………………………………………………….

Plan de pruebas………………………………………………………………………………………..

Recursos utilizados………………………………………………………………………………………………

Bibliografía…………………………………………………………………………………………………………..

**Título.**

Computer Incidents “Share your Problem!”

**Descripción.**

Desarrollo de una aplicación web para la gestión de incidencias y averías de los equipos informáticos de un centro educativo.

**Autor.**

José Javier Sáez González

**Destinatarios.**

El servicio ofrecido por la aplicación irá dirigido a la Escuela Virgen de Guadalupe (Badajoz) como demandante de las funcionalidades referentes a gestionar las incidencias y averías de los equipos informáticos que se puedan producir en el centro.

**Objetivos y Justificación.**

El objetivo primordial de la aplicación será que en el centro se pueda tener controlado todas las averías y fallos en los equipos informáticos (PC, monitores, impresoras, pizarras digitales, etc.) de cualquier tipo, así como tener un control más completo y exacto de que es lo que ha fallado y cuál es su reparación, sabiendo en todo momento el estado de estas.

La justificación al desarrollo de esta aplicación es la necesidad de tener un control sobre incidencias y averías de los equipos y poder tenerlas almacenadas para futuras consultas, permitiendo así también una mayor comodidad en la gestión de las mismas tanto a la hora de emitir como de recibir y gestionar por parte de Profesores y Administrador de la aplicación.

**DOCUMENTACIÓN**

**DE LA APLICACIÓN**

**Recogida de Información**

En este momento en la Escuela Virgen de Guadalupe se gestionan las incidencias y averías de los equipos informáticos sin realizar ninguna informatización de los datos referentes, hemos recogido información sobre cómo se realiza este proceso por el momento en el centro y es la siguiente:

* Para empezar es necesario recopilar toda la información de los equipos informáticos que se encuentran en el centro (sistema operativo, nombre del equipo, Dirección Ip, marca, modelo, procesador,…) al igual que datos identificativos sobre demás Gadgets (monitores, impresoras,…).
* Al empezar el curso se elabora una codificación que hace referencia a la ubicación de los equipos que se hayan añadido nuevos, el resto ya estarán codificados de la misma forma, atendiendo al lugar donde se sitúe (aula, despacho, sala de profesores).
* Las incidencias o averías en los equipos son comunicadas por él profesor que se encuentre en ese momento en la ubicación del mismo de forma verbal o por un correo electrónico.
* Estas incidencias son recogidas por el encargado del mantenimiento de los equipos en este caso el informático de la escuela que lleva a cabo una solución de las mismas.

Los recursos utilizados son:

* Información de los dispositivos que se encuentran en cada ubicación y cuál es la codificación asignada para los mismos.
* Información de los profesores que obtendremos de rayuela en documento de tipo Excel.

**Estudio de Procesos**

Proceso 0. Instalación de la aplicación, creación de usuarios y la Base de Datos.

El usuario Administrador se asignará en un principio al responsable de la zona de informática, encargado del mantenimiento. Realizará tareas de obtención de datos, importación de estos y gestión de incidencias (proceso).

Los usuarios Profesores estarán asignados a cada uno de los profesores del centro y se encargarán de comunicar a través de la aplicación las incidencias o averías en los diferentes equipos informáticos (proceso).

1.- Acceso y Gestión de Usuarios

* Se accederá a la aplicación mediante un formulario de identificación de usuarios, que será la página inicial, nadie tendrá acceso a la aplicación si no está registrado.
* La primera vez que tanto el usuario “Admin” como los usuarios identificados como “Profesores” accedan con sus credenciales proporcionadas tendrán que cambiar su contraseña por seguridad.

2.- Almacenamiento de Datos

* El administrador se encarga de recoger de rayuela la información de los profesores registrados en ese año lectivo (proporcionados en Excel).
* Al acceder a la aplicación se tendrán que guardar primero, los datos de los profesores y en segundo lugar las ubicaciones donde se encuentran los equipos informáticos que se recogerán en la aplicación.

1. Importar Profesores: al seleccionar esta opción podremos elegir el archivo de tipo “.xls” que contiene los datos de los profesores y estos quedarán guardados.
2. Registrar Ubicaciones: al seleccionar esta opción veremos un formulario de entrada de datos que habrá que rellenar de forma manual.

* Posteriormente se recogerá el número de equipos que se encuentran presentes en cada ubicación, y se almacenarán los datos correspondientes a cada uno de ellos (dirección ip, memoria, etc.), se le aplicará a cada uno de ellos la codificación establecida que depende de la localización de los mismos.

3.- Mantenimiento Ubicaciones

* Tendremos un proceso en el que se muestren todas las ubicaciones del colegio dando la opción en cada una de ellas de modificar y elimina. También se tendrá la opción de añadir una nueva ubicación.

1. Alta Ubicación: al seleccionarla nos llevará a una página donde se selecciona el tipo de ubicación y se introduce el nº de piso y ubicación.
2. Modificar Sección: al seleccionarla nos llevará a una página donde se indicará la ubicación a modificar y un formulario para la entrada para modificar el nº de piso y ubicación.
3. Borrar Ubicación: mostrará un mensaje de confirmación o error.

4.- Mantenimiento Profesores

* Tendremos un proceso que mostrará todos los profesores y teniendo en cuenta las posibilidades de que un profesor cause baje temporal, se jubile o se incorporé uno nuevo tendremos 3 opciones en cada profesor que nos permitirán modificar, borrar y añadir:

1. Alta Profesor: al seleccionarla nos llevará a una página donde se introducirán todos los datos.
2. Modificar Profesor: al seleccionarla nos llevará a una página donde se indicará el profesor a modificar.
3. Borrar Profesor: Mostrará un mensaje.

5.- Mantenimiento Equipos

* Este proceso mostrará todos los equipos clasificados por su ubicación y teniendo en cuenta las posibilidades de que un equipo sea sustituido, se le cambien componentes o se incorporé un nuevo a una ubicación tendremos 2 opciones en cada equipo que nos permitirá modificar y borrar, en la misma página tendremos la opción de añadir.

1. Alta Equipo: al seleccionar esta opción nos llevará a una página donde se introducirán los datos referentes al mismo.
2. Modificar Equipo: nos llevará a una página donde se mostrará el equipo y un formulario de entrada de datos donde se podrán modificar los mismos.
3. Borrar Equipo: mostrará un mensaje de confirmación o error.

6.- Emisión de Incidencias

* En este proceso los encargados de la emisión de Incidencias, en este caso los usuarios registrados como Profesores tendrán una pantalla donde se muestren todas las Ubicaciones, al seleccionar una de estas se obtendrá el listado de los equipos que se encuentran en ella. En este momento se podrá elegir uno de los Equipos para proceder con el formulario de registro de datos correspondientes a la Incidencia, en el que se mostrarán Dirección Ip, Marca, Modelo, 4 seleccionables de Averías comunes y un espacio para introducir Descripción.

Una vez completado el formulario se podrá Enviar la Incidencia, añadiéndose a estos datos se almacenarán también Fecha y Hora actuales del equipo desde donde se realice.

7.- Modificación de Incidencias

* Una vez se tengan almacenadas Incidencias cada usuario Profesor contará con la posibilidad de obtener un listado con todas sus Incidencias emitidas en cada una de ellas tendremos las opción de Revisar.

1. Revisar: al seleccionar esta opción se nos mostrará el formulario de registro de datos con la posibilidad de modificar los Seleccionables y la Descripción de la Avería.

8.- Gestión de Incidencias

* Se contará con un proceso en el que usuario Administrador obtendrá todas las Incidencias clasificadas por su Estado (Completadas, En Curso, No Revisadas) y la Ubicación del Equipo que la conlleva.

En cada una de las Incidencias tendremos la opción de Ver la misma completa, al seleccionar esta opción veremos Dirección Ip, Marca, Modelo y Avería. Junto con esto se tendrá un seleccionable para clasificar el Estado de la Incidencia.

Una vez completado el formulario se podrá guardar y quedará agrupada por su Estado, teniendo la opción en todas de Modificar para cambiar el Estado y que el usuario Profesor pueda seguir el progreso de la Incidencia.

9.- Listado de Incidencias

* Teniendo en cuenta que se van a tener Incidencias que ya estén revisadas y solucionadas e Incidencias pendientes se va a tener un proceso que nos permitirá obtener mediante un archivo “.pdf” un listado de todas aquellas que se encuentran en este estado.

10.- Gestión de Usuarios

* Tendremos un proceso para la gestión de Usuarios. Cada usuario Profesor podrá acceder a la información de su perfil para gestionar el cambio de Password y otros datos como su Correo Electrónico.

El usuario Administrador podrá acceder a su información y podrá Modificar su Usuario, Password y Correo Electrónico, al igual podrá Modificar toda la información de los usuarios Profesores.

**Entidad-Relación.**



**Base de datos Relacional.**



**Diccionario de Datos.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **USERS** | | | | | |
| **Campo** | **Tipo de Dato** | **Tamaño** | **Descripción** | **Restricción** | **Propiedades** |
| **id** | Tinyint | 4 | Identificador Usuario | **Clave Principal** | Not Null, AI\* |
| **DNI** | Char | 9 | Dni Usuario | **Unique** | Not Null |
| **User** | Varchar | 10 | Ident. Usuario |  | Not Null |
| **Password** | Varchar | 100 | Contraseña |  | Not Null |
| **Nombre** | Varchar | 50 | Nombre completo Usuario |  | Not Null |
| **Tipo** | ENUM | 1 | Identificador tipo de Usuario | CHECK (A,P) | Not Null |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UBICACIÓN** | | | | | |
| **Campo** | **Tipo de Dato** | **Tamaño** | **Descripción** | **Restricción** | **Propiedades** |
| **Id** | Tinyint | 4 | Identificador Ubicación | **Clave Principal** | Not Null, AI |
| **Cod\_Ubicación** | Char | 4 | Código de Ubicación | **Unique** | Not Null |
| **Tipo** | Varchar | 30 | Tipo de Ubicación (Aula,..) |  | Not Null |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DISPOSITIVO** | | | | | |
| **Campo** | **Tipo de Dato** | **Tamaño** | **Descripción** | **Restricción** | **Propiedades** |
| **id** | Tinyint | 4 | Identificador Dispositivo | **Clave Principal** | Not Null, AI |
| **Direccion\_Ip** | Char | 15 | Dirección Ip | **Unique** | Admite Null |
| **Marca** | Varchar | 30 | Marca |  | Not Null |
| **Modelo** | Varchar | 50 | Modelo |  | Not Null |
| **id\_Ubicación** | Tinyint | 4 | Identificador Ubicación | **Clave Foránea Ubicaciones** | Not Null |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EQUIPO** | | | | | |
| **Campo** | **Tipo de Dato** | **Tamaño** | **Descripción** | **Restricción** | **Propiedades** |
| **Id** | Tinyint | 4 | Identificador\_Dispositivo | **Clave Principal** | Not Null,AI |
| **Id\_dispositivo** | Tinyint | 4 | Identificador Dispositivo | **Unique** | Not Null |
| **Memoria** | Varchar | 50 | Modelo Memoria |  | Not Null |
| **Procesador** | Varchar | 50 | Modelo Procesador |  | Not Null |
| **Tarjeta\_Grafica** | Varchar | 50 | Modelo Tarj\_Grafica |  | Not Null |
| **Tarjeta\_Red** | Varchar | 50 | Modelo Tarj\_Red |  | Not Null |
| **Fuent\_Alimentacion** | Varchar | 50 | Fuente de Alimentación |  | Not Null |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **GADGET** | | | | | |
| **Campo** | **Tipo de Dato** | **Tamaño** | **Descripción** | **Restricción** | **Propiedades** |
| **id** | Tinyint | 4 | Identificador Dispositivo | **Clave Principal** | Not Null,AI |
| **Id\_dispositivo** | Tinyint | 4 | Identificador Dispositivo | **Unique** | Not Null |
| **Tipo** | Varchar | 30 | Tipo de Gadget (Impresora,..) |  | Not Null |

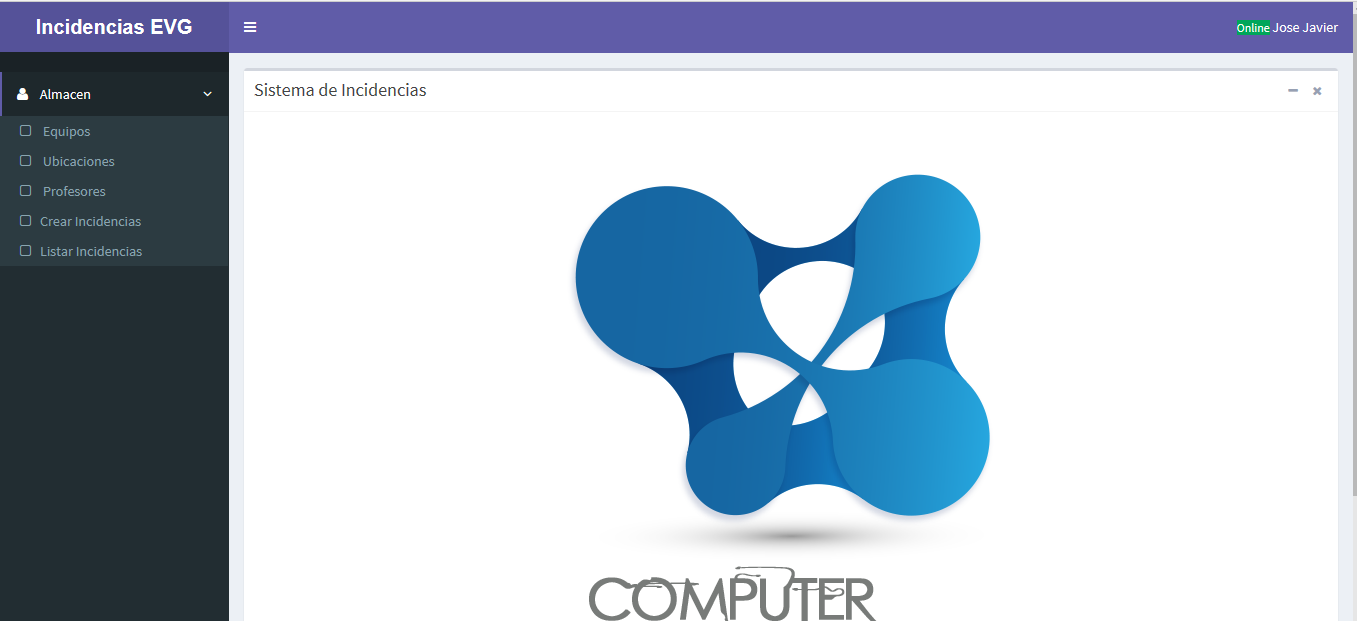
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ESTADO** | | | | | |
| **Campo** | **Tipo de Dato** | **Tamaño** | **Descripción** | **Restricción** | **Propiedades** |
| **id** | Tinyint | 4 | Identificador Estado | **Clave Principal** | Not Null,AI |
| **Descripcion** | Varchar | 50 | Descripción Estado |  | Not Null |

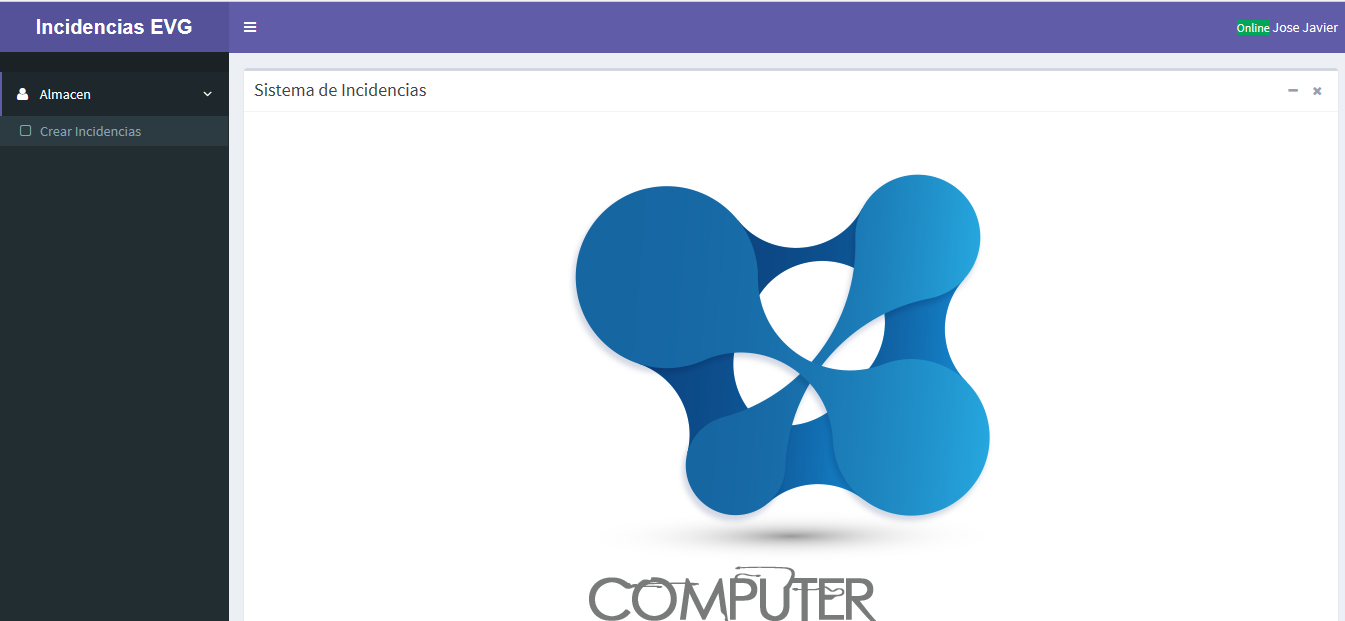
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **INCIDENCIAS** | | | | | |
| **Campo** | **Tipo de Dato** | **Tamaño** | **Descripción** | **Restricción** | **Propiedades** |
| **id** | Tinyint | 4 | Identificador Incidencia | **Clave Principal** | Not Null,AI |
| **Hora** | Time | - | Hora Incidencia |  | Not Null |
| **Fecha** | Date | - | Fecha Incidencia |  | Not Null |
| **Observaciones** | Varchar | 50 | Observaciones |  | Not Null |
| **Vista** | Enum | - | Indicador Vista | CHECK (‘si’,’no’) | Not Null |
| **Id\_dispositivo** | Tinyint | 4 | Identificador Dispositivo | **Clave Foránea Dispositivo** | Not Null |
| **Id\_usuario** | Tinyint | 4 | Identificador Usuario | **Clave Foránea Usuario** | Not Null |
| **Estado\*** | Tinyint | 4 | Identificador Estado | **Clave Foránea Estado** | Not Null |

\***Estado**: tiene un valor predeterminado = 1;

\***AI** = Auto\_Increment;

**Boceto del Diseño**

**Boceto Administrador**

**** **Boceto Profesor**

**Tutorial de la Herramienta Laravel**

**¿Qué es Laravel?**

Laravel es un framework de código abierto cuyo enfoque es el desarrollo de aplicaciones y servicios web con una versión de PHP igual o superior a 5.

Cuenta con una comunidad llena de energía, documentación atractiva de contenido claro y completo; y, además, ofrece las funcionalidades necesarias para **desarrollar aplicaciones modernas de manera fácil y segura.**

Está hecho para **arquitectura MVC** (Modelo Vista Controlador) y resuelve necesidades actuales como **manejo de eventos y autenticación de usuarios**. Además, cuenta con un código modular y extensible por medio de un administrador de paquetes y un soporte robusto para manejo de bases de datos.

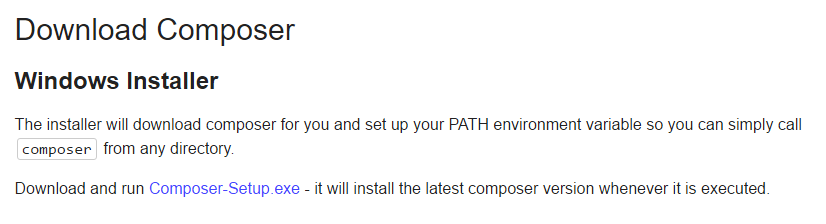
**Instalación de Laravel**

Para hacer una correcta instalación del framework Laravel es necesario tener previamente instalado la distribución de Apache, Xampp.

1.- Instalación de Composer

Composer es una herramienta para gestionar las dependencias en PHP. Te permite declarar las librerías de las cuales tu proyecto depende o necesita y las instala en el proyecto por ti.

Para instalar Composer en Windows debemos descargarlo de su página oficial: (<https://getcomposer.org/download/>)



Una vez instalado Composer para comprobar que la instalación se ha realizado correctamente abriremos símbolo de sistema (**cmd**) y teclearemos en el mismo:

**php –v** (ver versión de php)

**composer –version** (ver versión de Composer)

2.- Instalación y Creación del proyecto Laravel

Accedemos a símbolo de sistema (**cmd**) y una vez abierto en nuestro terminal accederemos a la ruta donde vamos a crear nuestro proyecto. Por defecto esta ruta va a ser (C: /xampp/htdocs), por lo tanto:

**cd /**

**cd xampp/htdocs**

Una vez ubicados en esta ruta procederemos a crear nuestro proyecto con la última versión disponible de dos maneras:

1.- Vía el Instalador Laravel -> **composer global require “laravel/installer=~1.1”**

Una vez instalado, el simple comando **laravel new nombre\_del\_proyecto** creará el proyecto.

2.- Vía Composer create-project -> **composer create-project laravel/laravel nombre\_del\_proyecto --prefer-dist** (esto requiere un poco de tiempo)

Con esto tendremos creado nuestro proyecto actualizado, para comprobar que la instalación se ha realizado correctamente dentro de nuestro terminal accederemos a la ruta.

**cd nombre\_del\_proyecto**

**php artisan serve**

Este último comando nos iniciará nuestro proyecto en el puerto 8000, finalmente si todo ha ido bien podremos verificar que la creación se realizó de manera correcta accediendo desde nuestro navegador a:

**localhost:8000/**

****

**Configuración Básica de Laravel**

Todos los archivos de configuración del framework de Laravel se pueden localizar en la carpeta **app/config**. Cada opción está documentada. Siéntete libre para observar a través de los archivos y familiarizarte con todas las opciones disponibles.

Permisos del Directorio

Después de instalar Laravel, puede que tengamos que configurar algunos permisos. Los directorios dentro **storage** y de **bootstrap/cache** deben ser escritos por su servidor web. Si está utilizando la máquina virtual Homestead, estos permisos ya deben estar establecidos.

Clave de la Aplicación

Lo siguiente que debe hacer después de instalar es configurar la clave de la aplicación. Si se instaló Laravel a través de Composer o el instalador de Laravel, esta clave ya se ha establecido mediante **key:generate.** La clave puede establecerse en el archivo de entorno **.env**. Si no se a renombrado el archivo **.env.example** a **.env** debe hacerlo ahora.

Configuración Adicional

Laravel casi no necesita configuración para comenzar a trabajar. Sin embargo, es recomendable revisar el archivo **app/config/app.php** y su documentación ya que contiene varias opciones como **timezone** (zona horaria) y **locale** (localización) que puedes cambiar de acuerdo a tu aplicación.

**¿Qué es Homestead?**

Laravel Homestead es el box de Vagrant oficial para Laravel, en sí es una herramienta de desarrollo, la cual nos brinda un ambiente lo más cercano posible a un entorno de producción solo que en nuestra propia máquina local, esto con la finalidad de que al estar desarrollando nuestras aplicaciones podamos estar probándolo constantemente con la máquina virtual (que será como nuestro servidor) y así poder estar seguros de que cuando vayamos a hacer el Deploy a producción no tengamos ningún problema.

Laravel Homestead, a grandes rasgos, es una máquina virtual con un Ubuntu Server y tiene instaladas varias herramientas out of the box, es decir, ya están pre-cargadas para no preocuparnos por instalar software que probablemente vayamos a utilizar.

**Instalación Homestead Laravel**

1.- Requisitos Previos

Dado que estaremos haciendo la instalación para Windows, debemos de habilitar la virtualización por hardware (VT-x), esto varía mucho debido a que esto se debe de habilitar en el BIOS.

Lo siguiente que se debe tomar en cuenta es no utilizar la consola de Windows, esto debido a que no tiene soporte nativo para la generación de llaves SSH las cuales vamos a necesitar para conectarnos a nuestro servidor virtual, en su lugar utilizo la consola de Git. (<https://git-scm.com/download>)

Después de tener instalado composer en nuestro equipo necesitamos instalar **VirtualBox** (<https://www.virtualbox.org/>) o Vmware (<http://www.vmware.com/?ClickID=dkxrm22hc2mnrntocwsotrhrbxhzcocynhks>) para poder correr nuestra máquina virtual.

Lo siguiente que debemos de hacer es tener **Vagrant** (<https://www.vagrantup.com/>) instalado en nuestro equipo, esto debido a que Homestead es una box de Vagrant, es decir, todo lo que necesitamos para tener nuestro entorno de desarrollo listo para empezar a trabajar con él. Una vez instalado Vagrant debemos agregar el box, lo cual se hace desde la consola (**cmd**) con el siguiente comando:

**vagrant box add laravel/homestead**

y cuando nos pregunte que provider vamos a utilizar seleccionamos Virtualbox o Vmware dependiendo de nuestra máquina virtual.

Por último debemos de tener un cliente SSH para poder conectarnos a nuestra máquina virtual. Dado que Windows no tiene soporte nativo para conexiones de este tipo nosotros utilizaremos **PuTTY** un cliente SSH muy ligero.

2.- Instalación Homestead

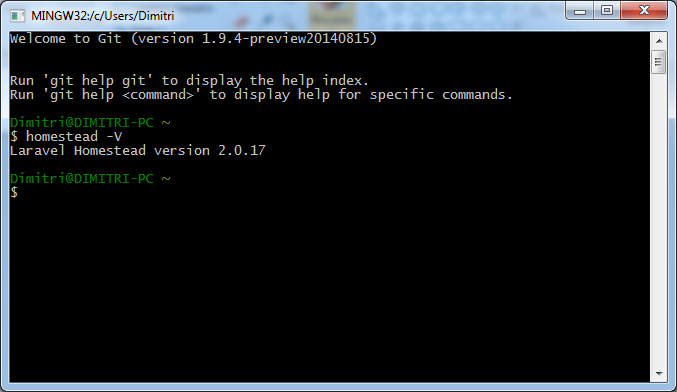
Una vez que tengamos todos los requerimientos podemos empezar con la instalación de Laravel Homestead, desde nuestra consola de **GIT** tecleamos el comando:

**composer global require “laravel/homestead=~2.0”**

Para verificar que la instalación se ha realizado correctamente vamos a abrir la consola de **GIT** y tecleamos:

**homestead -V**

Con esto podemos verificar que nuestra ruta se haya configurado correctamente y podremos ver la versión de Homestead que se ha instalado en nuestro sistema



Ya que estamos en la consola de **GIT**, aprovechemos de una vez para generar nuestras llaves SSH si es que no lo hemos hecho con anterioridad, para esto vamos a teclear el comando:

**ssh-keygen –t rsa –C “direccion@correo”**

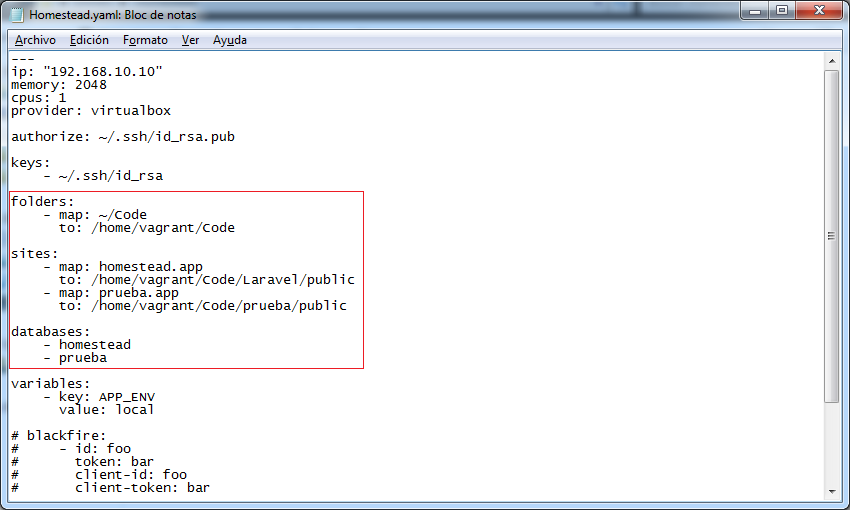
Ya con nuestra llave lista podemos proceder a correr el comando para iniciar Homestead, esto es importante para que se genere nuestro archivo de configuración, para esto tecleamos:

**homestead init**

Después tecleamos

**homestead edit**

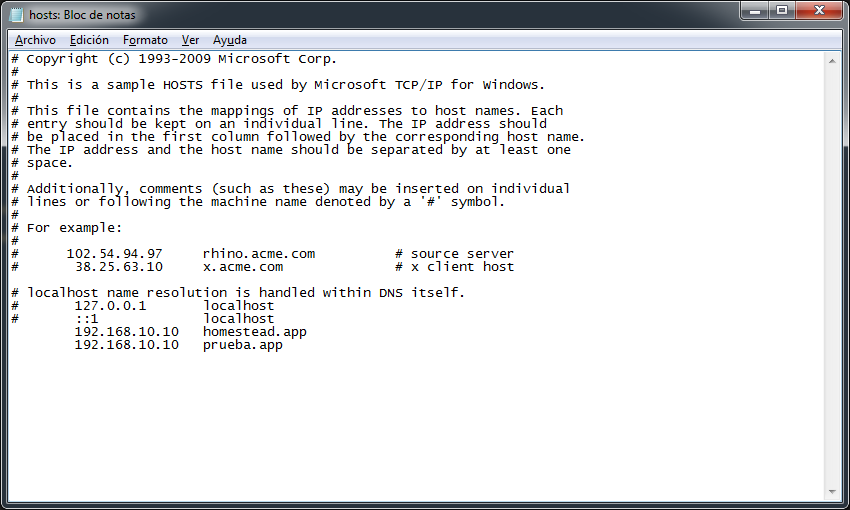
Para poder configurar nuestro entorno, al hacer esto nos aparecerá un bloc de notas con las configuraciones por default, aquí solo debemos agregar o modificar los parámetros que necesitemos, tales como agregar o quitar proyectos, incrementar o decrementar la memoria ram, indicar la ruta de nuestras llaves SSH, etc.



Debemos tener una carpeta llamada Code en nuestro home de Windows, así que debemos crearla tecleando en la consola de **GIT**:

**mkdir ~/Code**

Al igual debemos modificar nuestro archivo de hosts ubicado en C:\Windows\System32\drivers\etc y debe estar como se muestra a continuación:



3.- Correr Homestead

Una vez que tengamos todo listo solo debemos teclear:

**homestead up**

y automáticamente se empezará a cargar nuestra máquina virtual con homestead

4.- Importante

Una vez que se termine de cargar la máquina virtual, lo más común es que intenten entrar a ella con el comando:

**homestead ssh**

pero esto en windows no es así dado que puede ocasionar que la consola muestre símbolos raros y no podamos ver nada, aquí es donde entra **PuTTY** al rescate, al abrir **PuTTY** nosotros podemos acceder por SSH a nuestro servidor sin problemas, solo debemos indicarle la dirección IP de la máquina virtual y listo, nos permitirá entrar a ella, al hacerlo nos pedirá usuario y password, para ambos es **vagrant**

Una vez dentro de homestead tenemos todo el servidor de **Ubuntu** a nuestra disposición para crear, eliminar y trabajar sobre proyectos sin restricción alguna, vamos a crear nuestro proyecto de prueba, para esto primero vamos a la carpeta que configuramos para el proyecto, la cual es **/home/vagrant/Code/prueba/** así que para esto primero vamos a la carpeta Code tecleando:

**cd /home/vagrant/code/**

una vez que estemos ahí podemos crear nuestro proyecto de la manera habitual:

**composer create-project laravel/laravel prueba**

Por último para detener la máquina virtual solo deben teclear:

**homestead halt**

esto apagará la máquina virtual sin afectar los proyectos en los que estemos trabajando.

**Manual de Usuario de cada perfil**

**Administrador**

* Importar: desde este apartado podremos importar los datos de los profesores recogidos en formato Excel “.csv”.
* Ubicaciones: se listan las ubicaciones registradas y cuenta con tres opciones Alta, Editar y Eliminar.
* Equipos: se listan los equipos registrados y se cuenta con tres opciones Alta, Editar y Eliminar.
* Profesores: se listan los usuarios profesores registrados y se cuenta con tres opciones Alta, Editar y Eliminar.
* Incidencias: desde este apartado podremos Enviar Incidencias o Ver Incidencias.

Enviar Incidencias: Se selecciona la Ubicación y el Equipo que procede y se añadirá una descripción del problema.

Ver Incidencias: Se podrán filtrar las Incidencias por su Estado y se mostrarán datos relevantes de la misma al igual se podrá modificar el Estado para poder hacer un seguimiento por parte de todos los Usuarios.

Listado Incidencias: Al igual se podrán filtrar por su Estado y obtener a través de un botón en la interfaz este listado en archivo de tipo PDF (.pdf)

* Gestión de Usuario: se muestran todos los usuarios registrados para cada uno de ellos tendremos la opción de Editar y Eliminar, al igual tendremos la opción Alta para añadir un nuevo Usuario profesor.

**Profesores**

* Incidencias: desde este apartado podremos Enviar Incidencias.

Enviar Incidencias: Se selecciona la Ubicación y el Equipo que procede y se añadirá una descripción del problema.

* Gestión de Usuario: se muestran todos los datos del profesor autenticado y se dará la posibilidad de modificar los datos Dni, Usuario, Password y Nombre.

**Manual del Programador**

**Administrador**

* Login: validamos que exista el Usuario y que la Contraseña coincida con la almacenada y relacionada con dicho usuario.
* Ubicaciones:

- Se valida que los campos sean requeridos

- Se valida que el campo Código Ubicación no tenga más de 4 caracteres y que no coincida con ninguno de los registrados.

- Se valida que el campo Tipo no sea numérico.

* Equipos:

- Se valida que los campos sean requeridos

- Se valida que el campo Dirección Ip no tenga más de 15 caracteres y que no coincida con ninguno de los registrados.

- Se valida que los campos Marca, Modelo, Memoria, Procesador, Tarjeta Gráfica, Tarjeta Red y Fuente de Alimentación no contengan mayor número de caracteres de los que están permitidos.

* Profesores:

- Se valida que los campos sean requeridos.

- Se valida que el campo DNI no tenga más de 9 caracteres y que no coincida con ninguno de los registrados al igual que tenga un formato Dni válido.

- Se valida que los campos Password y Confirmar Password coincidan y que tengan una longitud mayor a 6 caracteres.

* Incidencias:

- Se valida que tenga que existir un dispositivo registrado si no es así habrá que añadirlo.

- Se valida que los campos sean requeridos y que tengan la longitud máxima permitida.

**Profesores**

* Login: validamos que exista el Usuario y que la Contraseña coincida con la almacenada y relacionada con dicho usuario.

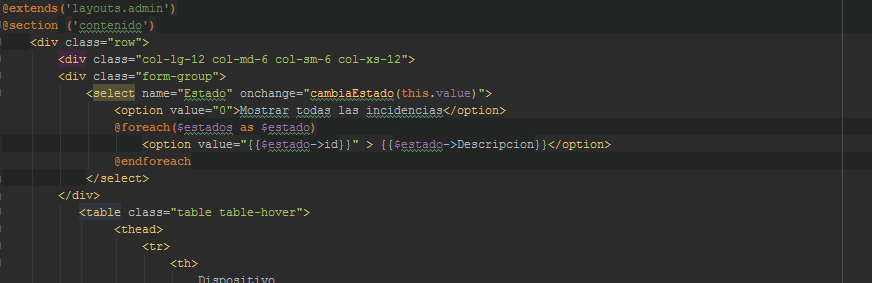
No podrá acceder a ningún sitio salvo a Enviar Incidencias, Página Principal y Gestión de su propio usuario.

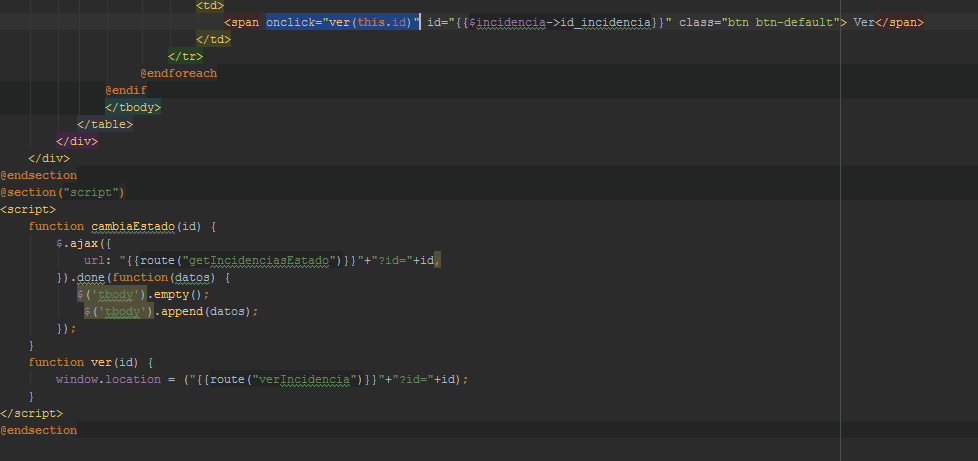
* Incidencias:

- Se valida que tenga que existir un dispositivo registrado si no es así habrá que añadirlo.

- Se valida que los campos sean requeridos y que tengan la longitud máxima permitida.

**Funciones más relevantes**

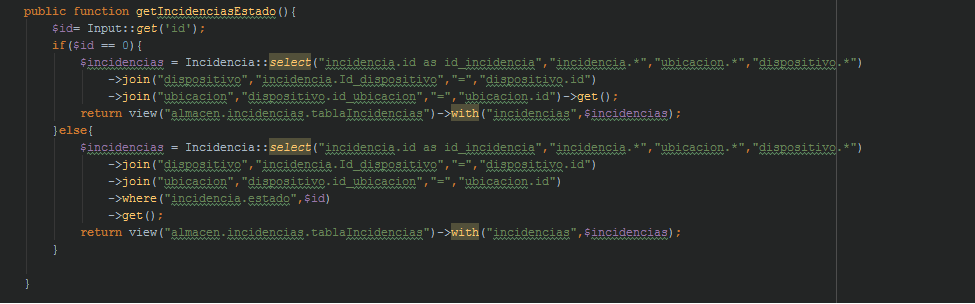




Con estos fragmentos de código dentro la vista **listar.blade.php** haremos una vista dinámica de las Incidencias clasificándolas por el tipo de Estado.

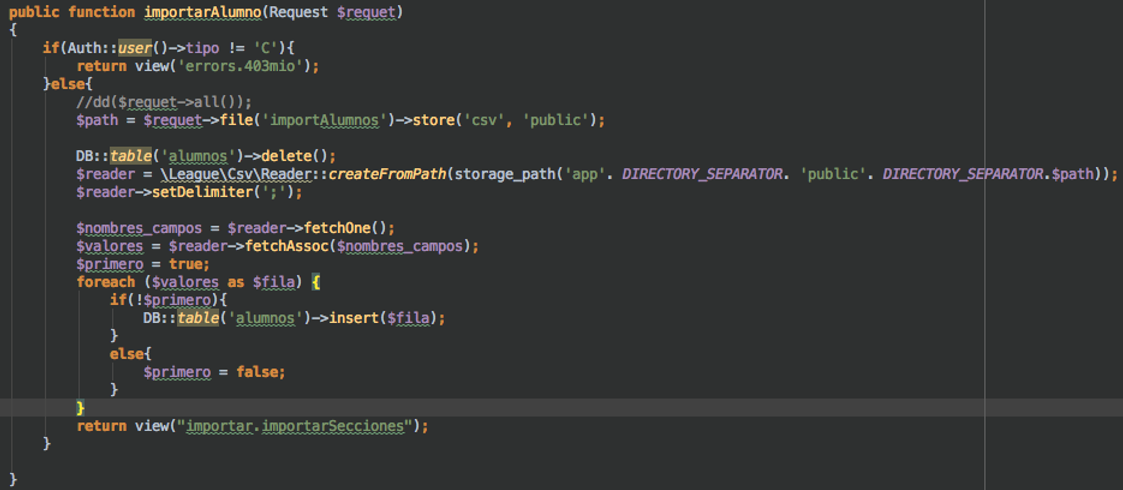
Se podrán mostrar:

* Todas las Incidencias
* Incidencias filtradas por Estado (Enviada, Recibida, En Trámite, Solucionada)
* Todo esto teniendo en cuenta que esta vista solo estará disponible para el usuario **Administrador**.



Con este fragmento de código que se encuentra en **IncidenciaController.php** obtendremos el Estado asociado a la Incidencia.

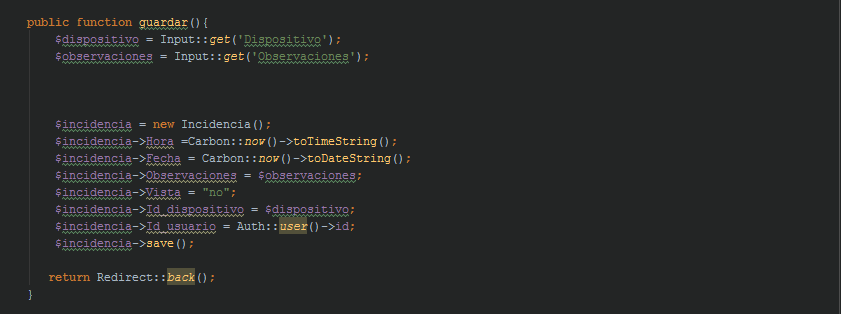
* Sacamos todas las Incidencias con su Estado asociado
* El acceso a estas funciones se hará solo por parte del Administrador
* Devolvemos a la vista **tablaIncidencias.blade.php** todos los datos obtenidos



Esta función es la encargada de importar los profesores a nuestra Base de Datos

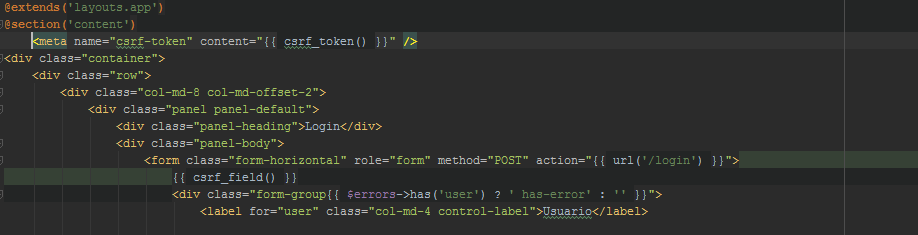
Si se produce algún error nos devuelve a la vista **ImportarProfesor.blade.php** si no es así:

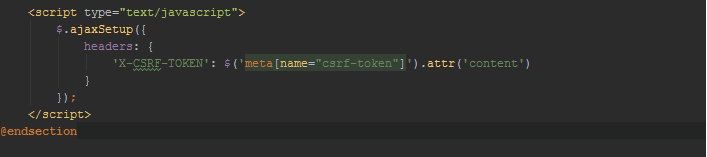
* El código que se encuentra después de la cláusula **else** lo que hace es recoger el archivo que se introduce en el **import type=”File”**, borra toda la tabla a la que hace referencia, se coge bien la ruta donde se encuentra la librería y con lo que vamos a leer el archivo
* Sacamos los nombres de los campos que serán la primera línea de nuestro archivo, posteriormente se obtiene los valores por cada campo.
* Esta opción de importar Profesores solo estará disponible para el usuario Administrador.



Con este fragmento del código dentro de **IncidenciaController.php** se realiza el guardado de los datos de Incidencias dentro de nuestra Base de Datos.

Obtenemos Hora y Fecha del sistema y Usuario autenticado al igual que las Observaciones de la Incidencia y establecemos el campo **Vista = ‘no’** para saber cuáles son las que ya se han visto por el Administrador y que el usuario Profesor pueda tener un control dinámico del proceso.





**Csrf-token:** Al introducir esto en nuestro código hacemos referencia a la generación de tokens específicos para cada tipo de acción realizable en la aplicación y el registro de estos tokens en la sesión del usuario. La función genera un token de forma aleatoria y guarda en sesión, asociado al nombre del formulario recibido, el token generado y la fecha de generación.

**Csrf\_field:** Introduciendo esta sentencia en nuestro código que contenga etiquetas **<form>** protegemos nuestro formulario y el posterior envío de los datos contenidos en el mismo contra ataques CSRF.

**Archivos Include**

Librería para la importación de los archivos “.csv”

* <http://csv.thephpleague.com/>

Librería para las fuentes utilizadas en la aplicación (Font Awesome)

* <http://fontawesome.io/>

Librería para la plantilla AdminLte utilizada en la aplicación

* <https://almsaeedstudio.com/themes/AdminLTE/documentation/index.html>

**Plan de Pruebas**

* Login:

1. Comprobamos que en la pantalla de autenticación de usuarios si introducimos un usuario no válido se muestra un mensaje de error.
2. Comprobamos que solo pasan de la pantalla de autenticación los usuarios registrados.

* Equipos:

1. Comprobamos que se listan todos los equipos que se encuentran registrados.
2. Alta: al proceder al alta de usuarios comprobamos que la dirección ip no puede repetirse es decir si introducimos una dirección ip ya registrada nos mostrara un mensaje de error
3. Editar: al editar usuarios comprobaremos que todos los campos del formulario están rellenos si esto no es así nos devolverá un mensaje de error, al igual este mensaje aparecerá si alguno de los campos no cumple con los requisitos de validación.
4. Eliminar: al eliminar un usuario se mostrará un mensaje de que se ha borrado satisfactoriamente si el proceso no se ejecuta correctamente aparecerá un mensaje de error.

* Ubicaciones:

1. Comprobamos que se listan todas las ubicaciones que se encuentran registradas.
2. Alta: al proceder al alta de ubicaciones comprobamos que el Código Ubicación no puede repetirse es decir si introducimos un Código ya registrado nos mostrara un mensaje de error
3. Editar: al editar ubicaciones comprobaremos que todos los campos del formulario están rellenos si esto no es así nos devolverá un mensaje de error, al igual este mensaje aparecerá si alguno de los campos no cumple con los requisitos de validación.
4. Eliminar: al eliminar una ubicación se mostrará un mensaje de que se ha borrado satisfactoriamente si el proceso no se ejecuta correctamente aparecerá un mensaje de error.

* Profesores:

1. Comprobamos que se listan todos los profesores que se encuentran registrados.
2. Alta: al proceder al alta de profesores comprobamos que el Dni no puede repetirse es decir si introducimos un Dni ya registrado nos mostrara un mensaje de error
3. Editar: al editar profesores comprobaremos que todos los campos del formulario están rellenos si esto no es así nos devolverá un mensaje de error, al igual este mensaje aparecerá si alguno de los campos no cumple con los requisitos de validación.
4. Eliminar: al eliminar un profesor se mostrará un mensaje de que se ha borrado satisfactoriamente si el proceso no se ejecuta correctamente aparecerá un mensaje de error.

* Incidencias:

1. Comprobamos que se listan todas las incidencias que se encuentran registradas.
2. Enviar: al proceder a enviar incidencias de ubicaciones el Código Ubicación debe coincidir con el su equipo relacionado, al igual la descripción debe estar rellena de no ser así mostrará un mensaje de error pidiendo que se rellene este campo.
3. Listar: al listar incidencias si no se han registrado incidencias por el momento nos encontraremos con un mensaje de error.

El filtrado de incidencias debe mostrar solo las incidencias que contienen ese tipo de estado, si no hay ninguna con el estado seleccionado no se mostrarán.

1. Obtener Pdf: en este apartado si no se han registrado todavía incidencias mostrará un mensaje de error.

En el mismo si el filtrado de incidencias no hace referencia a ningún estado de las mismas no se mostrará ninguna.

**Recursos Utilizados**

**Framework Laravel**

**Bootstrap**

**AdminLte**

**FontAwesome**

**DataTable**

**IDE PHPStorm**

**IDE SublimeText 3**

**Bibliografía**

Recopilación de fuentes documentales que se revisan, así como direcciones web consultadas.

* Curso Laravel 5.3

<https://www.youtube.com/user/jcarlosad7/videos>

* Direcciones Web Consultadas

<https://scotch.io/tag/laravel>

<http://laraveles.com/docs/5.1/>

<http://es.stackoverflow.com/questions/tagged/laravel>

<http://dev.tutorialspoint.com/laravel/index.htm>

Enlace web utilizado para la descarga de recursos necesario en la implementación de la solución.

* Instalación de Xampp

<https://www.apachefriends.org/es/download.html>

* Instalación Composer y Laravel

<https://styde.net/instalacion-de-composer-y-laravel-5-1/>

* FontAwesome

<http://fontawesome.io/>

* DataTables

<https://www.datatables.net/>

* AdminLte

<https://github.com/almasaeed2010/AdminLTE/>

* PhpStorm

<https://www.jetbrains.com/phpstorm/>

* SublimeText

<https://www.sublimetext.com/>