Yolanda Morales Zamora
20/03/2023
DEWS
IES Punta del Verde. Sevilla

TAREA LARAVEL

A. Pasos para la instalación del framework. LARAVEL.

¿Qué es Laravel?

Laravel es un framework de desarrollo de aplicaciones web PHP de código abierto, que se utiliza para crear aplicaciones web robustas, escalables y seguras.

Desarrollado por Taylor Otwell en 2011 y desde entonces ha crecido en popularidad debido a su facilidad de uso, modularidad y amplia gama de características avanzadas.

Laravel utiliza el patrón de diseño MVC (Modelo Vista Controlador) para separar la lógica de presentación de la lógica de negocio y de acceso a los datos, lo que hace que sea fácil de mantener y escalar. Además, Laravel cuenta con un conjunto completo de características como rutas, controladores, vistas, migraciones de bases de datos, autenticación, envío de correos electrónicos, colas, programación de tareas y más, lo que permite a los desarrolladores construir aplicaciones web de alta calidad de manera rápida y eficiente.

Entre las principales características de Laravel se encuentran la simplicidad y la elegancia del código, la documentación extensa y clara, la robustez de seguridad, la facilidad de integración con bibliotecas y paquetes de terceros y la capacidad de crear API y aplicaciones de tiempo real. Laravel es una excelente opción para desarrollar proyectos web empresariales y de cualquier otro tipo.

¿Qué necesitamos para instalar Laravel?

Para empezar:

- XAMPP
- COMPOSER
- CMD ó GIT BASH
- COMANDOS (Instalación y comprobación).
- VSC

https://www.youtube.com/watch?v=bNNAexB7sCw

1. XAMPP

   No vamos a profundizar en este apartado, porque ya lo hemos realizado en trabajos anteriores, pero sí podemos indicar que la ubicación desde dónde podemos descargarlo, es la dirección que indicamos a continuación.

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

2. COMPOSER:

   a. ¿qué es Composer?

Composer es una herramienta de gestión de dependencias para PHP que permite instalar, actualizar y administrar las bibliotecas y paquetes de terceros que una aplicación PHP necesita para funcionar.

Con Composer, los desarrolladores pueden especificar las dependencias de sus proyectos PHP en un archivo llamado "composer.json" y Composer se encarga de descargar e instalar las bibliotecas y paquetes necesarios en la estructura de directorios del proyecto. Además, Composer garantiza que todas las dependencias estén en las versiones correctas y maneja los conflictos entre dependencias.

De esta manera, Composer facilita la gestión y el mantenimiento de los proyectos PHP, ya que los desarrolladores no tienen que preocuparse por descargar y actualizar manualmente las bibliotecas y paquetes de terceros que sus proyectos requieren. En lugar de eso, pueden centrarse en escribir el código de su aplicación y dejar que Composer maneje las dependencias.

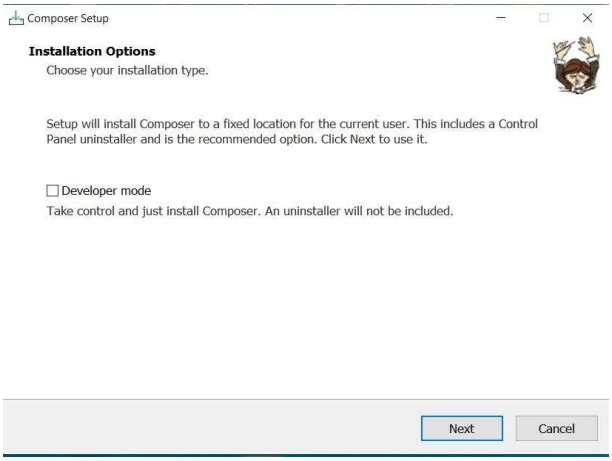
En resumen, Composer es una herramienta esencial para cualquier proyecto PHP moderno, ya que permite una gestión de dependencias fácil y automatizada, lo que ahorra tiempo y reduce la posibilidad de errores.

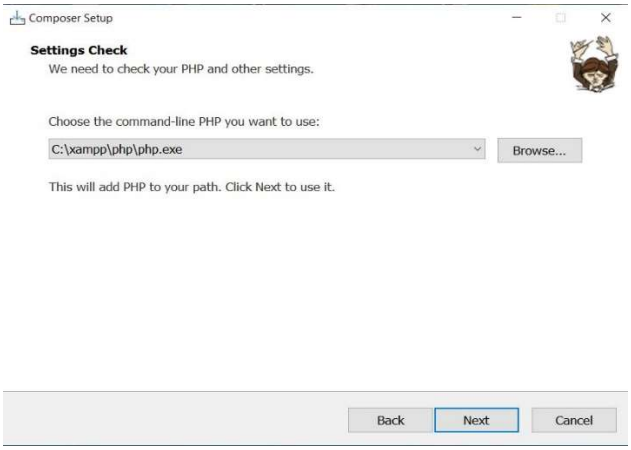
b. Instalación de Composer de forma Manual a través de su interfaz.

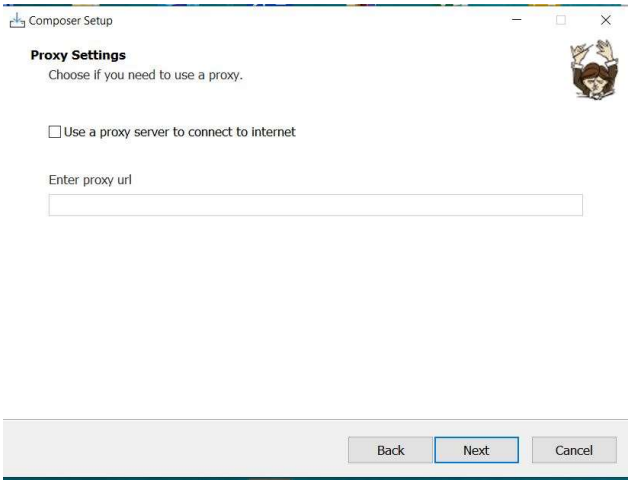
Descargamos el fichero desde la ruta:

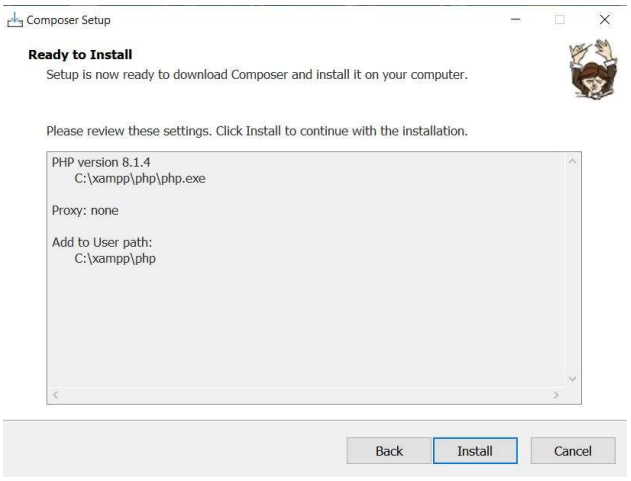
https://getcomposer.org/download/

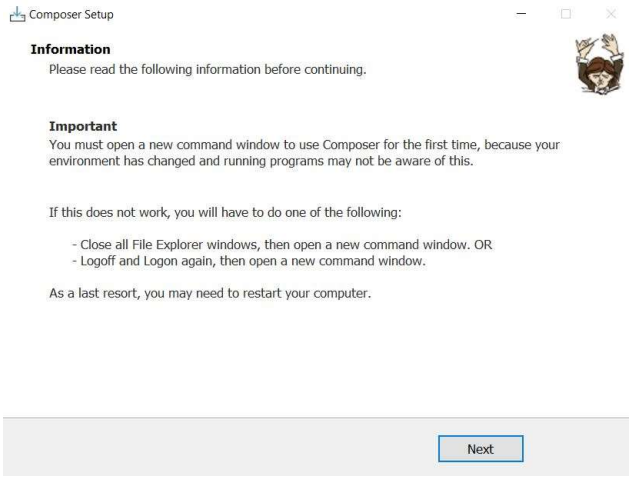
Una vez descargado seguimos las siguientes pautas para su instalación:



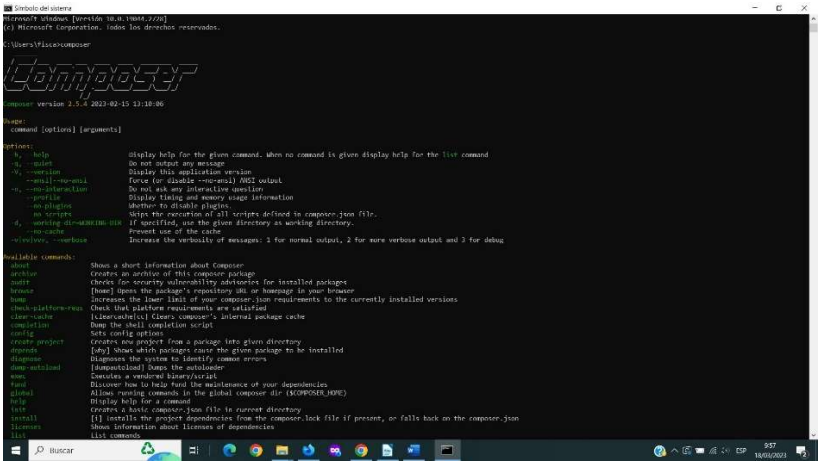

Composer Setup
Settings Check
We need to check your PHP and other settings.
Choose the command-line PHP you want to use:
C:\xampp\php\php.exe
Browse...
This will add PHP to your path. Click Next to use it.
Back
Next
Cancel

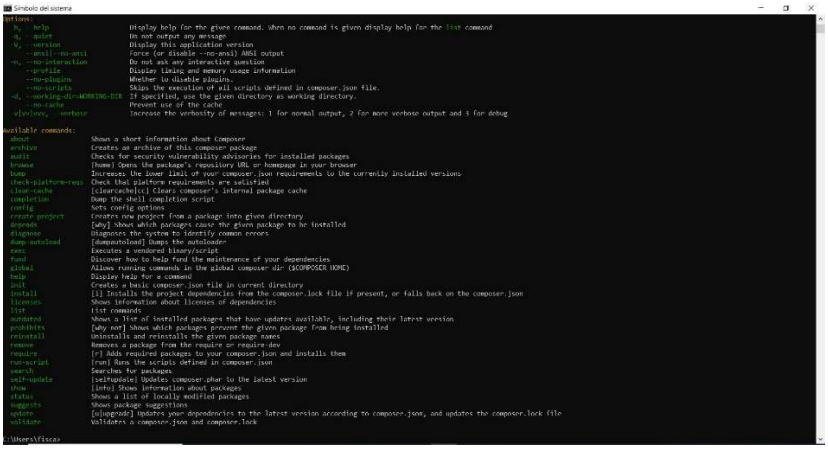

Composer Setup
Proxy Settings
Choose if you need to use a proxy.
Use a proxy server to connect to internet
Enter proxy url
Back
Next
Cancel


Composer Setup
Ready to Install
Setup is now ready to download Composer and install it on your computer.
Please review these settings. Click Install to continue with the installation.
PHP version 8.1.4
C:\xampp\php\php.exe
Proxy: none
Add to User path:
C:\xampp\php
Back
Install
Cancel



Comprobamos que la instalación se ha realizado de forma correcta:
¿Cómo lo hacemos? Abrimos el terminal de windos con ‘cmd’ y una vez en él escribimos ‘Composer’ ejecutamos y este es el resultado que obtenemos para comprobar que se ha instalado de forma correcta.





c. Comandos para instalar Composer a través de comandos.
Tutorial explicativo para hacerlo en windows
https://www.youtube.com/watch?v=yp04wvbAJPs

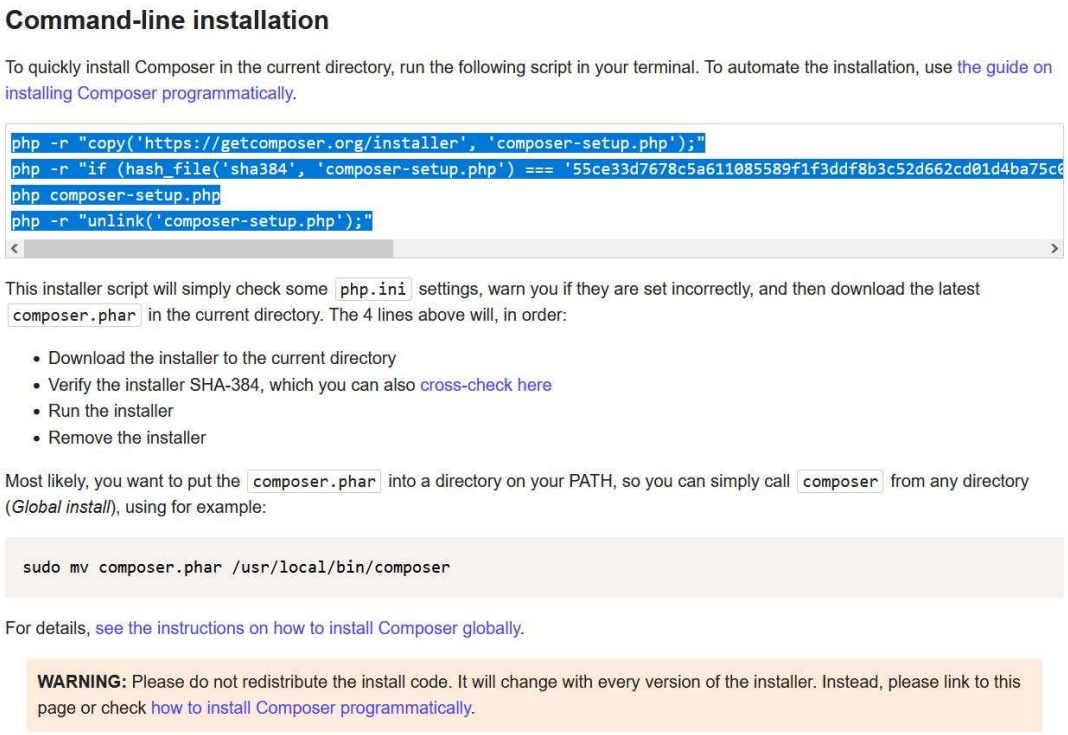
Comandos a utilizar para la instalación de Comproser:

```
php -r "copy('https://getcomposer.org/installer',
'composer-setup.php');"

php -r "if (hash_file('sha384', 'composer-setup.php') ===
   '55ce33d7678c5a611085589f1f3ddf8b3c52d662cd01d4ba75c0ee
   0459970c2200a51f492d557530c71c15d8dba01eae') { echo
   'Installer verified'; } else { echo 'Installer
   corrupt'; unlink('composer-setup.php'); } echo
   PHP_EOL;"

php composer-setup.php

php -r "unlink('composer-setup.php');"
```



Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use the guide on installing Composer programmatically.

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === '55ce33d7678c5a611085589f1f3ddf8b3c52d662cd01d4ba75c6
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

This installer script will simply check some `php.ini` settings, warn you if they are set incorrectly, and then download the latest `composer.phar` in the current directory. The 4 lines above will, in order:

- Download the installer to the current directory
- Verify the installer SHA-384, which you can also cross-check here
- Run the installer
- Remove the installer

Most likely, you want to put the `composer.phar` into a directory on your PATH, so you can simply call `composer` from any directory (*Global install*), using for example:

```
sudo mv composer.phar /usr/local/bin/composer
```

For details, see the instructions on how to install Composer globally.

WARNING: Please do not redistribute the install code. It will change with every version of the installer. Instead, please link to this page or check how to install Composer programmatically.

3. Documentación Laravel: comandos instalación Laravel.

   Los comandos tanto para la instalación de Laravel, como para consultar su documentación lo podemos encontrar en el siguiente enlace:
   http://laravel.com/docs/10.x

   Comandos de instalación:

   Composer create-project laravel/laravel *ProjecNam*

   'OR'

   Composer global require laravel/installer

4. Nombre del proyecto: comandos para crear proyecto
   a. Comandos para crear proyecto
      laravel new *ProjecName*

   b. Comprobación del proyecto creado.
      http://localhost/s1project/public/

5. Visual Studio Code.

   Hemos realizado la instalación de paquetes necesarios para Laravel, a través de la terminal de VSC.

Son beneficiosas determinadas extensiones para instalar en VSC, como las siguientes:

- Laravel snippets
- Laravel Blade format
- Laravel Blade Snippets
- Laravel go to view
- Php intelliphense
- Spanish Language Pack for VSC
- Tailwind CSS intellisense
- Alpine.js intellisense
- Github Copilot

B. ¿Cuáles son las características principales?

Laravel es un framework de PHP moderno y robusto que tiene varias características principales que lo hacen popular y ampliamente utilizado en la comunidad de desarrollo web. A continuación, se presentan algunas de las características principales de Laravel:

Enrutamiento elegante y fácil de usar: Laravel proporciona una sintaxis de enrutamiento fácil de usar que permite definir rutas con parámetros, prefijos y restricciones.

Capacidad de autenticación integrada: Laravel viene con características integradas de autenticación, lo que facilita la implementación de la autenticación y autorización en aplicaciones web.

ORM (Mapeo objeto-relacional) integrado: Laravel incluye Eloquent, un ORM integrado que permite la interacción con bases de datos de forma sencilla y natural, utilizando una sintaxis elegante y expresiva.

Sistema de plantillas Blade: Laravel proporciona un sistema de plantillas llamado Blade que permite crear vistas de forma sencilla y expresiva. Blade incluye características como herencia de plantillas, control de flujo y directivas personalizadas.

Consola de Artisan: Laravel viene con una herramienta de línea de comandos llamada Artisan que permite la creación de migraciones, modelos, controladores, pruebas y más. Artisan también permite la personalización de comandos para automatizar tareas repetitivas.

Facilita la creación de API: Laravel facilita la creación de API mediante el uso de controladores de recursos y el soporte para autenticación API mediante tokens.

Soporte para caché: Laravel proporciona una API de caché simple y expresiva que permite almacenar en caché datos de forma sencilla.

Arquitectura MVC: Laravel sigue una arquitectura MVC (Modelo-Vista-Controlador) que permite separar la lógica de la aplicación en diferentes capas.

Estas son solo algunas de las características principales de Laravel que lo hacen popular y ampliamente utilizado en la comunidad de desarrollo web. Laravel también cuenta con una amplia documentación y una comunidad activa de desarrolladores, lo que lo hace aún más atractivo para el desarrollo de aplicaciones web.

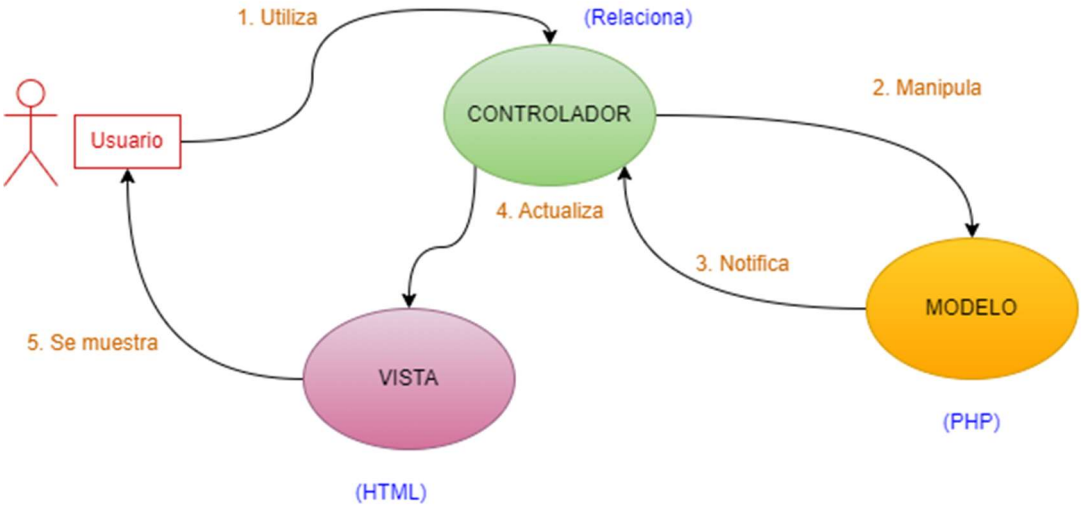
C. Se basa en la estructura MVC

MVC significa "Modelo-Vista-Controlador" y es un patrón de diseño de software comúnmente utilizado en el desarrollo de aplicaciones web. Es utilizado por Laravel para organizar la lógica de la aplicación en diferentes capas, lo que permite separar las responsabilidades y facilita la gestión del código.

En Laravel, el modelo representa los datos y la lógica de la aplicación relacionados con esos datos. El controlador se encarga de procesar las solicitudes del usuario y controlar la lógica de negocio de la aplicación. La vista es responsable de mostrar los datos al usuario y proporcionar una interfaz de usuario interactiva.

El modelo, la vista y el controlador se comunican entre sí mediante una serie de reglas y convenciones que Laravel establece. Por ejemplo, el controlador puede llamar al modelo para recuperar los datos de la base de datos y luego pasar esos datos a la vista para que se muestren al usuario.

La arquitectura MVC de Laravel permite una mayor modularidad y flexibilidad en el desarrollo de aplicaciones web. También facilita la colaboración en equipo, ya que cada desarrollador puede trabajar en una capa diferente de la aplicación sin afectar el trabajo de los demás.



D. ¿Qué es un ORM? ¿Cuál es el ORM que utiliza Laravel por defecto?

Un ORM (Object-Relational Mapping) es una técnica de programación que permite a los desarrolladores interactuar con una base de datos relacional utilizando objetos en lugar de sentencias SQL. En lugar de tener que escribir código SQL para cada operación de base de datos, los desarrolladores pueden trabajar con objetos en su lenguaje de programación, lo que simplifica el proceso de desarrollo y hace que el código sea más legible y mantenible.

En Laravel, el ORM por defecto es Eloquent. Eloquent es un ORM de alto rendimiento y fácil de usar que utiliza convenciones de nomenclatura para mapear modelos de objetos a tablas de base de datos y relaciones entre ellos. Con Eloquent, los desarrolladores pueden interactuar con sus bases de datos utilizando una sintaxis limpia y concisa en lugar de tener que escribir consultas SQL directamente. Eloquent admite una amplia gama de relaciones, como uno a uno, uno a muchos, muchos a muchos y polimórficas, y también incluye características avanzadas como la carga impaciente, el acceso a atributos, el control de versiones y la migración de esquemas de base de datos.

Eloquent es un ORM potente y fácil de usar que simplifica el acceso a la base de datos en Laravel, esto ayuda a que los desarrolladores puedan centrarse en el desarrollo de la lógica de la aplicación en lugar de preocuparse por la gestión de la base de datos.

E. Realizar una aplicación sencilla y analizar las partes que consideréis más importantes del desarrollo.

Hemos realizado una aplicación para que un usuario pueda registrarse y logarse a través de una página web, en dicha aplicación hemos utilizado lenguaje php, html, css y javascript en cuanto a las secciones de Servidor y Cliente.

La aplicación la hemos realizado con VSC, a través de su terminal hemos insertado los comandos más importantes como son:

Crear proyecto laravel: laravel new *ProjectName*
Levantar servidor: php artisan serve
Migrar tablas a nuestra base de datos de PhpmyAdmin: php artisan migrate
Crear controladores: php artisan make:controller *NameController*
Crear petición: php artisan make:request *NameRequest*

Las carpetas y ficheros del proyecto realizado con Laravel sobre las que hemos trabajado son:

- App/Http/Controllers:
  - ✓ HomeController.php
  - ✓ LoginController.php
  - ✓ LogoutController.php
  - ✓ RegisterController.php

- App/Http/Requests:
  - ✓ LoginRequest.php
  - ✓ RegisterRequest.php

- App/Models:
  - ✓ User.php

- Database/migrations
-
- Database/migrations/css:
  - ✓ Bootstrap.min.css
  - ✓ Bootstrap.min.css.map

- Public/assets/js:
  - ✓ Bootstrap.bundle.min.js
  - ✓ Bootstrap.bundle.min.js.map

- Resources/views/auth:
  - ✓ Login.blade.php
  - ✓ Register.blade.php

- Resources/views/home:
  - ✓ Index.blade.php

- Resources/views/layouts/partials:
  - ✓ Messages.blade.php
  - ✓ Navbar.blade.php

- Resources/views:
  - ✓ App-master.blade.php
  - ✓ Auth-master.blade.php

- Routes:

- ✓ Web.php

Hemos empezado levantando el servidor con su comando correspondientes antes mencionado.

En primer lugar accedemos a la carpeta .env , para verificar que tanto el user, el password, el host y el nombre de la base de datos coincidan con la base de datos que hemos creado en phpmyadmin.

Modificamos el fichero User.php, incluimos en este ficher el username porque lo vamos a necesitar. Además tenemos que guardar el password, como se guarda en texto plano, tenemos que crear un método para que se encripte el password, por motivos evidentes de seguridad.

El ORM Enloquet, nos permite trabajar con los getter (acedemos las propiedades) and setter (modificamos valores, o por ejemplo a través de un método set, encriptamos un valor y lo guardamos).

Es muy importante crear los controladores, ya hemos indicado su correspondiente comando con anterioridad, los controladores nos permiten devolver vistas (a través de los métodos view()).

También son importantes las creaciones de peticiones (Request) ya que nos va a permitir establecer una serie de reglas para que nuestro programa opte por un camino y otro en su recorrido.

Volvemos a modificar el fichero User.php, donde vamos a incluir nuevas rutas, es la ubicación donde determinamos las rutas haciendo llamadas a clases que ya tenemos declaradas.

Creamos en la carpeta view el fichero auth.blade.php, para crear la vista e incluir en ella la dirección y otras etiquetas.

Volvemos a modificar Web.php para meter la ruta con los nuevos métodos que hemos implementado.

Hemos creado una función register() en la que insertamos el método redirect() para que una vez el usuario sea registrado pueda acceder a una nueva pagina.

Si encontramos el error 419 es que nos faltaba incluir @csrf .

Seguimos creando las vistas, las rutas y todos los Controller y Request que necesitamos para nuestra aplicación.

Al realizar la parte del método para que nos muestre el nombre de la persona que se ha registrado, a través del uso ternario, lo gestionamos de tal forma que si no tenemos el name, nos muestre le username o viceversa.

Además de crear el Register y el Login en la web le añadimos también el Logout, para que podamos cerrar la sesión.

Es importante no olvidar importar las clases desde las que vamos a invocar métodos, atributos u objetos, a través de **use** …(ruta de la clase/Nombre de la clase).

Cuando ya tenemos la aplicación funcionando, pasamos a darle formato, a la parte de Cliente.

Nos ayudamos de la página web de Bootstrap, desde la que descargamos ficheros css y js, en la carpeta públic del proyecto creamos las carpetas css y js, dentro de cada una pegamos sus correspondientes códigos que hemos copiado de Bootstrap.

Una vez terminada la parte de estilos, nos vamos a la carpeta ya creada con nombre 'partials', para incluir los mensajes de avisos a través del fichero messages.blade.php. Por ejemplo para que aparezcan mensajes de aviso si los datos no son correctos, o si no existe el usuario.

Y para finalizar, es muy importante saber que en la carpeta config es el lugar del proyecto donde se determina cómo va a funcionar nuestra aplicación a nivel de apis o librerías de Laravel. Podemos acceder y ver los métodos o clases con los que hemos trabajado, porque están previamente implementados. En otros lenguajes como Python también disponen de muchas librerías con métodos o clases previamente implementados, y a los que se puede acceder de la misma forma, pulsando control y pinchando con el cursor.

No ha sido fácil realizar la aplicación, ya que el concepto de mvc en un primer momento no me ha parecido demasiado fácil, ya que tienes que tener muy claro qué es y para qué es lo que estás haciendo en cada momento, pues finalmente es como si todo estuviera encadenado de una forma o de otra. Aunque el esquema para entender a priori el mvc parece fácil, una vez estás trabajando con Laravel, tienes que tener muy claro todo para que no empiecen a surgir errores. A pesar de ello he terminado satisfecha.