# Introducción

El Proyecto de Fin de Curso del Ciclo Formativo de Grado Superior Desarrollo de Aplicaciones Web cuyo título es MovieExprert, es una interfaz web de cine donde podrás ver información sobre las películas de estreno y dispondrá de una comunidad donde se realizaran. Este proyecto será desarrollado por Alicia de la Cruz Peris y Sheila Bravo Sánchez alumnas del centro IES Augustóbriga en la promoción 2015/2017.

# Descripción del Proyecto

MovieExpert es una interfaz web de cine donde podrás ver información sobre las películas de estreno y dispondrá de una comunidad donde se realizarán concursos de  películas, cortometrajes,.. que suban los usuarios.

Es una aplicación donde se actualizarán las películas de estreno con su descripción resumida, director, guionistas, actores principales… y donde se podrá ampliar la información de cada película más detalladamente y con comentarios de usuarios acerca de la película. También habrá una sección de concurso de cortometrajes, donde los usuarios podrán inscribirse al concurso si entran dentro del plazo, subiendo algún proyecto de cortometraje, o también podrán valorar el resto de cortometrajes, y darles una puntuación y añadir comentarios. El cortometraje que mayor puntuación tenga será el ganador.

El perfil de usuario se introducirá la experiencia laboral y la categoría (sí es director, cámara, guionista, director de fotografía, etc.) del usuario así como enlaces a sus proyectos externos a la aplicación, si sus proyectos están dentro de la aplicación se le podrá vincular a dichos proyectos.

Habrá un chat/foro que podrá iniciar un usuario que inicie un proyecto de cortometraje para poder interactuar con otros usuarios que estén interesados en participar en el proyecto, así este podrá ponerse en contacto y participar juntos en la creación del cortometraje cada uno participando según su especialidad, si uno está interesado y es cámara, será el cámara, otro será el guionista, y así hasta formar el equipo completo y al terminar podrán inscribirse al concurso.

Sólo podrán inscribirse al concurso los usuarios que estén registrados y logueados en la aplicación, así como iniciar un chat.

# Análisis de Requisitos

# Objetivos del Proyecto

Crear una interfaz web (frontend y backend) relacionada con el cine. Donde se podrá buscar películas y ver su información, tráiler, ranking,… Esta página también dispondrá de una comunidad para usuarios registrados donde podrás conocer a personal técnico cinematográfico y poder desarrollar un proyecto cinematográfico juntos. Y se realizarán concursos de películas, cortometrajes,...

Hemos elegido este proyecto porque creemos necesario un apoyo en el sector cinematográfico para aquellos profesionales que necesiten desarrollar sus ideas, y no tienen los contactos suficientes para llevarlas a cabo. Además se incluyen concursos y críticas para impulsar y darse a conocer sus ideas de forma profesional con otras personas del sector. Creemos que esta aplicación puede servir para iniciarlos profesionalmente en el cine, televisión, web series.

# Tecnologías Utilizadas

Lenguaje:

* HTML
* CSS
* PHP
* JavaScript

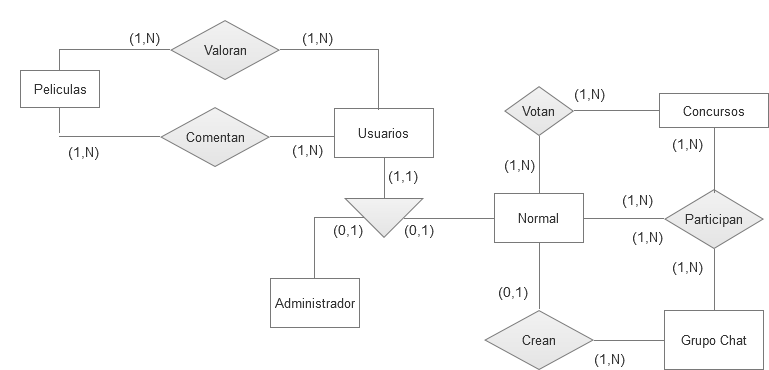
Herramientas:

* Bootstrap
* JQuery
* Framework: Laravel

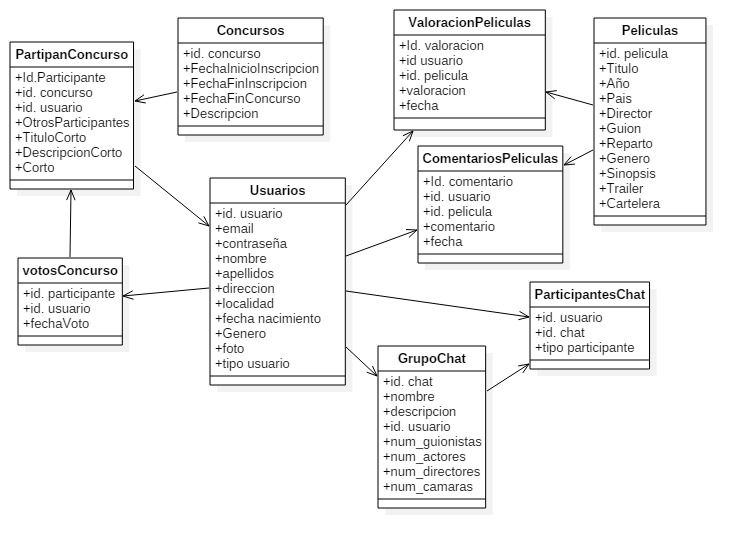
# Desarrollo del proyecto

# Base de datos

# Modelo Entidad Relación

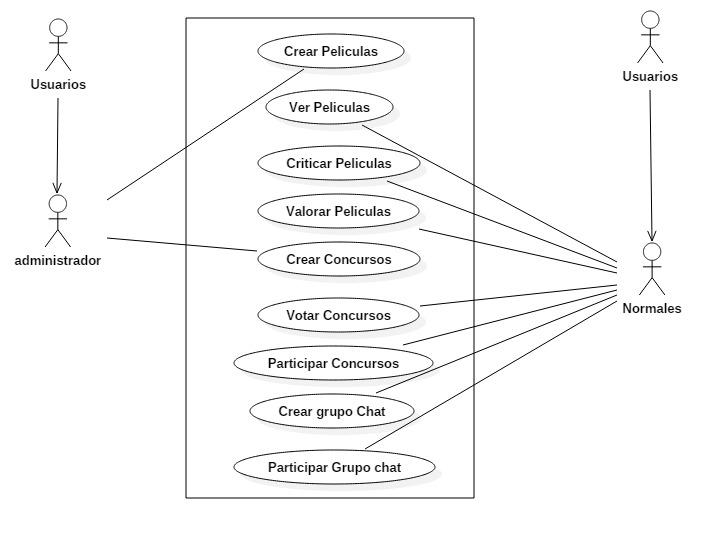


# Modelo Relacional



# Base de datos

# Diagrama Casos de Usos

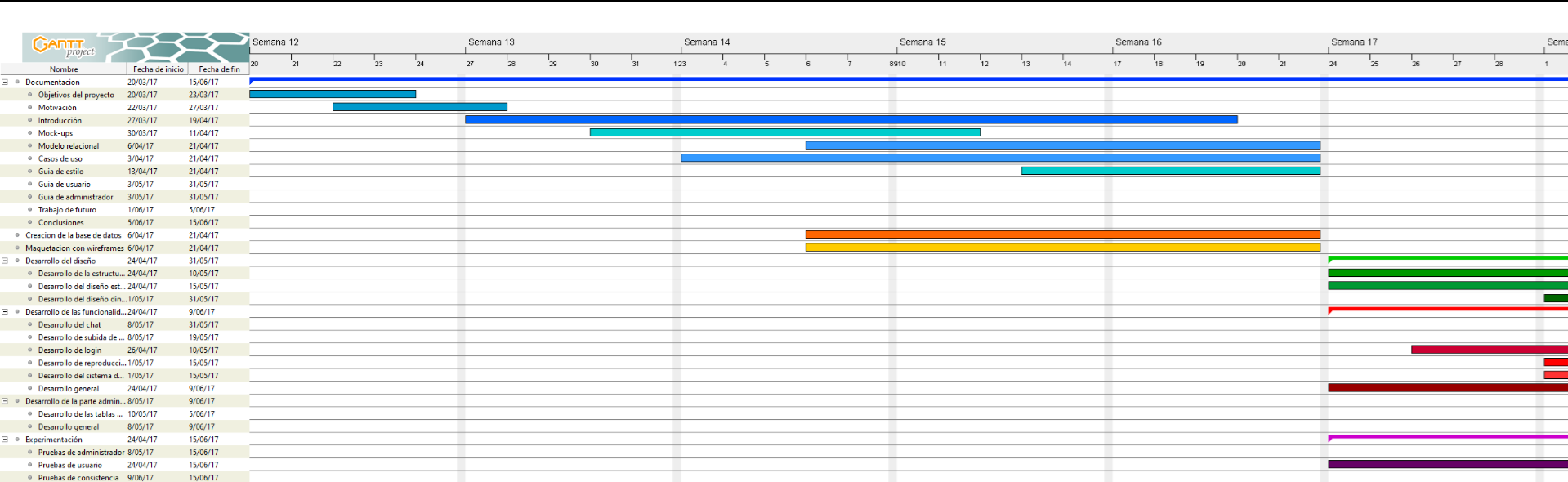


# DESCRIPCIÓN DE MODULOS

* **Frontend:** Sección de la aplicación web con funcionalidad limitada para el uso de los usuarios normales o usuarios anónimos que puedan visitar la página. Solo se ofrecerá información sobre las películas para los usuarios anónimos o no registrados, mientras que a los usuarios normales o registrados además de ver la información de las películas podrán criticarlas y valorarlas. También tendrán acceso a otras dos secciones dentro de la página, a la sección de concursos y a la sección del chat.
* **Backend:** Sección de la aplicación web con funcionalidad completa de administrador para manipular toda la información de la base de datos, gestionar usuarios, películas, concursos,…
* **Módulos MVC:** Estos módulos separan los datos y la lógica de una aplicación de la interfaz de usuario y el modulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes ara la representación de la información, y por otro lado para la interacción del usuario.
  + Modelos: Gestiona todos los accesos a la información con la que opera el sistema, tanto consultas como actualizaciones, implementando los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación. Envía a la vista aquella arte de la información en cada momento se le solicite para que sea mostrada. Las peticiones de acceso o manipulación de información llegan al ‘modelo’ a través del ‘controlador’.
    - ***Nuestros Modelos***
    - ***Nuestros Modelos***
  + Controlador: Responde a eventos e invoca peticiones al ‘modelo’ cuando se le hace alguna solicitud. Tambien tiene que enviar comandos a su ‘ vista’ asociada si se solicita un cambio en la forma en que se presenta el ‘modelo’. El controlador hace de intermediario entre la ‘vista’ y el ‘modelo’.
    - ***Nuestros Controladores***
    - ***Nuestros Controladores***
  + Vistas: Presenta el ‘modelo’ en un formato adecuado para interactuar, normalmente la interfaz de usuario.
    - ***Nuestras Vistas***
    - ***Nuestras Vistas***

# Diagrama de Gant

El diagrama de Gantt es una herramienta para **planificar y programar tareas** a lo largo de un período determinado. Gracias a una fácil y cómoda visualización de las acciones previstas, permite realizar el seguimiento y control del progreso de cada una de las etapas de un proyecto y, además, reproduce gráficamente las tareas, su duración y secuencia, además del calendario general del proyecto.



# Guía de estilos de MoviExpert

**Introducción**

Una guía de estilo es un documento que establece el diseño a nivel de formato de una

página web. En ella se establecen las directrices comunes para el estilo de textos e

imágenes, el uso de colores y fuentes, las variaciones de logotipo que se pueden

utilizar, etc.

Es un documento que sirve para que los desarrolladores implicados en diversas partes

de la página web, puedan seguir delante de forma que sus desarrollos sean

consistentes entre sí. Por lo tanto, es muy importante en proyectos en los que va a

participar más de una persona.

Los puntos que se tratarán en esta guía de estilo serán los siguientes:

**1.** Objetivos de este documento

En este punto se tratará cual es el objetivo del documento, se definirá el concepto

de MoviExpert para entender a fondo el contexto de la página web y el diseño.

**2.** Usabilidad

En este apartado se definirá a qué tipo de público está dirigida la página y que

facilidades debe dar al usuario.

**3.** Accesibilidad

La accesibilidad de página web se refiere al uso de ésta por parte de personas

discapacitadas principalmente.

**4.** Tipografía y texto

Se refiere al tipo y tamaño de letra, en toda la página, definiendo los títulos,

subtítulos, descripciones, formulario, cabecera, pie, etc. Además también incluye el

formato del texto, el espaciado etc.

**5.** Paleta de colores

Aquí se definirá la gama de colores presente en la página web, los colores que

deben tener los títulos, las imágenes, la letra, las secciones, etc.

**6.** Cabecera y pie

Se define el estilo y diseño de la cabecera y pie de página.

**7.** Logo y fotos

Se indica el logo a utilizar y el tipo de imágenes que se deberán usar, indicando

tamaños, dimensiones, etc.

**8.** Iconografía

Se mostrarán la gama de iconos que utiliza la web.

**9.** Elementos de interfaz

Se indican el tipo de botones a utilizar, el diseño de elementos gráficos, etc.

Una vez vista cual es la estructura del documento, se puede comenzar a desglosar

correspondientemente.

* **Objetivos de este documento**

Con este documento se pretende hacer una guía para los desarrolladores que deseen

realizar mejoras a la página web, puedan hacerlo sin crear inconsistencias.

La página web que se va a desarrollar tiene por nombre “MoviExpert”. Es una página

de estrens de cine. Por lo tanto, la página web debe tener un aspecto sencillo e informativo, donde lo importante es el contenido. Porque los usuarios quieren información acerca de las películas, las opiniones de otros usuarios para decidir si ver la película o no. Aparte es una página que sirve para que gente del sector audiovisual, que no se conocen con anterioridad, puedan conocerse y realizar proyectos juntos, participando de esta forma, en concursos internos de la página.

El estilo y el diseño juegan una parte muy importante a la hora de mostrar una interfaz

“user-friendly” o amigable.

*  **Usabilidad**

La utilización de esta página web está dedicada a personas de todas las edades que les guste el mundo del cine. Esto implica diversos estilos de diseño:

 Debe ser una página sencilla

 Debe ser confiable y seguro ya que los usuarios podrán acceder al perfil de otros usuarios para poder ver sus trabajos realizados con anterioridad.

Debe ser fácil de usar, de forma que el usuario sepa en cada momento dónde

está.

*  **Accesibilidad**

La accesibilidad de esta web, no dispone, actualmente, de facilidades para personas

ciegas. La página podrá verse a través de dispositivos móviles o tablets. Ya que es una página de información donde también podrás ver proyectos realizados de otros usuarios.

*  **Tipografía y texto**

La tipografía utilizada en la página web estará predominada por dos fuentes:

 La fuente “Arial”: esta fuente asegura la legibilidad en cualquier navegador.

 La fuente “Lucida Calligraphy”: esta fuente da un aspecto elegante y confiable a la

página web. Se utilizará principalmente en títulos y subtítulos.

Letra “Arial” : Arial,Arial,Arial.

Letra “Lucida Calligraphy” : Lucida,Lucida,Lucida.

El tamaño de la letra dependerá del contexto utilizado.

* Títulos: 26 píxeles.
* Subtítulos: 18 píxeles.
* General y otros: 12 píxeles.

El color del texto será predominantemente negro. Se incluirán en algunos detalles el

rojo.

La cabecera se tratará como títulos y subtítulos.

Los elementos del menú y el pie se tratarán como subtítulos.

Los formularios y la página en general utilizarán principalmente subtítulos y texto

general.

El formato de texto será principalmente normal, aunque se incluirá para remarcar temas importantes la negrita y para dar un aspecto más elegante la cursiva.

Se usará un espaciado interlineal de 1,15 píxeles principalmente. El espaciado entre

las letras será normal, salvo en algunos títulos y subtítulos que se podrá utilizar un

espaciado de hasta 5 píxeles.

* ** Paleta de colores**

Los colores principales de la web deben definir el sentido de la web, por lo que los

colores escogidos son:

#D69437 #000000 #CCCCCC

El negro será el color de la cabecera y de la letra del contenido de la página.

El rojo anaranjado será para el menú principal, y algunos títulos o subtitulos.

El gris para resaltar algo y hacerlo más llamativo.

El color blanco también es importante para los elementos del menú , el inicio de sesión y el registro.

* **Cabecera y pie**

La cabecera se situará el logo de la web a la izquierda, un buscador central, para hacer una búsqueda rápida de lo que se quiera encontrar, y el menú estará compuesto por

cuatro enlaces inclinados hacia la izquierda y simétricos para dar un aspecto limpio y sencillo.

En la esquina de la derecha se situará el login y el registro del usuario, como es habitual en la mayoría de las páginas web.



La fuente utilizada es “arial” normal, el tamaño del menú son subtítulos (14px) y el de identificación tiene un tamaño algo más pequeño de 12px.

Y en cuanto al footer o pie de página tendrá otro estilo, será de color gris, con letra negra de un tamaño de 14px al igual que el del menú.



* **Logo y fotos**

El logo se situará siempre en la cabecera. El color será los mismos que la página web , blanco, rojo anaranjado y gris, sobre fondo negro. El tamaño de este logo será siempre fijo y estará en la misma posición, es decir en la esquina superior izquierda, lo que permitirá seguir la lectura natural.



Las dimensiones del logo son : 387 x 67 píxeles.

En cuanto a las imágenes, se tienen 4 tipos de imágenes:

*  Imágenes de Inicio:



Dimensiones : 145 x 170 píxeles.

Fondo : blanco.

Es la fotografía que sirve de enlace para conocer la información técnica de la película.

*  Imágenes de ficha técnica de la película:

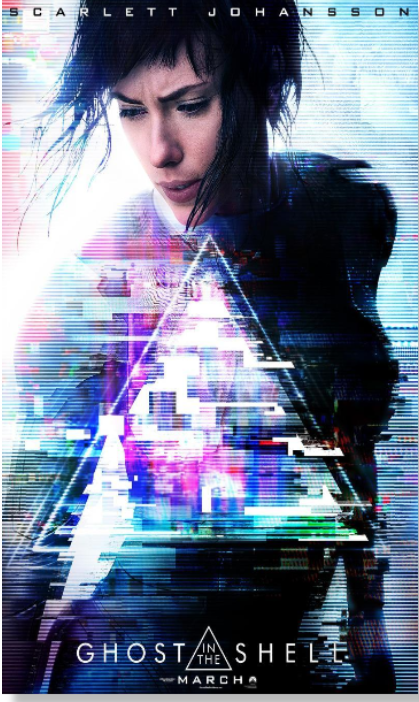


Dimensiones : 325 x 371 píxeles.

Fondo: blanco.

Es la imagen de la información detallada de la película.

* **Foto de tráiler**

****

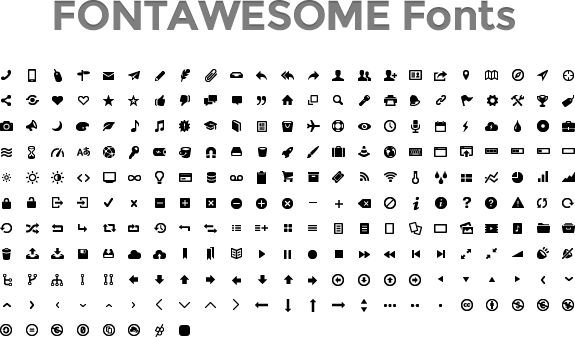
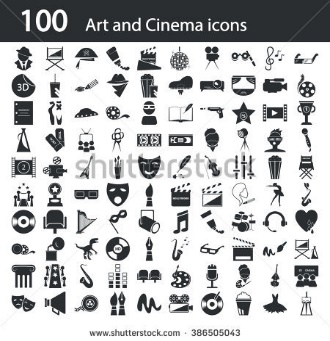
Dimensiones : 416 x 699 píxeles.

Fondo: blanco

Es donde se muestra el tráiler de la película.

* **Iconografía**

Los iconos a utilizar son los siguientes:



Estos iconos se usarán para los distintos motivos.

Los colores serán siempre de la paleta de colores.

*  **Elementos de interfaz**

Se entienden elementos de la interfaz como aquellos que permiten la interacción con el usuario, es decir, botones, aspectos gráficos y aquellas funciones que dan “movimiento” a la web.

Los botones deberán ser cuadrados con el fondo negro y las letras blancas. El

texto será como los subtítulos. Cuando se pase el ratón por encima de una imagen de coche. Esto dará un aspecto moderno a la página.



Algunos tendrán el borde redondeado.

* **Estructura de la web**

La estructura de la página web se estructura verticalmente, exponiendo la información de forma jerarquizada y explícita.

De esta manera, se consigue que el cliente pueda leer bien el contenido. La jerarquización de la información se basa en los contenidos más actualizados. La página estará continuamente actualizándose para mostrar los últimos estrenos cinematográficos. Esto implica que la visualización de la página debe ser limpia. En cuanto a las páginas de configuración y administración o formularios, deberán ser sencillos, de fácil usabilidad y que concuerden con la idea de la página web. La zona de trabajo será la central, dejando la parte superior y la inferior para una cabecera y un pie, respectivamente, estáticos.

* **Manual de administrador**

El administrador, cuando se loguee tendrá el menú igual que el resto de usuarios pudiendo ver lo que ven los usuarios, pero también podrá acceder a su propio menú pudiendo gestionar el contenido de la página web.



Este será su propio menú. En usuarios podrá ver todos los usuarios que están en la base de datos, y podrá eliminarles si lo necesita. En estrenos, están las listas de las películas, podrá añadir nuevas películas, modificarlas y eliminarlas.

En concurso, el administrador puede añadir un nuevo concurso para que los usuarios puedan inscribirse, y por último el correo, donde podrá interactuar directamente con los usuarios, y resolverles las dudas que tengan.

# Desarrollo de la Página Web.

Para desarrollar el proyecto vamos a utilizar un framework MVC (Modelo Vista Controlador) denominado Laravel, este framework permite el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP. Gran parte de Laravel está formado por dependencias, especialmente de Symfony.

El MVC es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador.

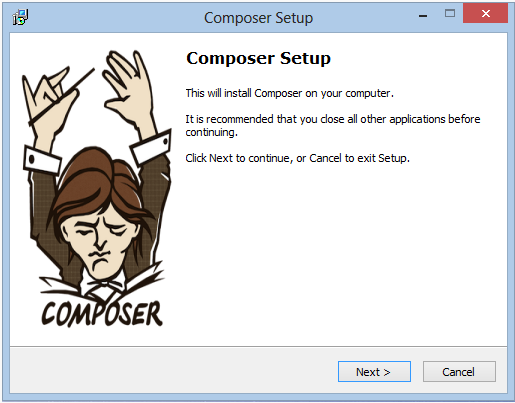
Previamente antes de instalar Laravel y empezar a programar nuestra página web necesitamos un servidor web y un servidor de base de datos, para el desarrollo del mismo vamos a trabajar con un servidor local llamado xampp.

Como el desarrollo del proyecto lo vamos a desarrollar entre dos personas para sincronizar el proyecto y poder programar a la vez vamos a utilizar una plataforma denominada github. Para poder utilizar esta herramienta en local y sincronizarlo a la plataforma en la red debemos tener instalado y configurado git en nuestro directorio del proyecto. Así podemos guardar los cambios realizados en el proyecto y poder sincronizarlo con nuestra compañera.

# Instalación Laravel

Para instalar Laravel necesitamos una herramienta conocida como Composer que se encarga de gestionar las dependencias en PHP. Composer te permite declarar las librerías de las cuales tu proyecto depende o necesita y las instala por ti.

Para descargar Composer vamos a la página oficial y descargamos el paquete de instalación. Ejecutamos el instalador descargado y seguimos la guía para configurar la instalación del paquete.



Una vez tenemos instalado Composer abrimos la consola de comando (CMD) nos situamos en la ruta donde queremos crear el proyecto, en nuestro caso dentro del servidor web en la ruta c:/xampp/htdocs y ejecutamos el siguiente comando para crear el proyecto:

**Composer create-proct Laravel/Laravel –prefer-dist**



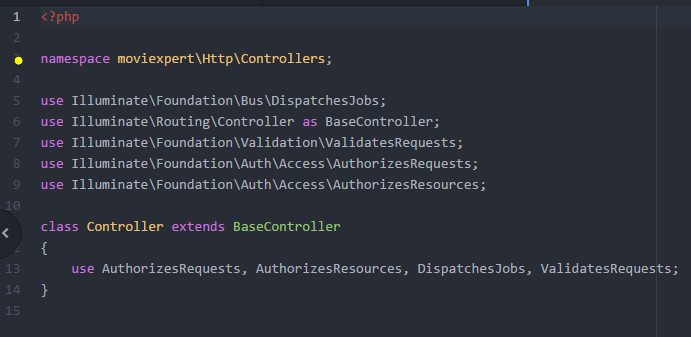
El siguiente paso antes de empezar a programar será configurar correctamente nuestro proyecto:

* Nombre del proyecto: vamos a darle a nuestro proyecto el nombre de moviexpert para ello desde el CMD ejecutamos el comando :

Php artisan app:name moviexpert



Comprobamos que se ha cambiado el nombre del proyecto abriendo cualquier fichero del proyecto y fijándonos en la primera línea donde hace referencia al namespace al que pertenece.



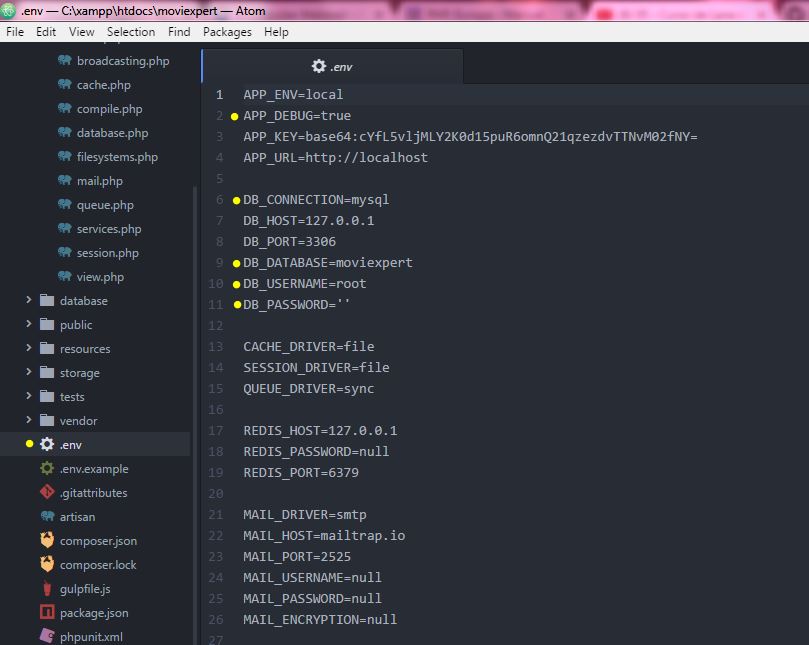
* Idioma del proyecto: por defecto viene configurado en ingles pero se puede cambiar para ello debemos editar el fichero /moviexpert/config/app.php, modificamos las siguientes líneas:
  + Línea 55: ‘timezon’ => ‘Europe/Madrid’
  + Linea 67: ‘locale’ => ‘es’

\*\*Previamente antes de modificar este fichero debemos hacer dos pasos:

1.- Tener la carpeta del idioma que queremos configurar en la ruta /moviexpert/resources/lang, sino es asi debemos descargar el paquete con ese idioma y colocar el directorio del idioma en la ruta mencionada anteriormente.

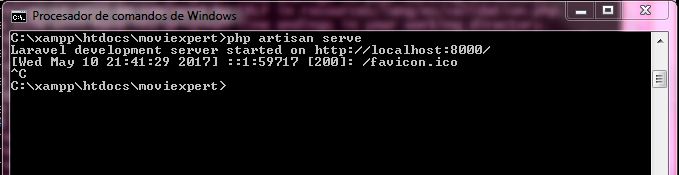
2.- Buscar en google el phptimezone para ver los datos de tu región.

* Gestor de la base de datos: podemos ver la configuración de la base de datos en el fichero /moviexpert/config/databases.php, pero el fichero que debemos modificar para especificar los valores a las variables de conexión de la base de datos, para ello debemos modificar el fichero /moviexpert/.env y pasarle los datos de conexión de nuestra base de datos.

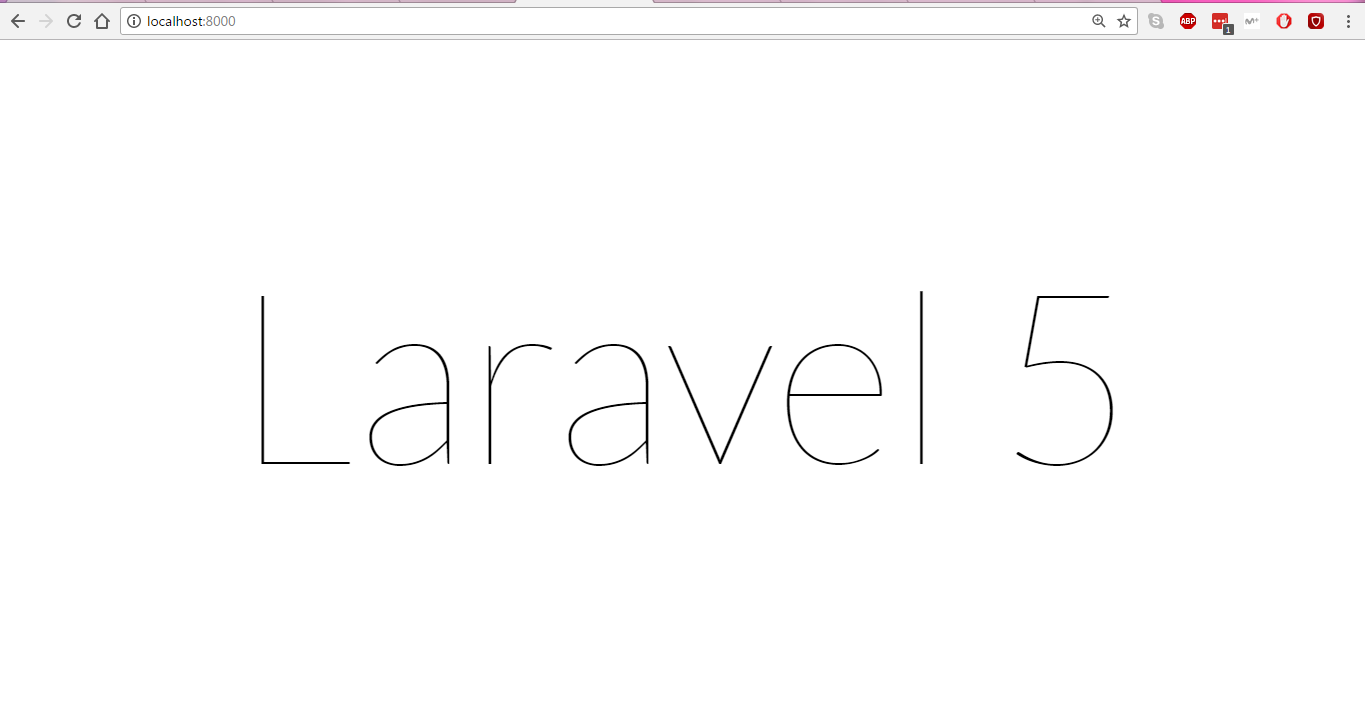


Comprobamos que todo está bien instalado y configuramos, abrimos el navegador y escribimos en la barra de navegación localhost:8000/ y cómo podemos apreciar se carga una página por defecto que instala Laravel en la instalación. Para poder verlo tenemos que tener arrancado nuestro servidor para ello en el CMD ejecuamos el siguiente comando:

Php artisan serve



Abrimos el navegador y vamos a la ruta localhost:8000 y cómo podemos apreciar se ha instalado Laravel correctamente

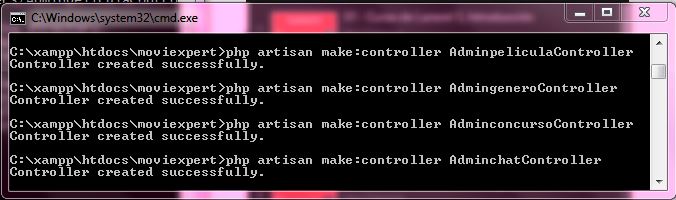


Empezamos el desarrollo de nuestra página web por el backend que es la parte de administración de la página, donde los usuarios con permiso de administrador pueden gestionar los usuarios, películas, géneros de películas, concursos y chat. Para ello creamos los controladores, modelos y vistas necesarios.

* Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información. Se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.
* Modelo:  Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación ([lógica de negocio](https://es.wikipedia.org/wiki/L%C3%B3gica_de_negocio)). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
* Vista: Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la [interfaz de usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario)) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

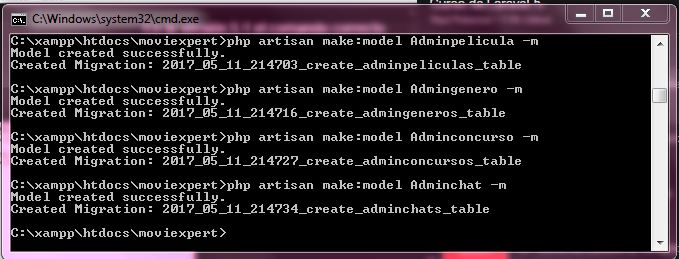
Para crear los controladores abrimos la consola de comandos y ejecutamos el siguiente comando:

Php artisan make:controller Nombre\_Controlador

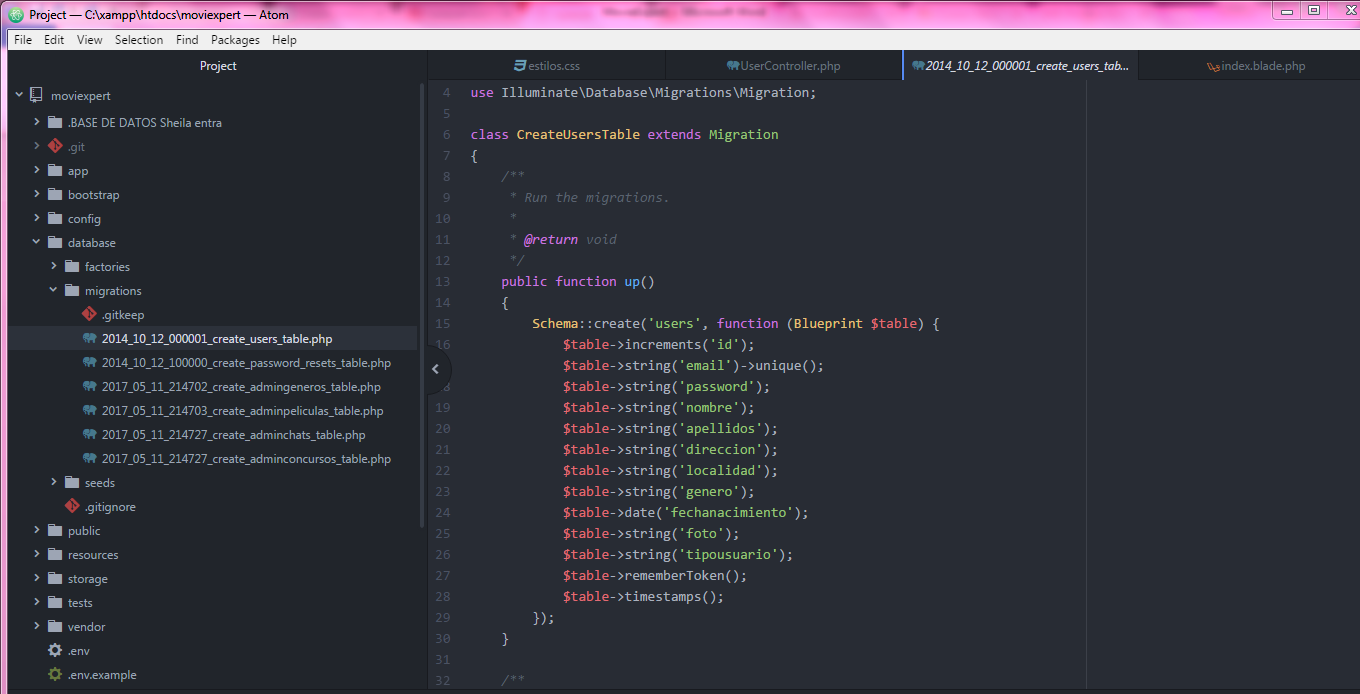


Para crea los modelos seguimos los mismos pasos para crear los controladores, pero son el siguiente comando:

Php artisan make:models Nombre\_Modelo –m

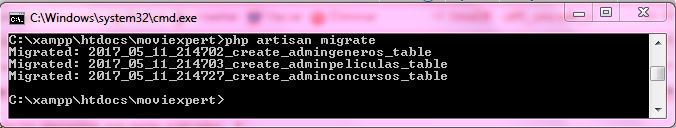


El parámetro –m se utiliza para que se nos creen los ficheros de migraciones a la base de datos. Estos ficheros tenemos que editarlos para indicar todos los campos que tendrá la tabla en la base de datos. Estos archivos se encuentran en la ruta /moviexpert/database/migrations



Una vez tenemos estos ficheros correctamente configurado pasamos a crear a las tablas en la base de datos. Abrimos el CMD y ejecutamos el siguiente comando:

Php artisan migrate



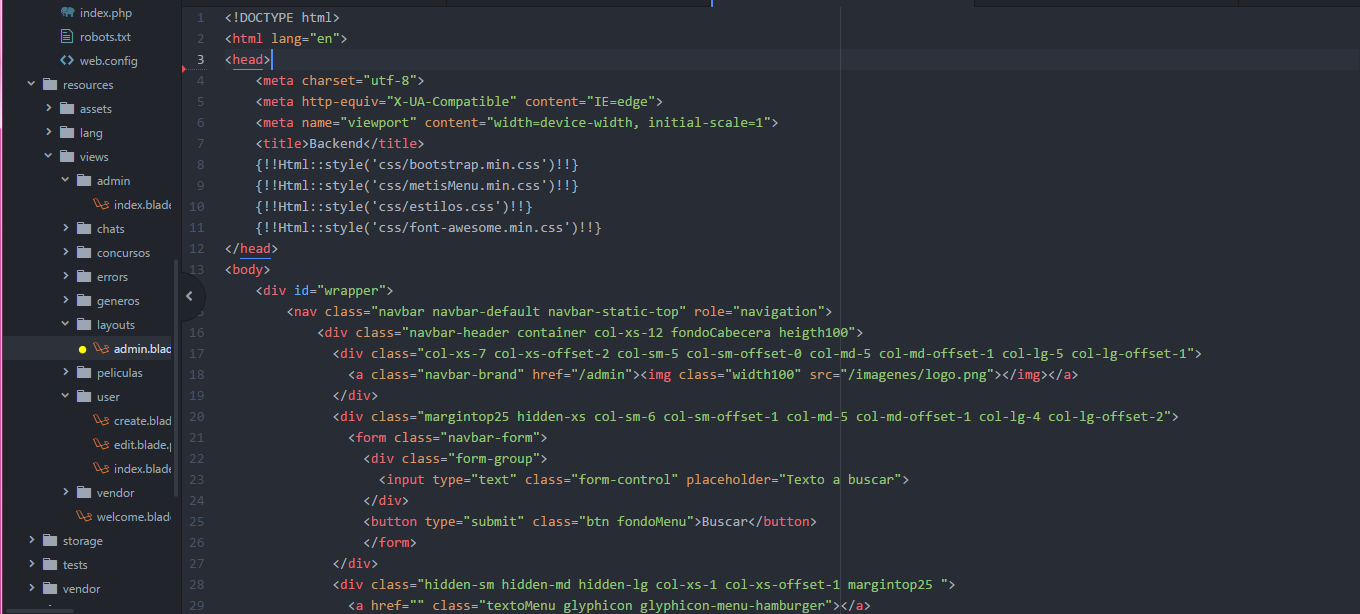
Para comprobar que todo se ha creado correctamente vamos a nuestra base de datos.



El siguiente paso será crear las vistas para ello en la ruta /moviexpert/resources/view creamos un directorio para cada controlador puesto que un mismo controlador va a estar asociado a varias vistas, este paso no es necesario es simplemente para tener más organizado el proyecto. Dentro de estos directorios empezamos a crear archivos para las vistas. Las vistas son ficheros que tiene el código de presentación. Por tanto, una vista será un archivo PHP que contendrá mayoritariamente código HTML, que se enviara al navegador para que este renderice la salida para el usuario.

Como todas las paginas tienen código en común como es la cabecera y el pie de página vamos a utilizar el motor de plantillas oficial de Laravel Blade. Para utilizar este gestor basta con añadir la extensión .blade.php al nombre de nuestro archivo, con esto le indicamos que utiliza el motor de plantillas blade y que el fichero es php.

Creamos una plantilla para el backend la cual tendrá la cabecera y el pie. Esta plantilla va a ser invocada desde todas las vistas que queramos que herede esta cabecera y pie.

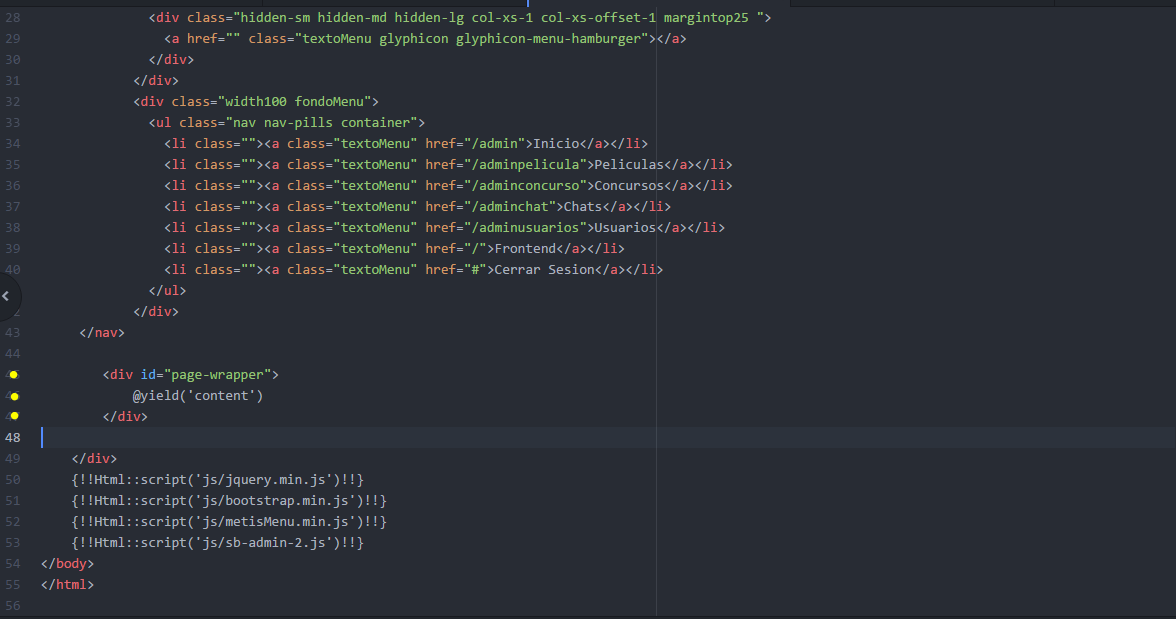


En la plantilla hay que indicarle en que parte de la estructura vamos a añadir el contenido específico de cada vista, para ello tenemos que definir una sección, para definirla usamos el siguiente código:

<div id="page-wrapper">

@yield('content') En esta línea definimos la sección.

</div>



Después de crear las plantillas creamos las vistas que vamos a utilizar. Para que la vista herede la plantilla que contiene la cabecera y el pie utilizamos la siguiente sintaxis:

@extends('layouts.admin') --> La plantilla de la que va a heredar la estructura

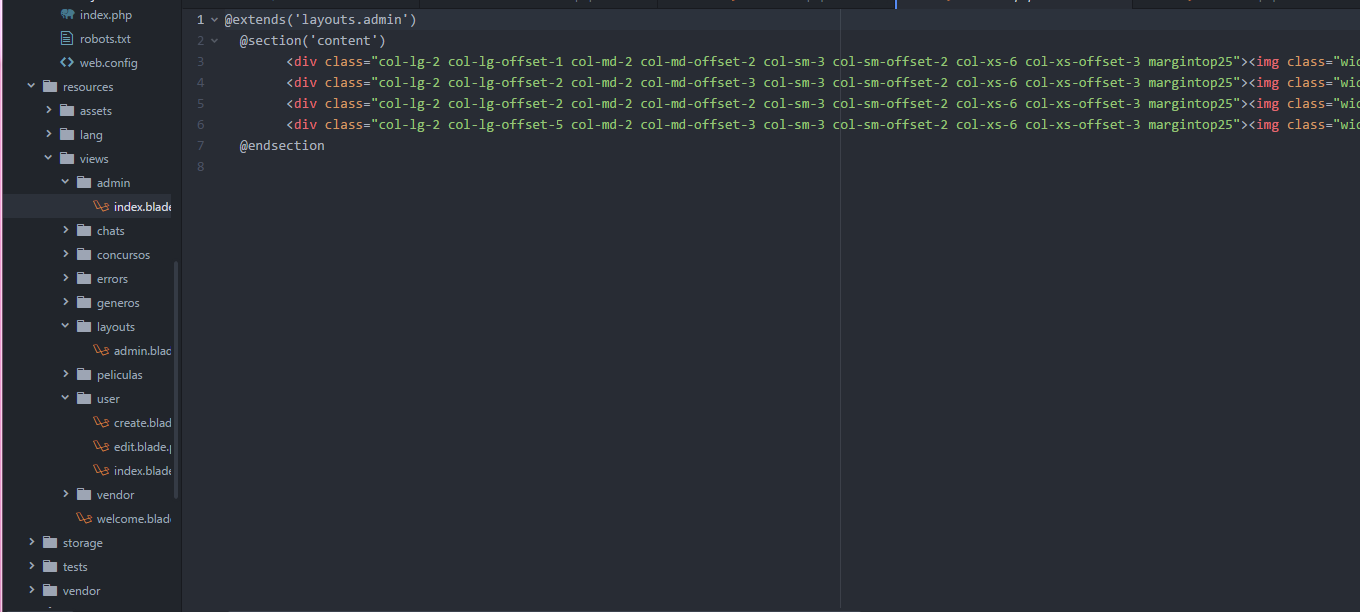
@section('content')--> La sección de la plantilla donde incluimos el código de la vista

<div>

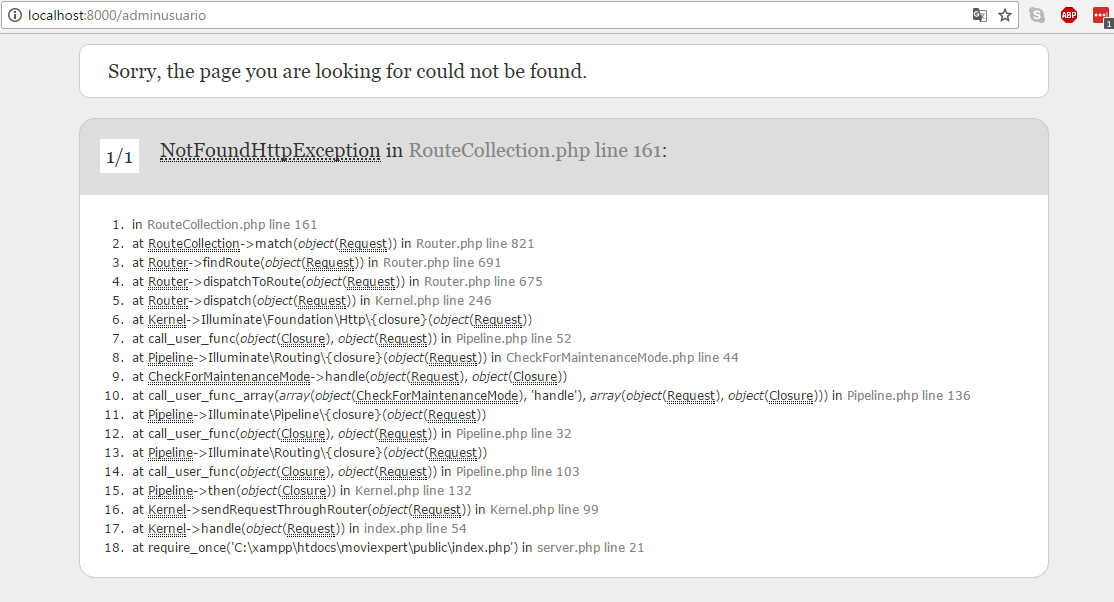
<p>Contenido en HTML específico para la vista</p>

</div>

@endsection



Una vez tenemos los tres componentes del MVC solo nos faltaría enlazarlos entre ellos para poder visualizarlo en el navegador. También hay que definir las rutas de navegación puesto que si no tenemos definidas las rutas aunque tengamos los tres componentes enlazados nos va a mostrar un error que nos indica que no está definida la ruta a la que queremos acceder.



Las rutas se definen en el fichero /moviexpert/app/http/routes.php

Existen dos formas de crear las rutas:

1. Route::get(‘ruta’,’nombreControlador@funcion o función’); -> Lo usamos para una ruta concreta que va a mostrar una vista determinada. Podemos ir directamente a una vista estática que no vaya estar enlazada con ningún controlador ni modelo o podemos utilizar una función concreta del controlador definido y este retorne la vista.

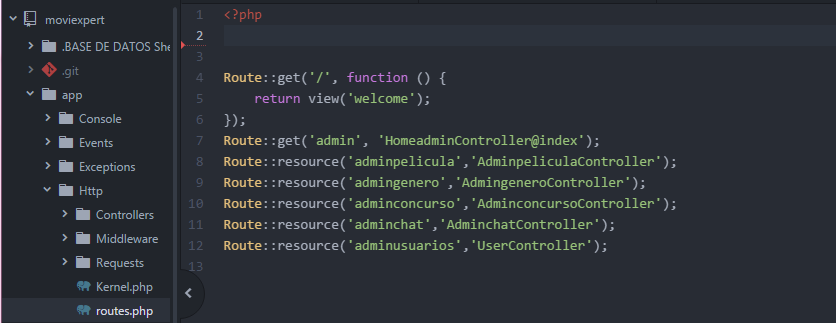
Ejemplo:

Route::get('admin', 'HomeadminController@index');

1. Route::resource(‘ruta’,’nombre\_controlador’); -> Este método agiliza mucho las cosas a la hora de utilizar un controlador como intermediario puesto si un controlador va a tener varias vistas asociadas no hace falta crear un Route::get para cada una de ellas, sino que con Route::resource le indicamos el controlador y lo único que tenemos que hacer es crear la vista con el nombre de la función a la que se va a enlazar al controlador y el solo interpreta las rutas.

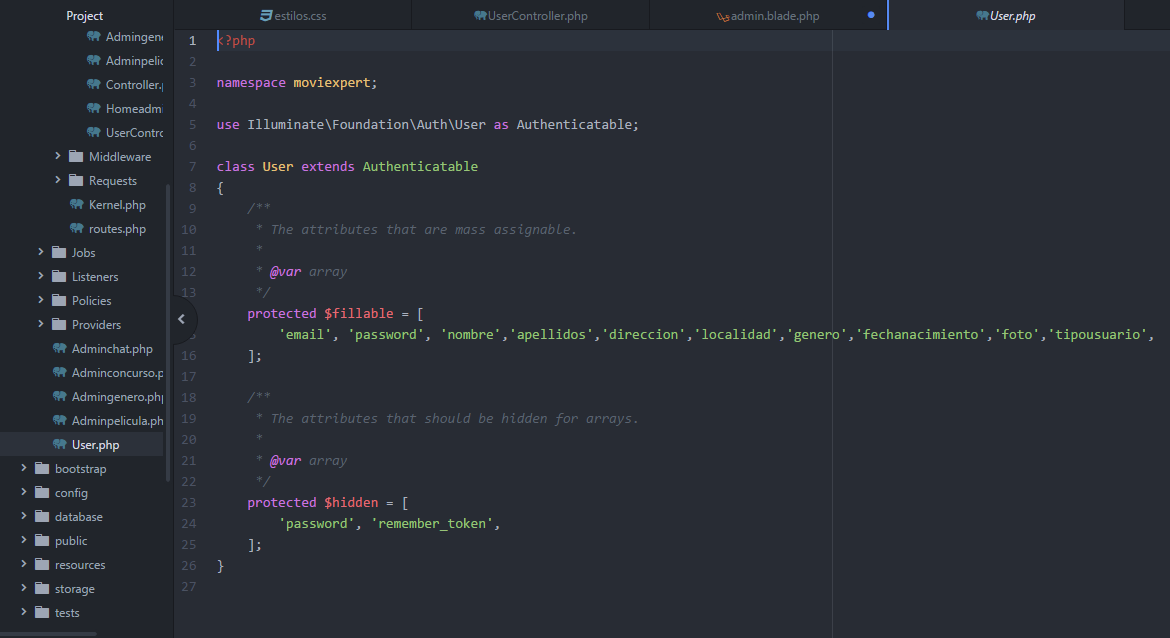
Ejemplo:

Route::resource('adminusuarios','UserController');

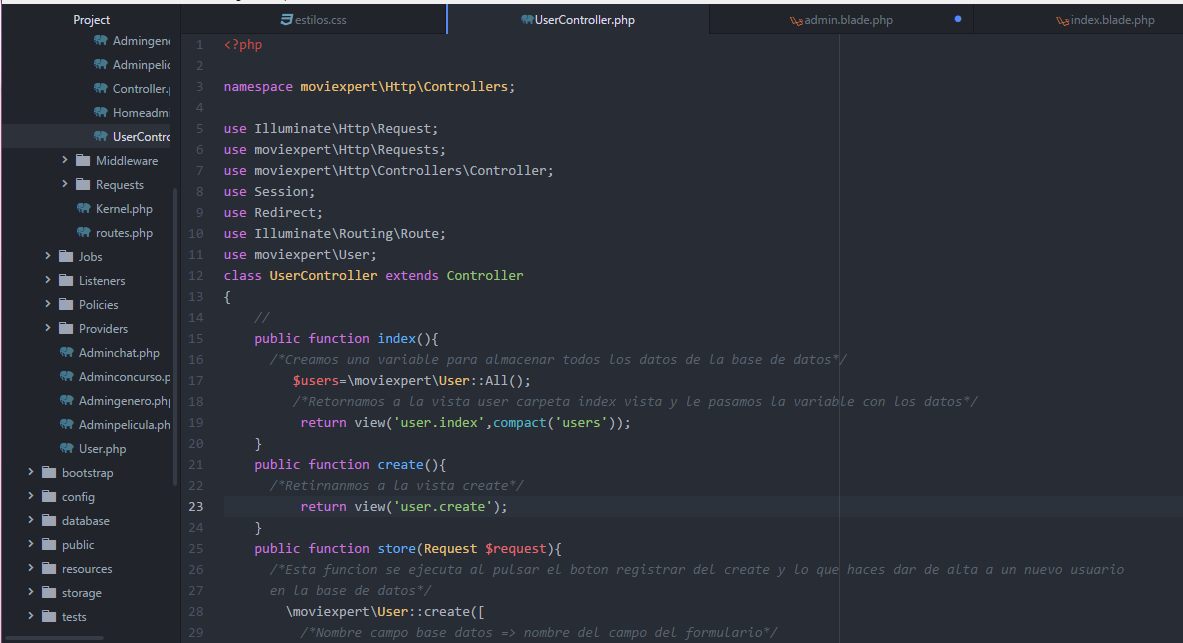


El siguiente paso sería enlazar los componentes para ello vamos realizamos las siguientes acciones:

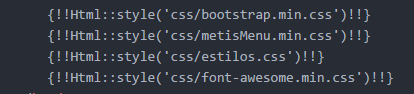
1. Editamos el modelo para indicarle cuales son los campos de la tabla con los que vamos a trabajar, es decir, que campos va a rellenar cuando hagamos una inserción de datos a esa tabla o los campos de los cuales nos va a devolver los datos cuando hagamos una consulta.

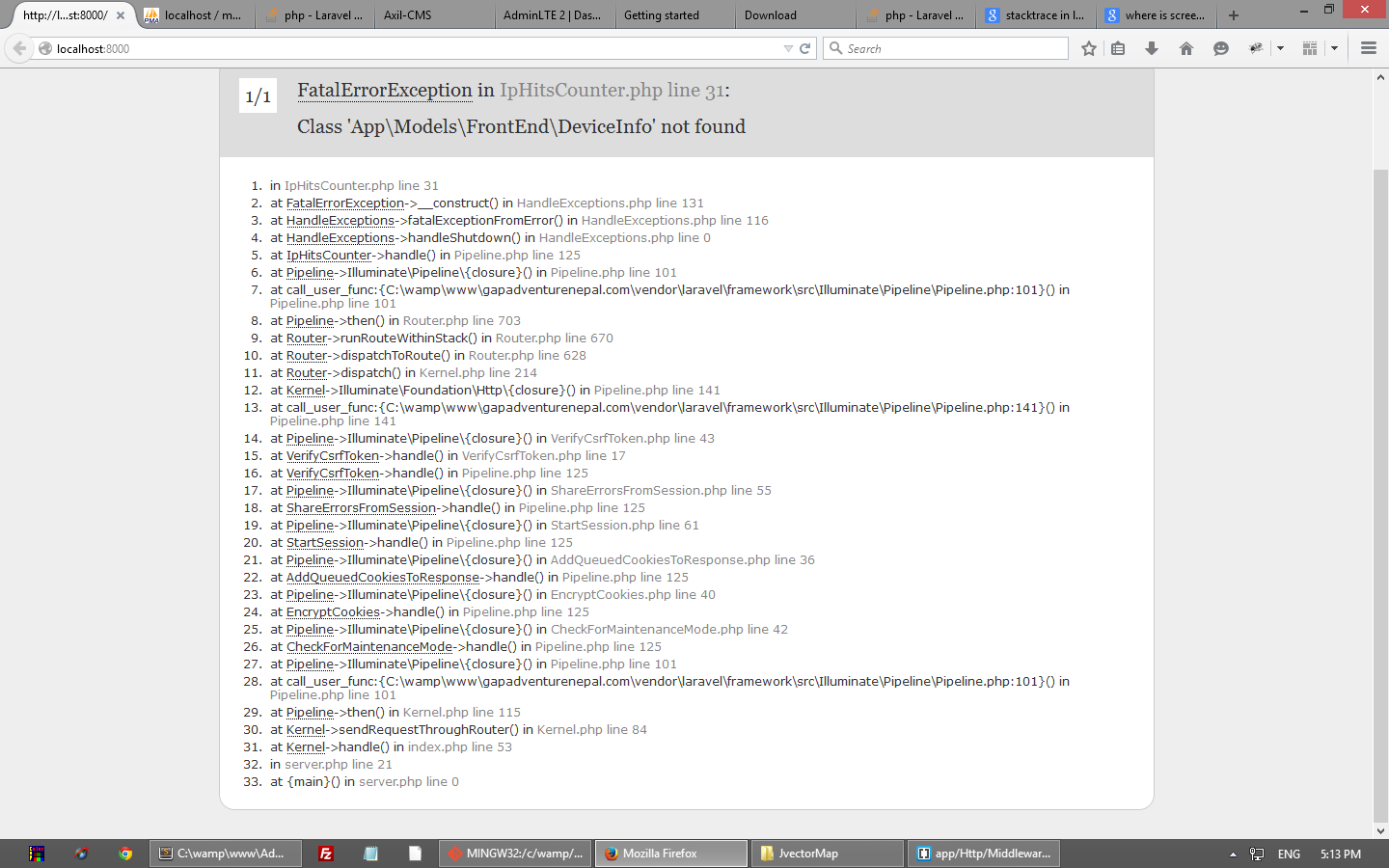


1. Editamos el controlador y vamos a la función asociada a la vista que estamos creando y vamos hacer dos cosas, la primera guardar los datos que le solicitamos al modelo en una variable y en segundo lugar retornamos a la vista pasándola la variable con todo los datos.



Vamos al navegador e introducimos la ruta localhost:8000/admin y nos debería mostrar la vista creada pero nos da un error que nos indica que no reconoce las etiquetas HTLM y FORM, estas etiquetas las utilizamos porque Laravel tiene un etiquetado propio a la hora de crear formularios y enlaces.





Para solucionar este problema tenemos que instalar el LaravelCollective vamos a la página oficial de Laravel para seguir el manual y poder configurarlo.



Cuando tengamos configurado y funcionando el LaravelCollective volvemos al navegar y volvemos hacer la prueba y cómo podemos apreciar ya podemos visualizar la vista. Configuramos todas las vistas, modelos y controladores para todas las opciones que aparecen en el menú.

