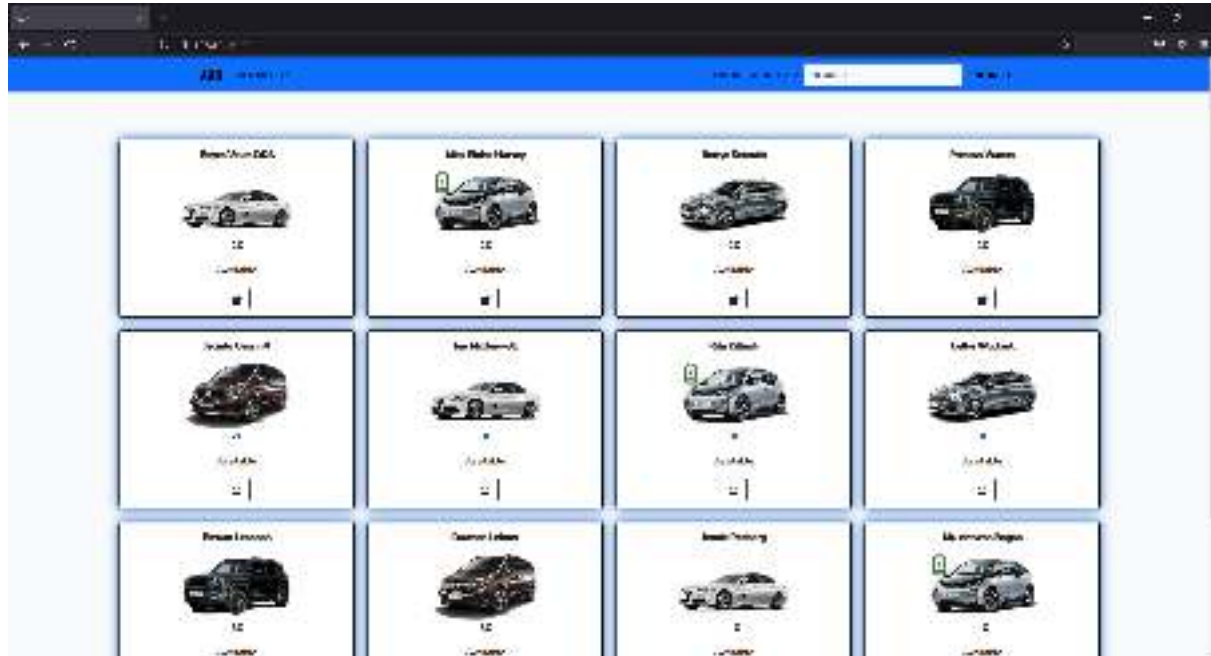


Mi primera página en laravel



[Repositorio del proyecto](#)

ÍNDICE

Introducción	3
Desarrollo	3
Primera Semana	3
Segunda Semana	4
Tercera Semana	5
Manejo de la página	6
Conclusión	12
Bibliografía	12

Introducción

El proyecto final de la asignatura de Desarrollo Web en Entorno Servidor iba ser una página que simula una biblioteca donde tengas un usuario administrador-cliente dónde puedas solicitar préstamos y estos mismo en caso de no cumplir la fecha de entrega, se genera una penalización.

En mi caso, como no quería hacer el proyecto sobre una biblioteca porque no me gustaba. Por lo tanto pensé hacerlo sobre algo que esté relacionado con los coches y que a la vez tenga la función de solicitar préstamos y generar penalizaciones. Con esto en mente, me decanté por hacer una página de alquileres de coches.

Teniendo en cuenta la idea que quería desarrollar, antes de empezar a codificar me organicé el proyecto de la siguiente manera:

- **Primera Semana:** Establecer las tablas que usaría en la base de datos relacional y una vez que tenga bien planteado como van a ser las relaciones, empezaría a codificar con el objetivo(sprint goal) tener todos las migraciones, seeders establecidas por último aplicar estilos de mi página.
- **Segunda Semana:** Si cumplía el primer plan lo siguiente era crear mis roles y permisos para empezar a crear las vistas de la página diferenciando entre el perfil administrador y cliente. Por último implementar la lógica de préstamos y la entrega de productos.
- **Tercera Semana:** Para la última semana quedaría implementar la lógica de penalización y añadir el sistema de búsqueda y hacer debugger a la página.

Desarrollo

Primera Semana

Como ya decía anteriormente, antes de empezar a codificar tenía que tener las tablas relacionadas bien planteadas. Para ello usé draw.io que lo podrás encontrar en la tienda de Microsoft para Windows.

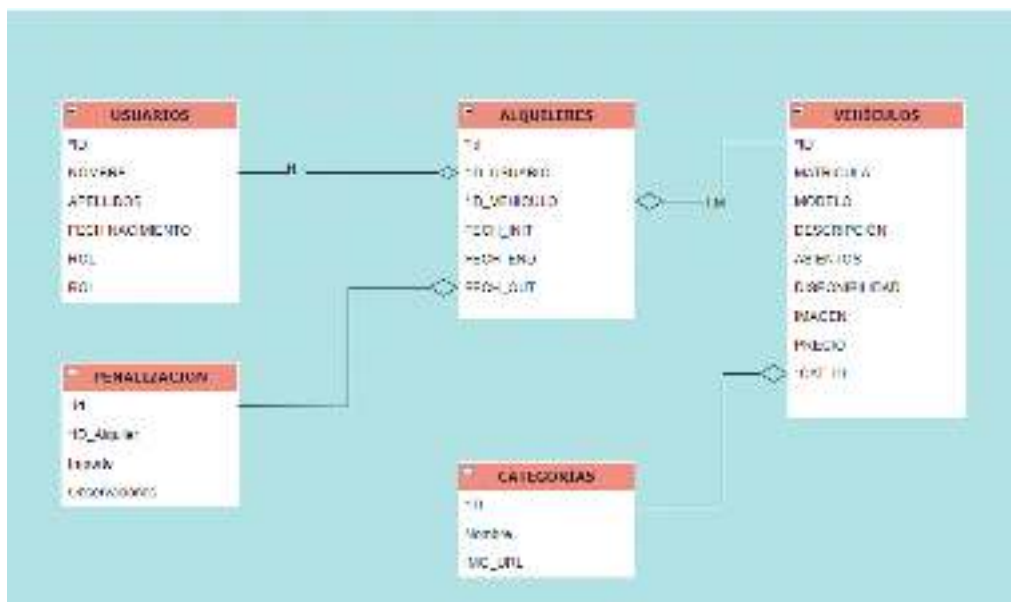
Al ser una alquiler de coches debía de tener las siguientes tablas:

Primero quería categorizar los tipos de vehículos en Monovolumen, Furgonetas, Eléctricos y Camiones dónde la relación sería de uno a muchos ya que por cada vehículo pertenece a una sola categoría pero dentro de una categoría habrá uno o más vehículos.

Segundo, como queremos alquilar necesitamos a unos usuarios que serán prácticamente clientes y un administrador que para ello creamos el campo rol. El objetivo es, que el clienta

pueda alquilar más de un vehículo, para creamos la tabla alquileres que nos permitirá relacionar usuarios y vehículos, con esto generamos un alquiler que tendrá la id de renta, id de usuario y la id del vehículo, por último los campos fechas que nos ayudará a tener un control de la renta ya que tendremos una fecha de inicio, una fecha de la entrega estimada y una fecha de devolución que lo podrá el administrador y el inicio y entrega las elige el cliente y estas no se podrán modificar por el cliente.

Tercero, gracias a los campos fechas podemos calcular el diferencial de días (fecha de entrega estimada y la fecha de devolución) en caso que se entregue fuera de plazo, se generará una penalización que almacenará la id de usuario e id de renta. Para simplificar el proyecto decidí que las penalizaciones de una alquiler de vehículos contengan en la tabla de penalizaciones un campo de observaciones dónde se comente las acciones por el cual ha sido penalizados como por ejemplo, parachoques roto, interior sucio, ... De esta manera me ahorraría crear tablas externas que se usarían específicamente para detallar las acciones penalizables. Por supuesto para el último campo que es importe que es donde tendrá exclusivamente el coste de la penalización que es un complementario del coste de alquiler.



Una vez hechas las tablas instalé la base de laravel dónde empecé a generar las migraciones y los seeders, esto no me ocasionó ningún problema ya que para generar el 90% de los datos usé la librería [Faker](#) y para las imágenes me descargué la página de [Cicar Las Palmas](#) para luego sustraer su directorio de imágenes que usaría en mi web, con esto ahorré mucho tiempo, porque si no, tendría que buscar cada imagen y editarlas individualmente para que una resolución específica para la página como por ejemplo pasar de 1980x1080 a 250x250 píxeles.

Para terminar la semana me creé una vista de entrada a la página que me mostrara todos los productos y en el menú de navegación tenía dropdown con las categorías, todo esto está paginado.

Segunda Semana

Para esta semana tuve que implementar un sistema de login y registro para poder de alta usuario y para acceder a la página según el rol con el paquete [Spatie](#), para el cifrado de contraseñas usé la función bCrypt. Con esto ya pude crear mis grupos de los roles y permisos que para ello me documenté en esta [página](#).

Además, con mis roles y permisos definidos me puse a crear todas las vistas que tendría mi usuario cliente, usuario administrador y el invitado, éste último solamente puede acceder a la página principal para ver los productos.

Mientras tanto, al tener conectadas las vistas con sus rutas según su grupo de roles y permisos, empecé a definir la lógica de la página dentro de los controladores. No fue complicado pero sí tedioso porque había mucho trabajo de por medio y hasta algunas veces repetitivas. Quizás podría ser complicado pero para mí no lo fue investigar cómo tratar las fechas porque en mi caso no quería transformar los datos de tipo date a String y para ello encontré que laravel tiene el tipo [Carbon](#) que trae muchas funcionalidades para trabajar con fechas y todo esta viene con la instalación base de laravel. Por último para establecer las fechas para el cliente implementé un jquery de la siguiente página [JQueryInterface](#).

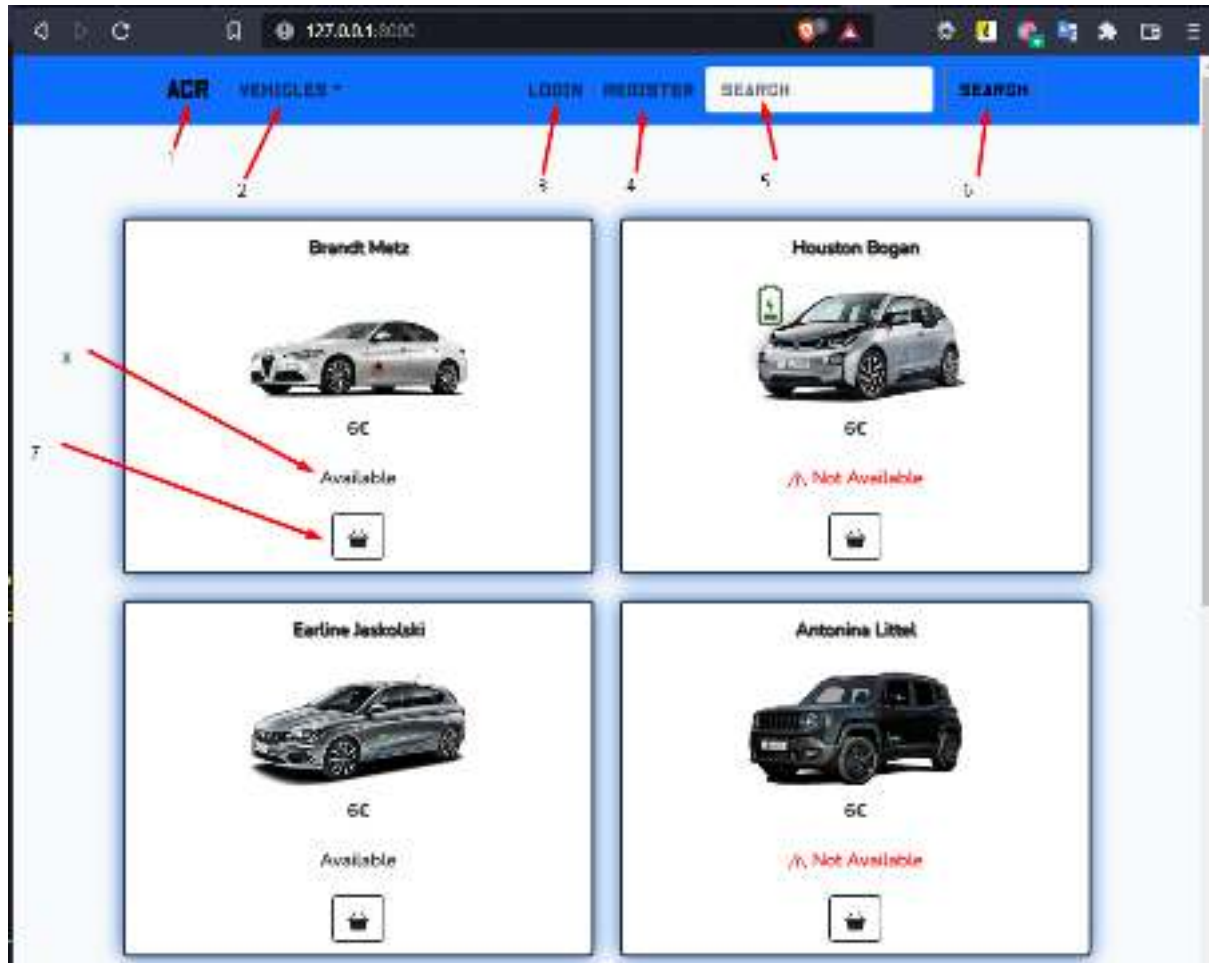
Lo más complicado que me sucedió esta semana fue editar los estilos de bootstrap que me hicieron perder una tarde entera para cambiar un color y establecer otra fuente de tipografía para mi navbar y encima también me había equivocado con el directorio resource donde estarían mis los estilos cuando deberían estar en el directorio public.

Tercera Semana

Con lo anterior ya tendría el 90% de la página hecha, ya solo me faltaba hacer la lógica de penalización relacionándolo con la renta. Hecho esto ya me puse hacer que fuera más usable la página independientemente del usuario como por ejemplo añadir sistema de búsqueda en todas las páginas, poner si un producto está disponible o no, en la parte administrativa, modifiqué las tablas para que fuera más legibles. Por ejemplo, cambié los id de usuario, vehículo,... por sus nombres pero sin perder su relación con las ID porque si no perdería funcionalidad para añadir, modificar y eliminar elementos ya que para realizar las acciones anteriores, en los controladores buscaba por ID en las tablas, para no cambiar la lógica. Para ello usé la siguiente documentación de laravel [Eloquent Relationships](#).

Manejo de la página

Estando en la página principal podemos lo siguiente:



1. Botón de retorno a la página principal.
2. Menú que nos redirija a otra página según la categoría.



3. Botón para iniciar sesión.

4. Botón de registro de usuario.
5. Cuadro para buscar un vehículo según nombre.
6. Botón aplicar el filtro de búsqueda.
7. Botón que nos abrirá una página con más información sobre el producto antes de realizar la compra.
8. Información sobre disponibilidad del producto.

Para probar la página ya tengo una serie de usuarios registrados previamente para testear.

```
public function run()
{
    User::create([
        'name' => 'agustin',
        'email' => 'agus@agus.es',
        'password' => bcrypt('123456789'),
    ])->assignRole('admin');

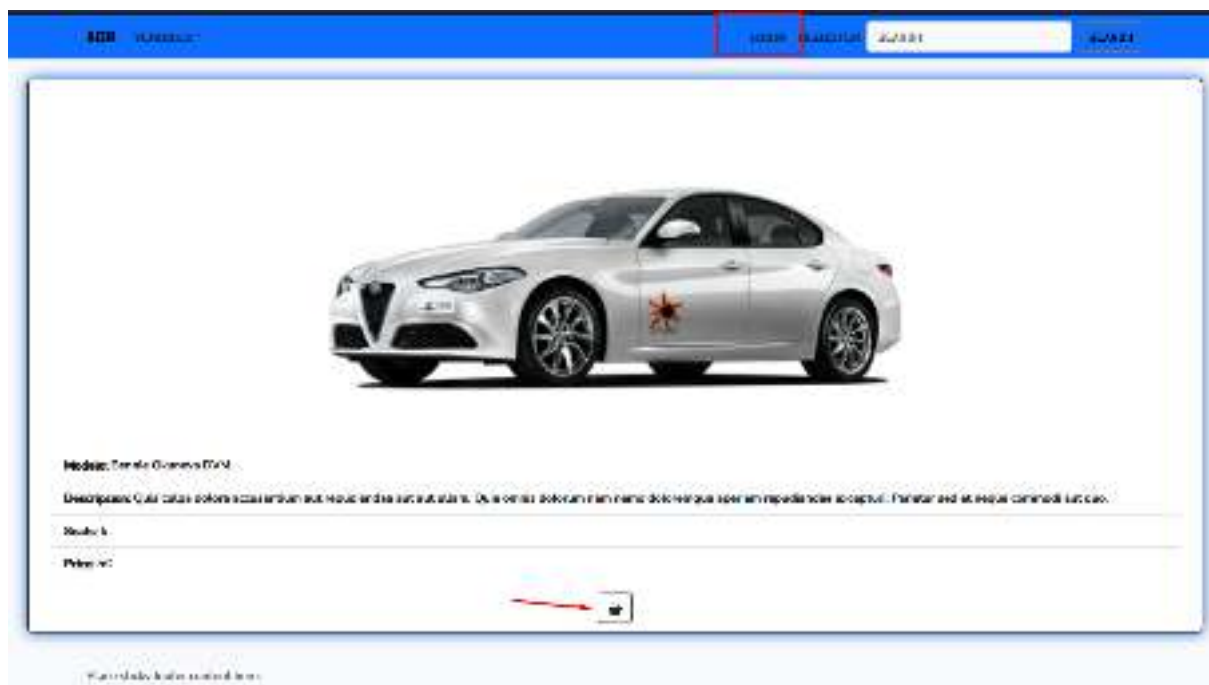
    User::create([
        'name' => 'test',
        'email' => 'test@test.es',
        'password' => bcrypt('123456789'),
    ])->assignRole('client');

    User::create([
        'name' => 'test2',
        'email' => 'test2@test2.es',
        'password' => bcrypt('123456789'),
    ])->assignRole('client');
}
```

Ahora vamos a simular el uso de la página, vamos a ser usuarios invitados y vamos a intentar alquilar el siguiente vehículo.



Una vez pulsemos el carrito nos redireccionará a una página que nos dará más detalle del producto y cabe destacar que no estamos logeados aún.



Si volvemos a pulsar sobre el botón del carrito nos pedirá que nos identifiquemos y en caso no poseer cuenta nos registramos pero en nuestro caso ya tenemos de usuarios cliente creados llamados test que lo mostramos anteriormente.

Una vez logueados nos vamos al producto que queríamos rentar y nos saldrá lo siguiente.

Seleccionamos el rango del alquiler(inicio y fin), y ya con esto tendríamos la solicitud de renta.

Si queremos visualizar el estado de nuestras rentas tenemos que pulsar en "My Rents". En mi rentas nos mostrará todos los datos que conlleva renta y lo más importante el estado del alquiler que por defecto estará en revisión hasta que un administrador nos acepte o rechace.



ID	Email	Vehicle ID	Date Start	Date End	Date Give	Cost	Status
1	test@tesla.com	1	2022-02-03 09:00:00	2022-03-02 16:00:00		0	expiracion

Please check Status column below.

Una vez hecho el alquiler el vehículo en cuestión nos aparecerá en la página como no disponible y vamos a intentar rentar ese vehículo con el usuario test2 por ejemplo.



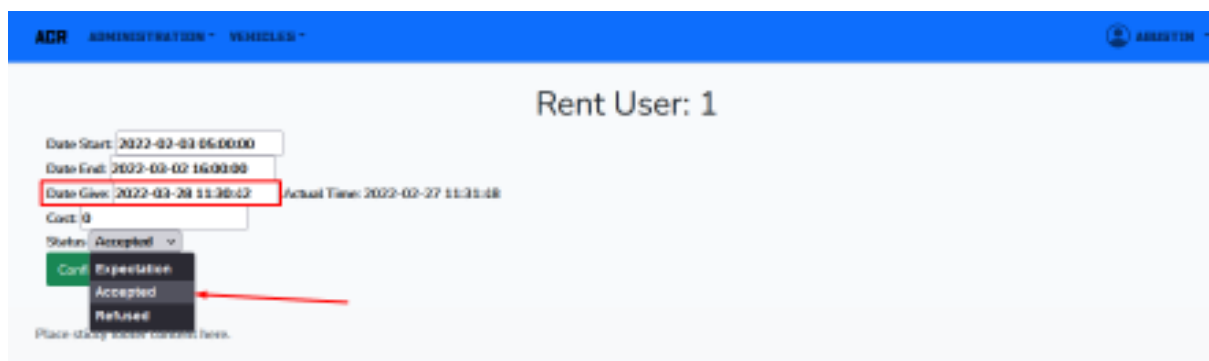
Como se puede observar cuando pulsamos en el carrito nos muestra un mensaje advirtiendonos que ese vehículo está rentado y por ello no nos deja alquilarlo.



Anteriormente comentábamos que la solicitud del préstamo está en revisión y queremos aceptar su solicitud, para ello nos logueamos con agus que es el usuario administrador nos vamos a la parte administrativa y entramos en rentas.



Pulsamos en editar y aprovechamos para darle una fecha de entrega tardía para que nos genere la penalización.



Cuando añadimos una fecha de entrega siempre nos va generar el coste del alquiler estimado como se puede observar en la siguiente captura y el coste de penalización, tendremos que ir a la parte administrativa de penalizaciones para ver los datos y también si queremos añadir comentarios negativos como por ejemplo depósito vacío, golpe en el parachoques,...



Como podemos observar tenemos el id de la renta, el coste de la penalización que siempre será un complemento extra del coste del alquiler y los comentarios



ID	Rent ID	Cost	Comments	ACTIONS
1	1	126.17	Day Late 24.01	 

Con esto ya estaría hecho el ejemplo de manejo de la página.

Conclusión

Estoy bastante orgulloso de mi resultado, lo he hecho a base de ver tutoriales, documentación o foros. He aprendido mucho haciéndolo totalmente por mi cuenta ya que era mi objetivo principal ponerme a prueba para saber si era capaz de cumplir con los requisitos a pesar que hubieron días que no me salían las cosas a la primera.

También tengo que decir que hubo días que me acostaba tarde haciendo el proyecto porque perdía bastante tiempo documentándome y por los errores. A pesar de llegar a un resultado óptimo y las horas de documentación siento que no he aprendido nada por culpa de las capacidades de laravel, porque tiene muchas librerías, métodos que nos podrían ahorrar mucho tiempo. A pesar de que pueda parecer una contradicción he aprendido bastante pero con respecto al laravel solo habré aprendido menos del 25%, quizás.

Para terminar pienso que con más tiempo se podría haber hecho mejor, pero para llevar aproximadamente tres semanas no me quejo y repito estoy altamente orgulloso de mi resultado ya que lo hice sobre algo que me gusta como son los coches.

Bibliografía

[Documentación Oficial Laravel8](#)

[JQuery UI](#)

[Bootstrap5](#)

[Spatie](#)

[Tutoriales Laravel8](#)

[Tutorial Roles y Permisos](#)

[Carbon](#)

[Sweet Alert](#)

[Mi repositorio de github](#)