



TÉCNICO SUPERIOR EN DESARROLLO DE APLICACIONES WEB

Departamento de Informática



MotoGest Manual Técnico

Autor/es: Sergio Espinosa Pascua
Curso Académico: 2023/2024

Índice

| | |
|---|-----------|
| 1 Introducción..... | 3 |
| 2 Arquitectura de la aplicación..... | 3 |
| 2.1 Frontend..... | 3 |
| 2.1.1 Tecnologías usadas..... | 3 |
| 2.1.2 Entorno de desarrollo..... | 4 |
| 2.2 Backend..... | 4 |
| 2.2.1 Tecnologías usadas..... | 4 |
| 2.2.2 Entorno de desarrollo..... | 4 |
| 3 Documentación técnica..... | 5 |
| 3.1 Análisis..... | 5 |
| 3.1.1 Diagrama de entidad relación..... | 5 |
| 3.1.2 Diagrama UML..... | 6 |
| 4 Proceso de despliegue..... | 6 |
| 4.1 Despliegue del hosting..... | 6 |
| 4.2 Despliegue local..... | 8 |
| 5 Propuestas de mejoras..... | 12 |
| 6 Bibliografía..... | 13 |

1 Introducción

Esta aplicación web tiene como objetivo principal brindar a los usuarios que la utilicen, una herramienta que les permita llevar un control sobre los mantenimientos y el kilometraje actual de su moto.

Los usuarios registrados tienen la posibilidad de crear tantas motos como tengan y poder acceder a la ficha de mantenimientos para llevar el control del kilometraje recorrido y en base a estos calcular cuando será el próximo mantenimiento recomendado. También tendrán la opción de llevar un pequeño control sobre los gastos que hayan tenido en base a los mantenimientos realizados.

Los usuarios con rol de administrador pueden gestionar los registros tanto de usuarios como de motos y sus mantenimientos. Pueden crear, editar y eliminar registros de usuarios y motos, los registros de mantenimientos solo se pueden eliminar. También podrán realizar las acciones de usuario normal.

2 Arquitectura de la aplicación

2.1 Frontend

2.1.1 Tecnologías usadas

Para el desarrollo del front, se han utilizado las siguientes tecnologías:

HTML: He utilizado HTML para crear la estructura y el contenido base de la aplicación web.

CSS: He utilizado algo de CSS para dar estilo personalizado.

Bootstrap: He utilizado Bootstrap en su versión 5 para simplificar el diseño y el desarrollo del front y tener un estilo simple pero agradable de ver.

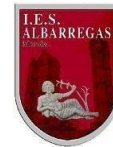
Blade: He utilizado Blade para poder añadir lógica y funcionalidades a los archivos donde se escribe código HTML.

Select2: He utilizado la librería de Select2 para poder incorporar selects con buscador por texto incorporado.

Font Awesome: He utilizado Font Awesome para añadir iconos en distintas partes de la web para dar una apariencia más divertida y original.

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: MotoGest



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

Sweetalert: He utilizado la librería de Sweetalert para incorporar algunas confirmaciones de acciones en la parte del administrador.

2.1.2 Entorno de desarrollo

El entorno de desarrollo utilizado para el front es el siguiente:

Visual Studio Code: He utilizado este IDE como editor de código principal para escribir y editar el código de HTML, CSS y JavaScript.

Navegador: He utilizado distintos navegadores como Google Chrome, Firefox, Microsoft Edge y Opera

Control de versiones: He utilizado GitHub para llevar un control de versiones del front de la aplicación.

2.2 Backend

2.2.1 Tecnologías usadas

Para el desarrollo del back, se han utilizado las siguientes tecnologías:

Lenguaje de programación: El lenguaje de programación que he utilizado para desarrollar la aplicación es PHP en su versión 8.1

Framework: He utilizado el Framework de Laravel en su versión 10.22.0 ya que destaca en el desarrollo web PHP gracias a su estructura MVC clara y organizada, su ORM Eloquent que simplifica las interacciones con la base de datos evitando tener que realizar consultas en bruto. También hace uso de un sistema de enrutamiento intuitivo, herramientas integradas para autenticación y autorización, junto con un amplio ecosistema de paquetes y una comunidad activa que brinda soporte constante, convirtiéndolo en una opción atractiva por su facilidad de uso y potencia para construir aplicaciones web.

Base de datos: He utilizado MYSQL 5.7.40 ya que tiene gran compatibilidad con múltiples sistemas y un buen rendimiento para el uso de aplicaciones web.

2.2.2 Entorno de desarrollo

Entornos de desarrollo utilizados en el back:

Visual Studio Code: He utilizado este IDE ya que es muy intuitivo y tiene un gran ecosistema de plugins y extensiones creadas por desarrolladores y la propia comunidad que agilizan el desarrollo además de la compatibilidad con todos los lenguajes utilizados.

PhpMyAdmin: He utilizado PhpMyAdmin ya que es gratuito y ofrece una interfaz web intuitiva para gestionar bases de datos MySQL, permitiendo crear, editar tablas, ejecutar consultas SQL, gestionar usuarios...

Control de versiones: He utilizado GitHub para llevar un control de versiones del back de la aplicación.

3 Documentación técnica

3.1 Análisis

3.1.1 Diagrama de entidad relación

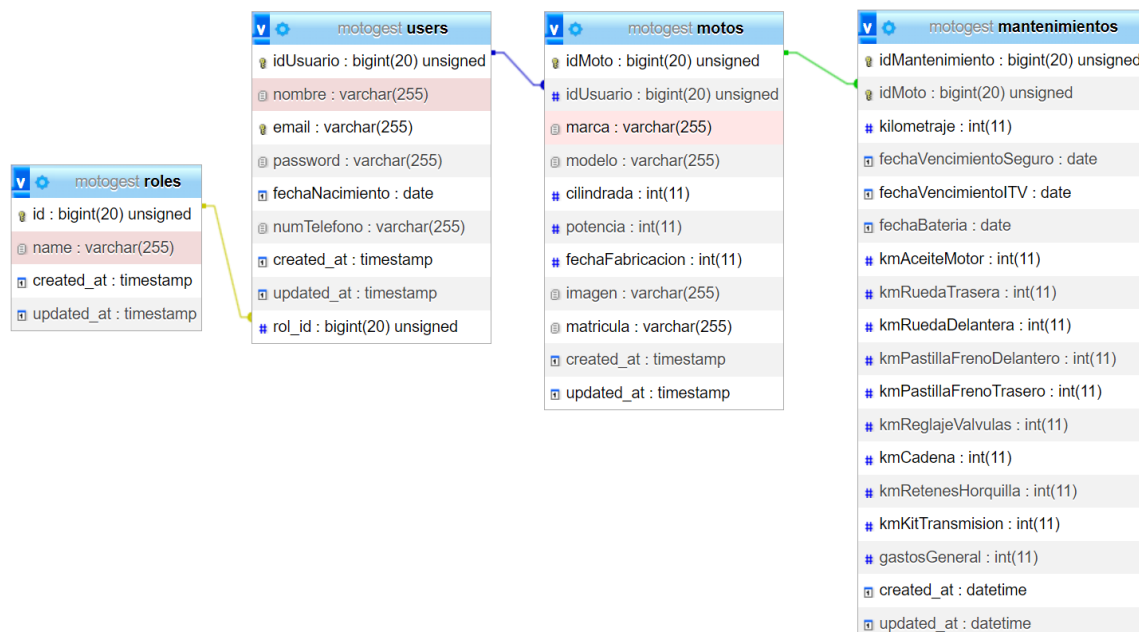


Imagen 1: Diagrama de entidad relación

3.1.2 Diagrama UML

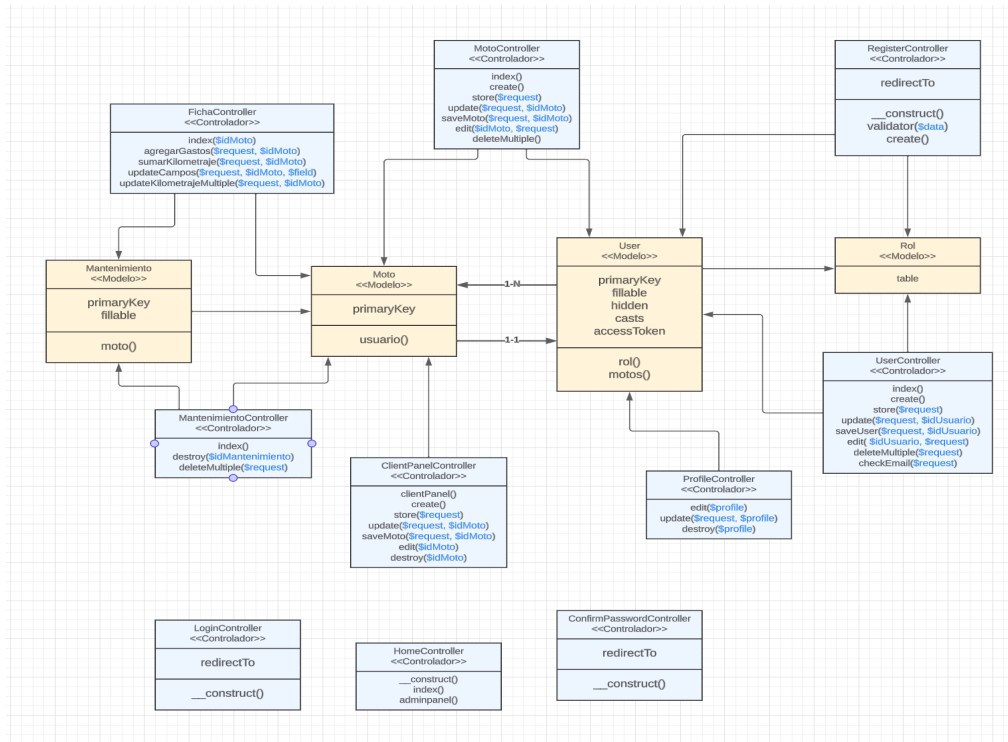


Imagen 2: Diagrama UML

4 Proceso de despliegue

La aplicación se ha desplegado en el siguiente dominio: <https://motogest.online/>

He creado un usuario con poderes de administrador para realizar pruebas:

Correo: iesalbarregas@gmail.com

Contraseña: [albarregas@2023](#)

Otro usuario con poderes normales:

Correo: iesalbarregasUser@gmail.com

Contraseña: [albarregas@2023](#)

4.1 Despliegue del hosting

La aplicación se ha desplegado en <https://www.hostinger.es/> . Hostinger es un proveedor de hostings de pago.

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: MotoGest



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

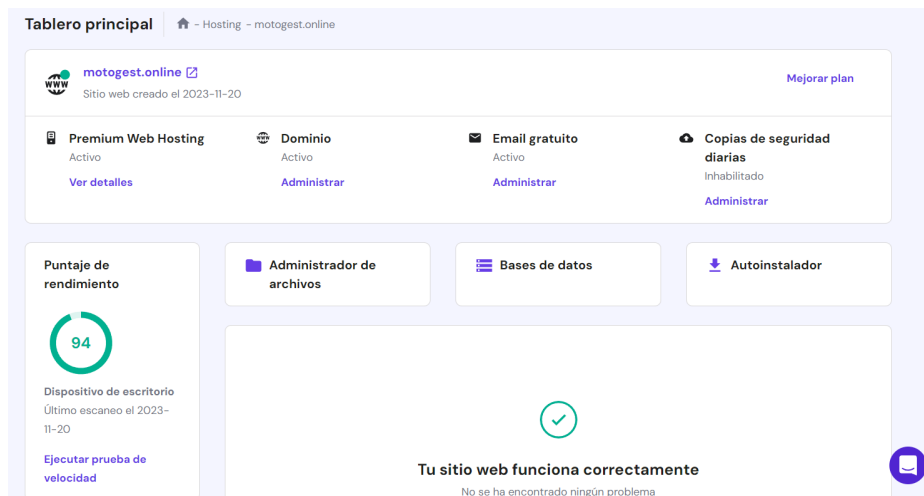


Imagen 3: Panel de control de hostinger

Este hosting incluye numerosas herramientas como un sistema de compra de dominios, gestor de bases de datos con PhpMyAdmin. He decidido usar hostinger ya que por el precio ofrece suficientes herramientas para gestionar fácilmente la web y realizar cambios de forma sencilla.

Gestor de bases de datos de Hostinger:

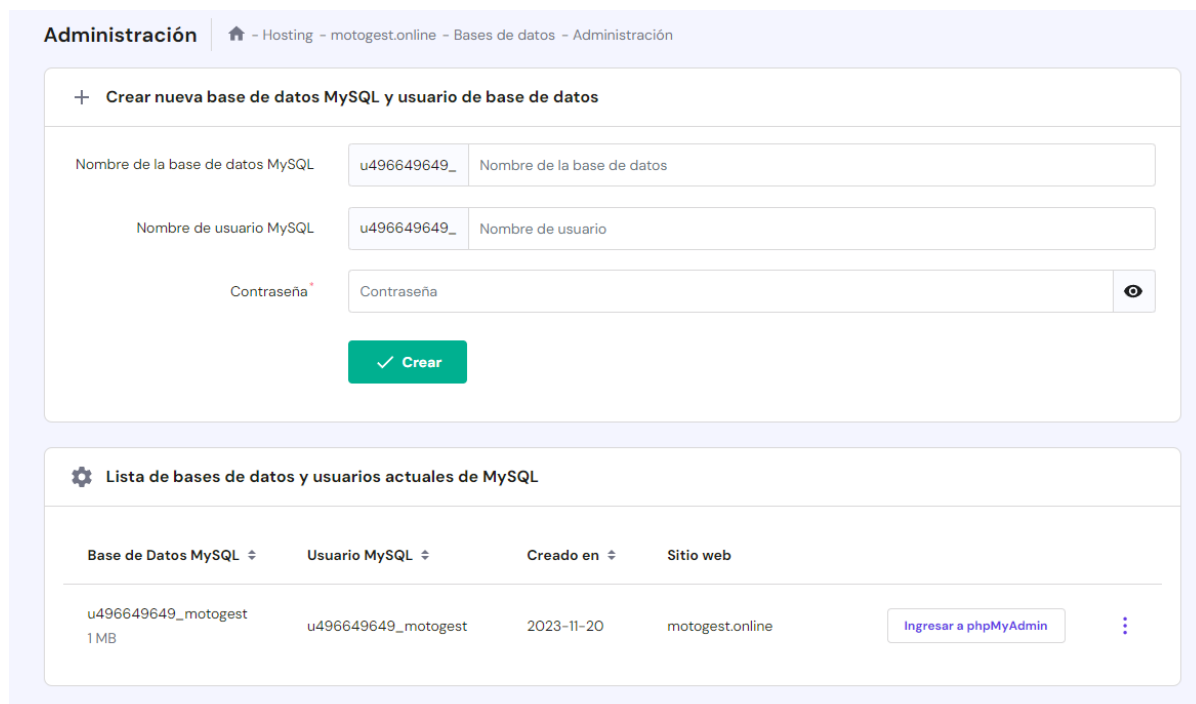


Imagen 4: Panel de control de la base de datos en hostinger

4.2 Despliegue local

A continuación se explicará cómo desplegar la aplicación en local:

Requisitos para desplegar en local la aplicación:

- Instalar PHP

Primeramente instalaremos PHP en su versión 8.1 ya que esta versión y las versiones superiores son las compatibles. Encontraremos el enlace de descarga en esta web:

<https://windows.php.net/download#php-8.1> Seguiremos los pasos de instalación y para verificar que se está se ha realizado correctamente abriremos una ventana de la consola y escribiremos “php –v”.

Nos deberá aparecer algo parecido a esto:

```
C:\Users\Sergio>php -v
PHP 8.2.0 (cli) (built: Dec 6 2022 15:31:23) (ZTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.2.0, Copyright (c) Zend Technologies

C:\Users\Sergio>
```

Imagen 5: Comprobación de instalación exitosa de PHP

- Instalar Composer

Después instalaremos Composer, un administrador de dependencias que nos facilitara la gestión de bibliotecas y paquetes de terceros en proyectos PHP. Encontraremos el enlace de descarga en esta web: <https://getcomposer.org/download/>. Seguiremos los sencillos pasos de instalación y para comprobar que se ha instalado correctamente abriremos la consola y escribiremos “composer”. Nos tendría que aparecer algo similar a esto:

```
C:\Users\Sergio>composer

Composer

Composer version 2.5.8 2023-06-09 17:13:21

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display help for the given command. When no command is given display help for the list
                             command
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi|--no-ansi          Force (or disable) ANSI output
  -n, --no-interaction      Do not ask any interactive question
  --profile                Display timing and memory usage information
  --no-plugins              Whether to disable plugins.
  --no-scripts              Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache                Prevent use of the cache
  -v|vv|vvv, --verbose     Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
                             3 for debug

Available commands:
  about               Display information about Composer
  archive             Create a distributable archive of the dependencies
  audit               Audit the dependencies for known vulnerabilities
  base                Display the project's base directory
  clear-cache         Clear the Composer cache
  config              Display the current configuration
  create-project       Create a new project from a template
  diagnose            Diagnose the system and display recommendations
  dump-autoload        Dump the autoload files
  exec                Execute a command in the working directory
  fund               Discover funding information for installed packages
  global-config        Display the global configuration
  global-status        Display the global status
  help               Display help for a command
  install             Install the dependencies
  license             Display the license for a package
  list               List the installed packages
  local               Display the local packages
  low-memory          Enable low memory mode
  maintenance         Run the maintenance script
  name                Display the package name
  out-of-date         Check if the dependencies are up to date
  optimize            Optimize the autoload files
  overwatch           Watch the dependencies for updates
  package-info        Display the package information
  packages            Display the installed packages
  parse               Parse the composer.json file
  platform            Display the platform information
  platform-validate   Validate the platform information
  query              Query the installed packages
  require            Require a package
  require-dev         Require a development package
  require-suggest     Require a suggested package
  require-with-dependencies Require a package with its dependencies
  rm                 Remove a package
  run                Run a script
  scripts            List the scripts defined in the composer.json file
  self               Update Composer
  set                 Set a configuration option
  show               Show the details of a package
  source              Display the source of a package
  status             Display the status of the dependencies
  suggest            Suggest packages
  update             Update the dependencies
  validate            Validate the composer.json file
  why                Show why a package is required
```

Imagen 6: Comprobación de instalación exitosa de Composer

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: MotoGest



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

- Instalar MYSQL

En este caso para instalar MYSQL utilizaremos el software de XAMPP. Encontraremos el enlace de descarga aquí: <https://www.apachefriends.org/es/download.html>.

Una vez descargado e instalado accederemos a su panel de control e iniciaremos el servicio de MYSQL y Apache.

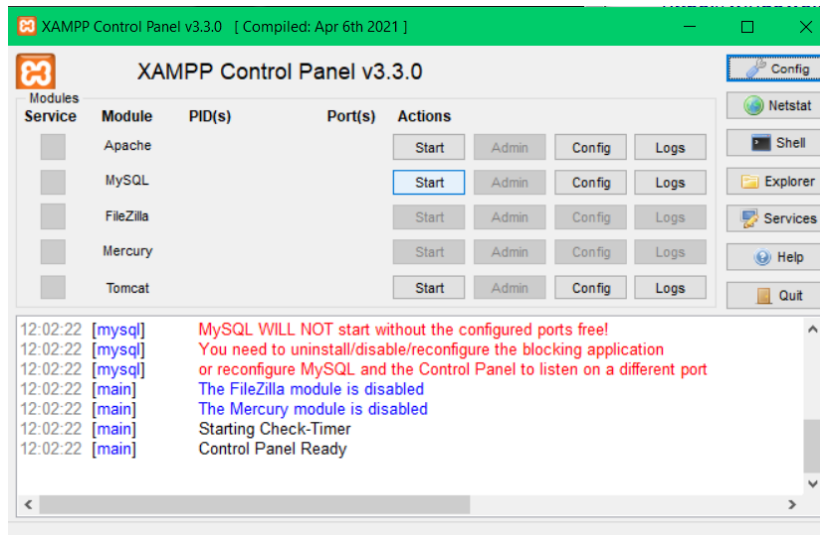


Imagen 7: Control Panel de XAMPP

Después accederemos a esta dirección en el navegador:

<http://localhost/phpmyadmin/index.php>

Introduciremos las credenciales:

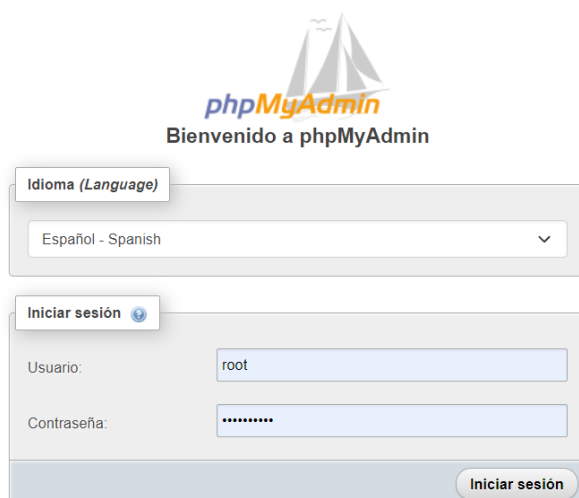
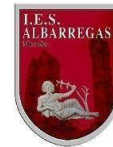


Imagen 8: Acceso a PhpMyAdmin

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: MotoGest



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

En el usuario introduciremos “root” y en la contraseña no pondremos nada pero en caso de no funcionar probaremos “password” ya que son las credenciales por defecto. En mi caso utilizare una contraseña establecida por mi que es “albarregas”. Más tarde necesitaremos esta contraseña, por lo que nos conviene recordarla.

Iniciaremos sesión y tendremos acceso a crear nuestras bases de datos. Crearemos una base de datos llamada “motogest” con la codificación “utf8_spanish_ci”.

Bases de datos

Imagen 9: Creación de la base de datos

-Descargar aplicación

Ahora descargamos la aplicación del siguiente repositorio: [GitHub](#).

Accedemos a la carpeta desde el IDE que queramos, en mi caso usaré Microsoft Visual Studio Code. Entramos al archivo .env y modificaremos las siguientes líneas con el nombre de nuestra base de datos y las credenciales que usamos anteriormente en phpmyadmin.

```
10 |
11 | DB_CONNECTION=mysql
12 | DB_HOST=127.0.0.1
13 | DB_PORT=3306
14 | DB_DATABASE=motogest
15 | DB_USERNAME=root
16 | DB_PASSWORD=albarregas
```

Imagen 10: Configuración de credenciales

Ahora abriremos un terminal en la raíz del proyecto y escribiremos “php artisan migrate”

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: MotoGest



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

Con este paso si todo ha ido bien se nos habrán creado las tablas y los campos en la base de datos:

| Tabla | Acción |
|---|-----------------------|
| <input type="checkbox"/> mantenimientos | ★ Examinar Estructura |
| <input type="checkbox"/> migrations | ★ Examinar Estructura |
| <input type="checkbox"/> motos | ★ Examinar Estructura |
| <input type="checkbox"/> roles | ★ Examinar Estructura |
| <input type="checkbox"/> users | ★ Examinar Estructura |
| 5 tablas | Número de filas |

Imagen 11: Base de datos creada con éxito

Ahora escribiremos en la consola otro comando que será “php artisan [db:seed](#)” Con este comando haremos que se inserten los roles necesarios y dos usuarios de prueba:

| | id | name |
|---|----|-------|
| <input type="checkbox"/> Editar Copiar Borrar | 1 | User |
| <input type="checkbox"/> Editar Copiar Borrar | 2 | Admin |

Imagen 12: Roles creados

| idUsuario | nombre | email |
|-----------|--------|-------------------------------|
| 1 | Admin | admin@gmail.com |
| 2 | Sergio | sespinosap05@iesalbarregas.es |

Imagen 13: Usuarios creados

Con todo esto realizado solamente nos faltará escribir en la consola “php artisan serve” y si todo ha funcionado como debería nos desplegará el proyecto en [“http://127.0.0.1:8000/”](http://127.0.0.1:8000/)

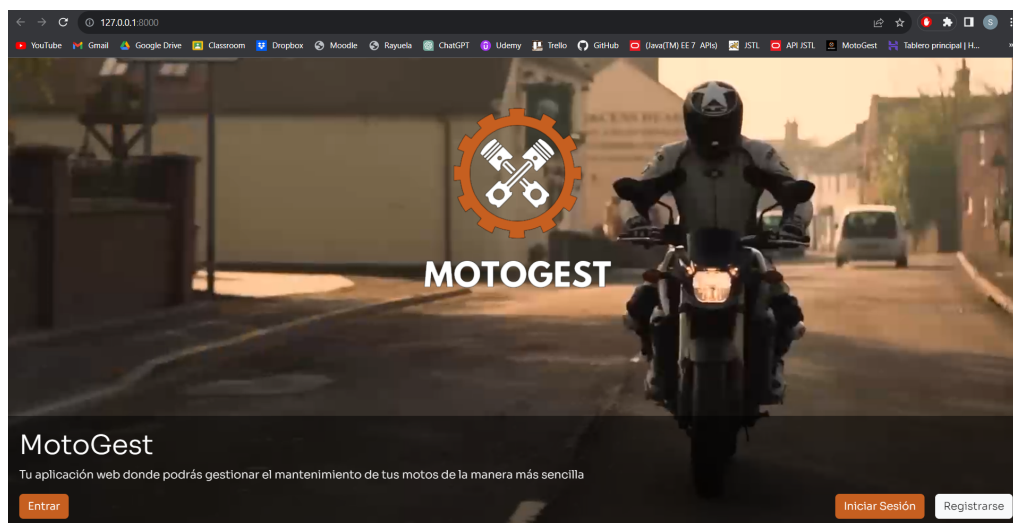


Imagen 14: Proyecto desplegado con éxito

5 Propuestas de mejoras

En un principio está planeado seguir con el desarrollo y las mejoras en esta aplicación, las futuras mejoras que tengo pensadas implementar son las siguientes:

- **Ampliar funcionalidades:** Me habría gustado disponer de más tiempo para desarrollar más funcionalidades para invitar al usuario que use la aplicación más habitualmente. Por ejemplo, podría implementarse una calculadora de consumo de combustible que sea intuitiva de usar y puedas llevar un control de los consumos realizados con tu moto, un sistema de creación de rutas mediante mapas y poder compartirlas con otros usuarios en un foro. También se podría implementar un sistema de búsqueda de gasolineras por ubicación.
- **Configurar servidor de correo:** Sería conveniente configurar un servidor de correo electrónico para que solo tuviesen acceso a la aplicación los correos electrónicos verificados.
- **Ordenación de tablas:** Actualmente la ordenación de tablas desde el administrador solo se realiza en la misma página donde se aplican, sería útil que se ordenara para todas las páginas restantes
- **Remodelación del sistema de mantenimientos:** Actualmente los mantenimientos que se pueden controlar y modificar están establecidos internamente por la base de datos y la programación de la aplicación. En un futuro me gustaría realizar una remodelación de este sistema y que el propio usuario tenga la libertad de decidir a qué componentes de la moto quiere hacerles un seguimiento.

6 Bibliografía

He utilizado los siguientes recursos:

- Laravel - the PHP framework for web artisans. (s. f.).
<https://laravel.com/docs/10.x/readme>
- Appearance | Select2 - the JQuery replacement for select boxes. (s. f.).
<https://select2.org/appearance>
- DevDocs — JavaScript documentation. (s. f.).
<https://devdocs.io/javascript/>
- PHP: manual de PHP - manual. (s. f.).
<https://www.php.net/manual/es/>
- Contributors,M.O.J.T.A.B. (s. f.). Get started with Bootstrap.
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- A, D. (s. f.). Tutoriales hostinger. Tutoriales Hostinger.
<https://www.hostinger.es/tutoriales/>
- Udemy. (2023). Udemy.
<https://www.udemy.com/course/desarrollo-de-apis-y-sitios-web-con-laravel-php-de-cero/>
- Software de diagramación en línea y solución visual | LucidChart. (s. f.). Lucidchart.
<https://www.lucidchart.com/pages/es>
- Find icons with the perfect look & feel | Font Awesome. (s. f.). Find Icons with the Perfect Look & Feel | Font Awesome.
<https://fontawesome.com/icons>
- SweetAlert2. (s. f.). a beautiful, responsive, customizable and accessible (WAI-ARIA) replacement for JavaScript's popup boxes.
<https://sweetalert2.github.io/>
- GitHub: Let's build from here. (s. f.). GitHub.
<https://github.com/>